

Markerless full-body human motion capture and combined motor action recognition for human-computer interaction

Luis Unzueta Irurtia

Tesis Doctoral, UNA, 2009



Director : Ángel María Suescun Cruces

URL: <http://edtb.euskomedia.org/id/eprint/5513>

RESUMEN / LABURPENA

Typically, people interact with computers by means of devices such as the keyboard and the mouse. Computers can detect the events coming from these devices, such as pushing down or releasing isolated or combined keyboard and mouse buttons, or the mouse motions, and then react according to the interpretation assigned to them. This communication procedure has been satisfactorily used for a wide range of applications. However, this approach lacks naturalness with respect to face-to-face human communication.

This thesis project presents a method for markerless real-time capture and automatic interpretation of full-body human movements for human-computer interaction (HCI).

Three stages can be distinguished in order to reach this objective: (1) the markerless tracking of as many of the user's body parts as possible, (2) the reconstruction of the kinematical 3D skeleton that represents the user's pose from the tracked body parts, and (3) the recognition of the movement patterns in order to make the computer "understand" the user's will and then react according to it. These three processes must be solved in real time in order to attain a satisfactory HCI.

The first stage can be solved by means of cameras focusing on the user and computer vision algorithms that extract and track the user's relevant characteristics from the images. This project proposes a method that combines color probabilities and optical flow. The second one requires to situate the kinematical 3D skeleton in plausible biomechanical poses fitted to the detected body parts, considering previous poses in order to obtain smooth motions. This project proposes an analytic-iterative inverse kinematics method that situates the body parts sequentially, from the torso to the upper and lower limbs, taking into account the biomechanical limits and the most relevant collisions. Finally, the last stage requires analyzing which are the significant features of motion in order to interpret patterns with artificial intelligence techniques. This project proposes a method to automatically extract potential gestures from the data flow and then label them, allowing the performance of combined actions.

AVISO LEGAL

Oharra: Tesi honen kontsulta burutzean, erabilerari dagokionez, beti ere honako baldintza hauek bete behar izango dira. Jabetza intelektualaren eskubideen titularrek baimena eman dute honako tesi hau eDTB zerbitzuaren bitartez zabaltzeko, baina eremu pribatuan landutako hezkuntza eta ikerkuntza-zko ekimenetan erabiltzeko bakarrik. Ezin da inolaz ere berau ezagutzera eman dirua irabazteko asmoarekin. Ezin da berau ezagutzera eman eDTB zerbitzutik kanpo dagoen inongo lekutik. Ezin da aurkeztu honen edukia eDTB-tik kanpo dagoen inongo leiho edo esparruan. Eskubideei buruzko baldintza hauek tesiaren aurkezpen laburpenari zein eduki osoari dagozkie. Tesia erabili edo honen pasarteak aipatzerakoan beti ere egilearen izena derrigorrez aipatu behar da.

Advertencia: La consulta de esta tesis queda condicionada a la aceptación de las siguientes condiciones de uso. La difusión de esta tesis por medio del servicio eDTB ha sido autorizada por los titulares de los derechos de propiedad intelectual únicamente para usos privados enmarcados en actividades de investigación y docencia. No se autoriza su reproducción con finalidades de lucro ni su difusión y puesta a disposición desde un sitio ajeno al servicio eDTB. No se autoriza la presentación de su contenido en una ventana o marco ajeno a eDTB (framing). Esta reserva de derechos afecta tanto al resumen de presentación de la tesis como a sus contenidos. En la utilización o cita de partes de la tesis es obligado indicar el nombre de la persona autora.

Warning: On having consulted this thesis you're accepting the following use conditions: Spreading this thesis by the eDTB service has been authorized by the titular of the intellectual property rights only for private uses placed in investigation and teaching activities. Reproduction with lucrative aims is not authorized neither its spreading and availability from a site foreign to the eDTB service. Introducing its content in a window or frame foreign to the eDTB service is not authorized (framing). This rights affect to the presentation summary of the thesis as well as to its contents. In the using or citation of parts of the thesis it's obliged to indicate the name of the author.

UNIVERSIDAD DE NAVARRA / NAFARROAKO UNIBERTSITATEA

**ESCUELA SUPERIOR DE INGENIEROS INDUSTRIALES /
INDUSTRIA INGENIARIEN GOI MAILAKO ESKOLA**

DONOSTIA - SAN SEBASTIÁN



**MARKERLESS FULL-BODY HUMAN MOTION CAPTURE AND
COMBINED MOTOR ACTION RECOGNITION FOR HUMAN-
COMPUTER INTERACTION**

MEMORIA

que para optar al Grado de Doctor
presenta

Luis Unzueta Irurtia-k

Doktore Gradua lortzeko
aurkezten duena

Donostia – San Sebastián, enero de 2009 / 2009-ko Urtarrila

*Dedicado a Hipaso de Metaponto,
"becario" de Pitágoras.*

AGRADECIMIENTOS

La elaboración de este proyecto de tesis ha sido como un largo caminar a través de la inmensidad del desierto en busca de la tierra prometida, con la canción de “La Historia Interminable” de fondo. No han faltado el hambre, la sed, la fatiga, el sofocante calor del día, el frío helador de la noche, los escorpiones, las serpientes, ni los espejismos. Pero por fortuna en el trayecto también me he topado con beduinos los cuales han compartido su sabiduría conmigo para hacer más llevadera esta travesía y poder llegar a buen puerto. A todos ellos les quiero agradecer su apoyo, ánimo y hospitalidad.

Como no podría ser de otra manera, para empezar agradezco a los directores del área de Simulación del CEIT contemporáneos a mi trabajo en ella, Juan Tomás Celigüeta y Luis Matey, el haberme dado su confianza para mi entrada y por renovarla año tras año.

Tampoco podía faltar mi jefe de proyectos, tutor y director de tesis, Ángel Suescun, el cual me ha dado la oportunidad de adentrarme en el apasionante mundo de la visión artificial, dándole una vuelta de tuerca más a mi gran afición a la informática, la cual nació allá por 1985 con un humilde MSX.

Entre los beduinos que me han acompañado a lo largo del trabajo de tesis destacan Javier Barandiaran e Iñaki Griego los cuales son unas sabias y grandes personas que no han dudado en ayudarme una y otra vez de una manera totalmente altruista pese a ser ayudas completamente oficiosas. También quiero destacar a Jokin Santurde y a Iñaki Moreno, los cuales, junto con los dos anteriores conformaron mi “Época Dorada” en el CEIT, tanto por los avances en el camino como por las risas que me pude echar junto con ellos.

Special thanks to Ronan Boulic from EPFL who has helped me enormously in the publication of my advances in the PhD. You have acted as a true thesis tutor by sharing your vast knowledge in R&D. Muchas gracias también a Manuel Peinado de la UAH, que pese a que nuestra relación ha sido enteramente por mail, ha estado a mi entera disposición para la casi eterna corrección del trabajo conjunto que hemos realizado.

Estoy enormemente agradecido también a Oskar Mena y Yaiza Vélaz los cuales han caminado junto a mí en la última etapa del trayecto, la más dura. Habéis sido un gran apoyo para la realización de pesadísimos experimentos, pruebas y demás, a partir de los cuales hemos aprendido unas cuantas cosas de la vida.

Muchas gracias a Basilio Sierra de la UPV/EHU por el interés y la dedicación que nos has prestado a Oskar y a mí para publicar y revisar nuestros avances cuando no teníamos para ofrecerte más que nuestros oídos.

Mi colaboración con el equipo de la UIB (Xavi Varona, Antoni Jaume, Jose María Buades, Cristina Manresa, Ramón Mas y Paco Perales) supuso la necesaria chispa inicial en esta andadura, y por ello, por su hospitalidad y también por todo lo que he aprendido gracias a ellos sobre visión artificial les estoy muy agradecido. Gracias también a Jordi González de la UAB por estar siempre dispuesto a resolver mis dudas sobre el método de reconocimiento de acciones de su tesis, la cual ha supuesto también mi bautismo de fuego en ese mundo, y también por revisar mi trabajo.

Muchas gracias también a Emilio Sánchez por haber sido un gran jefe de proyectos en cuyo contexto he realizado los avances decisivos de mi trabajo. También mil gracias a su hermano Jairo por compartir sus investigaciones sobre visión artificial las cuales me han sido de gran ayuda.

Muchas gracias a las entidades financiadoras de programas de becas predoctorales: la Fundación Centros Tecnológicos Iñaki Goenaga, el Departamento de Industria del Gobierno Vasco (Programa Ikertu), así como al Ministerio de Educación y Ciencia, junto con el Fondo Social Europeo por la concesión de una ayuda en el marco del programa de contratación de jóvenes investigadores (Programa Torres Quevedo).

Durante este periodo he compartido mi tiempo con mucha gente, a los cuales les estoy agradecido por el estímulo que han supuesto. Gracias especiales a Aitor Rodríguez, Javier Martín, Miguel Mateos, Mikel Ares, Xabier Carrera, Laurentzi Garmendia, Alberto Conde, Iparra, Aitor Cazón, Giovanni Berselli, Iker Aguinaga, Aiert Amundarain, Sergio Ausejo, Mikel y Xabi Pérez, Aritz Ustarroz, Mikel Osinalde, Miguel Castro, Erroman Telletxea, Diego Borro, Joan Savall, Javier Sánchez, Maite López, Raúl de la Riva, Aitor Ortuzar, Carlos Buchart, Eli Bengoechea, Jon Alonso, Hugo Álvarez, Alberto Lozano, Parri, Eva Elizalde, Gaizka San Vicente, Leire Suárez, Pablo Molinero, “Txusa”, Mariajo, Ana Leiza, Ana Sancho, Dimas López, Cristina Martín, Ion Irizar, Mario Álvarez, Mildred, Isa, Juan Liceaga, Iñigo García, Carlos Hernández, Alex Valero, Txemi, Ignacio Mansa, Txema e Iñigo Celigüeta, Manolo, Jose Luis, Imanol Puy, Raimundo, Paqui, Jeff Diamond, Leire Arizkuren, Jose María Sarriegi, Asier Alonso, Damien Maupu, Jorge Rodríguez, Javi Melo, Emanuele Ruffaldi, Carlo Avizzano, Alex Czarowicz, a mi cuadrilla y a todos aquellos que me perdonarán no haberlos nombrado pero que también están incluidos.

Finalmente muchas gracias a mi familia y a mi querida Ali por haberme comprendido y apoyado en este casi eterno camino.

CONTENTS

AGRADECIMIENTOS	I
CONTENTS	III
LIST OF FIGURES	VII
LIST OF TABLES	XIII
RESUMEN	XV
LABURPENA	XVII
ABSTRACT	XIX
1 INTRODUCTION	1
1.1 Motivation.....	1
1.1.1 <i>Human Motion Capture</i>	3
1.1.2 <i>Human Pose Reconstruction</i>	6
1.1.3 <i>Human Motor Action Recognition</i>	7
1.2 Contributions and Thesis Project Outline	8
2 RELATED WORK AND TAXONOMIES	11
2.1 General Problems and Assumptions.....	13
2.2 Applications	15
2.3 Camera Parameters and Multi-Camera Systems.....	16
2.4 Human Motion Reconstruction Survey.....	17
2.4.1 <i>Image-Features Extraction</i>	17
2.4.1.1 Feature Points	19
2.4.1.2 Edges.....	19
2.4.1.3 Blobs	19
2.4.1.4 Silhouettes	19
2.4.1.5 Visual Hull	20
2.4.2 <i>Human Body Parts and Human Shape Tracking</i>	20

2.4.2.1	Optical Flow	21
2.4.2.2	Point Tracking.....	21
2.4.2.3	Kernel Tracking.....	22
2.4.2.4	Silhouette Tracking	23
2.4.2.5	Occlusions.....	23
2.4.3	<i>Human Pose Reconstruction and Multibody Tracking</i>	24
2.4.3.1	Bottom-Up Approach	26
2.4.3.2	Top-Down Approach.....	28
2.5	Human Motor Action Recognition Survey.....	32
2.5.1	<i>Potentially Meaningful Motion-Features</i>	32
2.5.1.1	Templates	33
2.5.1.2	State-Space Models.....	34
2.5.2	<i>Motion-Features Classification Process</i>	35
2.5.2.1	Nearest Neighbors	36
2.5.2.2	Dynamic Time Warping.....	36
2.5.2.3	Hidden Markov Models	36
2.5.2.4	Dynamic Bayesian Networks	37
2.5.2.5	Neural Networks	37
2.5.2.6	Kernel Methods	37
2.5.3	<i>Gesture Spotting</i>	38
2.5.4	<i>Combined Motor Action Recognition</i>	38
2.6	Discussion	39
3	HUMAN BODY PART MARKERLESS TRACKING.....	41
3.1	Colored Features Optical Flow.....	42
3.2	Pelvis Position Tracking.....	48
3.3	Self-Occlusions.....	53
3.3.1	<i>Overlapping of Two Tracked Regions of Interest</i>	53
3.3.2	<i>Excessive Color Change of Tracked Regions of Interest</i>	55
3.4	Full-Body Capture Process Initialization.....	56
3.5	Experimental results	57

4 HUMAN FULL-BODY POSE RECONSTRUCTION	73
4.1 Sequential Inverse Kinematics	74
4.2 Spine Reconstruction.....	77
4.3 Clavicle Reconstruction.....	84
4.4 Upper and Lower Limb Reconstruction	85
4.5 Biomechanical Limits	88
4.6 Collision Avoidance.....	90
4.6.1 Torso-Elbow Collisions.....	91
4.6.2 Torso-Wrist Collisions.....	92
4.6.3 Foot-Floor Collisions	93
4.7 Experimental Results.....	94
4.7.1 Reconstruction Performance Evaluation.....	94
4.7.2 Sequential Inverse Kinematics Applied on a Marker-Based Mocap System.....	108
5 HUMAN MOTOR ACTION RECOGNITION.....	115
5.1 Gesture Spotting with Kinetic Pseudo-Energy History.....	117
5.2 Quasi-Static and Dynamic Gesture Recognition.....	119
5.3 Full-Body Pose Model for Database Search	121
5.4 Depth Warping in a Single-View Tracking	123
5.5 Human Full-Body Combined Actions.....	126
5.6 Experimental Results.....	127
5.6.1 Results on Dynamic Gesture Spotting and Recognition	127
5.6.2 Results on Combined Motor Action Recognition.....	133
6 CONCLUDING REMARKS	141
6.1 Conclusions.....	141
6.2 Future Work.....	144
GENERATED PUBLICATIONS.....	149
INDEX	169
REFERENCES	173

LIST OF FIGURES

Figure 1: Shannon and Weaver’s communication model.....	2
Figure 2: From left to right and top to bottom: Optical [© (Phasespace 2005)], magnetic [© (Ascension 2004)], mechanical [© (DoMotion 2004)] and inertial [© (XSens 2007)] marker-based motion capture systems.	4
Figure 3: The Nintendo Wii remote controller [© (Nintendo 2006)], the Apple iPhone [© (Apple 2007)] and the Sony Ericsson W910i [© (Sony- Ericsson 2007)] respectively.....	5
Figure 4: On the left a 2D image and on the right its corresponding depth map using a stereo camera [© (PointGrey 2007)]......	6
Figure 5: Skeletal multibody structure of a virtual character [©(MIRALab 2008)].	6
Figure 6: The pinhole camera model geometry.	16
Figure 7: From left to right and top to bottom: (a) Feature points, (b) edges, (c) the blobs of the doll’s hair and the bottle’s top, (d) the doll’s silhouette, and (e) the doll’s visual hull obtained from the silhouette backprojections of multiple views.	18
Figure 8: From left to right: point tracking, kernel tracking and silhouette tracking approaches.....	20
Figure 9: Pyramidal implementation of the Lucas-Kanade optical flow (Bouguet 1999; Lucas and Kanade 1981) applied to a rotating sphere. ...	21
Figure 10: Humanoids in the form of stick-, contour- and volumetric-figures [© (HUMODAN 2005)] respectively.	25
Figure 11: Softkinetic markerless motion capture [© (Softkinetic 2008)].	27
Figure 12: Organic Motion markerless motion capture [© (Organic-Motion 2008)].	32
Figure 13: The backprojection of a face obtained from the corresponding learned skin-color HS chrominance histograms.	45
Figure 14: Some samples of the tracking of a hand using CFOF. The forearm skin is also visible but it poses no problem for the system.	48
Figure 15: Background subtraction by Gaussian RGB model of pixels.	50
Figure 16: Background subtraction with a green backdrop behind the subject.	50

Figure 17: The estimated pelvis position (green dot) is calculated from the red pixels, which lie in the neighborhood of the median x coordinate of the highlighted pixels.	53
Figure 18: Markerless full-body motion capture samples maintaining the depths of the tracked body parts constant with respect to the view.	54
Figure 19: Left hand occluded by right hand. The occluded ROI boundary turns from green to red.....	54
Figure 20: Occlusion due to the excessive color change on the tracked right hand ROI. The occluded ROI boundary turns from green to red.....	56
Figure 21: Full-body markerless capture initialization process samples.....	57
Figure 22: Benchmark for evaluating 2D object tracking procedures.....	59
Figure 23: Histograms of the nRGB chrominance probability distributions in the face samples (values from 0 to 255).	62
Figure 24: Histograms of the HSV chrominance probability distributions in the face samples (values from 0 to 255).	62
Figure 25: Histograms of the HLS chrominance probability distributions in the face samples (values from 0 to 255).	63
Figure 26: Histograms of the YCrCb chrominance probability distributions in the face samples (values from 0 to 255).	63
Figure 27: Histograms of the Luv chrominance probability distributions in the face samples (values from 0 to 255).	64
Figure 28: Histograms of the xyY chrominance probability distributions in the face samples (values from 0 to 255).	64
Figure 29: An example of how too great a search area in Condensation may degrade the tracking if there is more than one blob with similar color characteristics.....	67
Figure 30: Face tracking (<i>Hispanic</i> sample) during partial occlusions using CFOF with PyrLK + CMDf + KF.	70
Figure 31: Face (<i>White</i> sample) trajectory differences running at 43 pixels/frame without occluding masks using CFOF (PyrLK + CMDf + KF) and the mPfinder (KF).	71
Figure 32: Face (<i>White</i> sample) trajectory differences running at 9 pixels/frame with occluding masks using CFOF (PyrLK + CMDf + KF) and Condensation.....	72
Figure 33: The readjustment of an equally distributed straight spine with 5 vertebrae.....	78

Figure 34: The resulting readjustment of a complete spine moving on its sagittal plane from the known positions of head and pelvis. On the top row, a bending movement, and on the bottom row, a stretching movement.	81
Figure 35: On the left, an outline of a spine region twisting procedure, and on the right, a full-spine twisting example where it can be seen how the vertebrae are gradually twisted in order to fit the end-effector orientations.	83
Figure 36: Diagram of the clavicle readjustment.	84
Figure 37: The resulting readjustment of a clavicle when its corresponding arm cannot reach the known wrist position by itself.	85
Figure 38: Diagram of the readjustment of a limb. On the left, the analytically calculable angles, and on the right, the undetermined swivel angle.	86
Figure 39: From left to right: swing and twist limits of a shoulder and flexion-extension limits of an elbow.....	89
Figure 40: Modeling of the swing biomechanical limits of a shoulder with a cubic spline.....	90
Figure 41: Diagram of the right elbow penetration depth estimation.....	91
Figure 42: Left elbow torso penetration: on the left the estimated penetration depth and on the right the penetration avoidance response posture.....	92
Figure 43: Left wrist-torso penetration: on the left the estimated penetration depth and on the right the penetration avoidance response posture.....	93
Figure 44: Right foot-floor penetration: on the left the estimated penetration depth and on the right the penetration avoidance response posture.....	94
Figure 45: Samples of the <i>Boxing</i> animation: on the left of each sample the SIK reconstructions and on the right the original postures.	95
Figure 46: Samples of the <i>Jump Kick</i> animation: on the left of each sample the SIK reconstructions and on the right the original postures.	96
Figure 47: Samples of the <i>Playground</i> animation: on the left of each sample the SIK reconstructions and on the right the original postures.	96
Figure 48: H-Anim structure (H-Anim 2008) of the humanoids of <i>Boxing</i> , <i>Jump Kick</i> and <i>Playground</i> animations. Highlighted joints are those whose positions are known.....	98
Figure 49: <i>Boxing</i> animation accuracy errors with Jacobian Transpose.....	101
Figure 50: <i>Boxing</i> animation accuracy errors with CCD.	102
Figure 51: <i>Boxing</i> animation accuracy errors.....	104
Figure 52: <i>Jump Kick</i> animation accuracy errors.....	105

Figure 53: <i>Playground</i> animation accuracy errors.	106
Figure 54: From left to right and top to bottom: KMA, CCD, Pseudoinverse, DLS, DLS with SVD, SDLS, PIK, SIK and TGBSIK reconstructions in frame 2,507 of the <i>Boxing</i> animation.	107
Figure 55: From left to right and top to bottom: KMA, CCD, Pseudoinverse, DLS, DLS with SVD, SDLS, PIK, SIK and TGBSIK reconstructions in frame 327 of the <i>Jump Kick</i> animation.	107
Figure 56: From left to right and top to bottom: KMA, CCD, Pseudoinverse, DLS, DLS with SVD, SDLS, PIK, SIK and TGBSIK reconstructions in frame 675 of the <i>Playground</i> animation.	108
Figure 57: The IMPULSE (Phasespace 2005) mocap system's marker configuration used for the experiment (Peinado et al. 2007).	109
Figure 58: On the left the virtual humanoid kinematical model and on the right input features for SIK [©(Vrlab 2008)].	109
Figure 59: The pose used for subject calibration.	111
Figure 60: Reconstructed poses for <i>Rising Arms</i> animation.	112
Figure 61: Reconstructed poses for <i>Head Movements</i> animation.	112
Figure 62: Reconstructed poses for <i>Stretching and Crouching</i> animation.	112
Figure 63: Reconstructed poses for <i>Picking Objects</i> animation.	112
Figure 64: Reconstructed poses for <i>Spinning Around</i> animation.	113
Figure 65: Reconstructed poses for <i>Walking Around</i> animation.	113
Figure 66: Reconstructed poses for <i>Flapping</i> animation.	113
Figure 67: Reconstructed poses for <i>Rotating Arms Backwards</i> animation.	113
Figure 68: Reconstructed poses for <i>Crossing Arms</i> animation.	114
Figure 69: Reconstructed poses for <i>Bending and Stretching Legs</i> animation.	114
Figure 70: Reconstructed poses for <i>Rolling</i> animation.	114
Figure 71: Mean kinetic pseudo-energy history delay correction for gesture spotting.	119
Figure 72: The dynamic gesture recognition procedure.	121
Figure 73: Vision measurements for depth warping and combined action recognition using a single standard camera for motion capture.	123
Figure 74: The left hand position depth warping on the camera view. The three projections are inside the red triangle, which is used for the depth calculation.	126

Figure 75: The set of dynamic gestures for the experiment. The dot represents the starting point while the arrow represents the direction and the end point.....	128
Figure 76: The continuous data flow containing 5 performances of number 7 made by subject 1 and its segmentation results.....	130
Figure 77: The continuous data flow containing 5 performances of number 7 made by subject 2 and its segmentation results.....	130
Figure 78: The three highest principal components of the gestures. The tendency to cluster can be observed. More disperse clusters need more principal components to be better defined. Note that PCA is not used for the classification.	133
Figure 79: Combined quasi-static gestures to be recognized: <i>neutral, initial, cross, crouch, left punch</i> and <i>right punch</i>	135
Figure 80: Interaction examples of subject 1 with Gaussian RGB model background: <i>neutral, initial, cross, crouch, left punch</i> and <i>right punch</i>	136
Figure 81: Interaction examples of subject 2 using chroma-key background subtraction: <i>neutral, initial, cross, crouch, left punch</i> and <i>right punch</i>	137
Figure 82: Trunk pose recognition areas in subjects 1 and 2.....	138
Figure 83: Left arm pose recognition areas in subjects 1 and 2.	138
Figure 84: Right arm pose recognition areas in subjects 1 and 2.....	139
Figure 85: Left leg pose recognition areas in subjects 1 and 2.	139
Figure 86: Right leg pose recognition areas in subjects 1 and 2.....	140

LIST OF TABLES

Table 1: Typical Human-to-Computer and Computer-to-Human communication procedures.....	2
Table 2: Masks used as path obstacles for the evaluation of tracking procedures.....	59
Table 3: Face samples considered for the evaluation of tracking procedures...	59
Table 4: Hue and saturation channel histograms in HSV color space for different skin-color face samples.....	60
Table 5: Human face and mask color model backprojections obtained from learning the chrominances of the faces' central subregion.....	61
Table 6: False positives in the scene chrominance spotting (number of pixels in the backprojections of masks). The median is obtained from a set of frames with artificial random noise added to pixel colors.....	65
Table 7: Parameters considered for the tracking performance comparison.	68
Table 8: Face tracking results. Highlighted values correspond to the best three mean maximum speeds and computation times (the best in bold font and the other two in italics).....	69
Table 9: Right wrist rotations. Note that the pronation and supination movements come from the forearm bones that orbit one another, and not the wrist joint itself. Axes are modeled according to H-Anim specification (H-Anim 2008).	88
Table 10: Full-body reconstruction methods comparison. Highlighted values correspond to the best three for each row (the best in bold font and the other two in italics).	102
Table 11: Relation between SIK controllable features and markers.....	110
Table 12: Considered parameters for gesture spotting and normalization.....	128
Table 13: Spotted gestures spatial form in the performances made by subjects 1 and 2 and their corresponding normalized shapes.	129
Table 14: Time variation in gestures performed by subjects 1 (S_1) and 2 (S_2) expressed as the standard deviation with respect to the mean in percentage.	131
Table 15: Parameters for the statistical classification comparison.....	131
Table 16: Confusion matrix of the classification using Random Forest.....	132

Table 17: Limb pose labels for combined actions building.	134
Table 18: Limb postures from the databases.	134
Table 19: Combined motor action semantic descriptions from individual limb pose combinations.	135

RESUMEN

Tradicionalmente las personas interactúan con los ordenadores mediante herramientas como el teclado y el ratón. Los ordenadores pueden detectar eventos provenientes de éstos, como pulsar y soltar botones por separado o combinados, o movimientos de ratón, y después reaccionar de acuerdo a la interpretación que se les asigne. Este procedimiento de comunicación ha sido empleado satisfactoriamente para un amplio número de aplicaciones. Sin embargo, este enfoque carece de la naturalidad de la comunicación humana cara a cara.

Este proyecto de tesis presenta un método para la captura sin marcadores en tiempo real y la interpretación automática de movimientos humanos de cuerpo entero para la interacción persona-computador.

Para la consecución de este objetivo se distinguen tres fases: (1) el seguimiento sin marcadores de cuantas más partes del cuerpo posibles, (2) la reconstrucción del esqueleto cinemático 3D que representa la postura del usuario a partir de dicho seguimiento, y (3) el reconocimiento de los patrones de movimiento con el objeto de hacer que el ordenador “entienda” la intención del usuario y después reaccione convenientemente. Estos tres procesos han de resolverse en tiempo real para así llegar a una interacción persona-computador aceptable.

La primera fase puede ser resuelta mediante cámaras que enfoquen al usuario y algoritmos de visión artificial que extraigan y hagan el seguimiento de las características relevantes del usuario en las imágenes. Este proyecto propone un método que combina probabilidades de color y flujo óptico. La segunda requiere situar el esqueleto cinemático 3D en posturas biomecánicamente posibles y ajustadas a las partes del cuerpo detectadas, considerando las posturas anteriores para así obtener movimientos suaves. Este proyecto propone un método de cinemática inversa analítica-iterativa que sitúa las partes del cuerpo secuencialmente, desde el torso hasta los miembros superiores e inferiores, teniendo en cuenta tanto los límites biomecánicos como las colisiones más relevantes. Finalmente, la última fase requiere el análisis de los rasgos significativos del movimiento, para así interpretar patrones mediante técnicas de inteligencia artificial. Este proyecto propone un método para extraer automáticamente los gestos potenciales del flujo de datos y clasificarlos, permitiendo asimismo la ejecución de acciones combinadas.

LABURPENA

Tradizionalki gizakiek ordenagailuekin teklatu eta xagua bezalako gailuen bidez interakzionatzen dute. Ordenagailuek hauengandik datozkien gertakizunak (bakarkako edo konbinaturiko botoiak sakatzea edo askatzea, edota xagua mugitzea bezalaxe) antzeman ditzakete, eta hauei dagozkien interpretazioen funtzioan erreakzionatu. Komunikazio prozedura hau arrakastatsuki erabilia izan da aplikazio anitzentzat. Hala ere, jardunbide honi gizakien arteko aurrez-aurreko komunikazioak duen naturaltasuna falta zaio.

Tesi projektu honek denbora errealean markagailurik gabeko gorputz osoko giza mugimenduen kaptura eta interpretazio automatikorako metodo bat aurkezten du giza-konputagailu interakziorako.

Helburu honen burutzapenarako hiru fase bereizten dira: (1) gorputzaren ahalik eta zati gehien markagailurik gabeko jarraipena, (2) jarraipen hauengandik gorputz-jarrera errepresentatzen duen 3D eskeletu zinematikoaren eraikitzea, eta (3) mugimendu patroien antzematea ordenagailuak gizakiaren asmoa “uler” dezan, honen ondorioz behar bezala erreakziona dezan. Hiru protzesu hauek denbora errealean lortu behar dira giza-konputagailu interakzio onargarria lortzearentzat.

Lehenengo fasea ebatzi daiteke gizakia begira dauden kamarak eta hauekin batera irudietatik haren ezaugarri esanguratsuenak antzeman eta jarraitzen dituzten bisio artifizialeko algoritmoen bidez. Projektu honek kolore probabilitatea eta fluxu optikoa konbinatzen dituen metodo bat aurkezten du. Bigarrenak 3D eskeletu zinematiko biomekanikoki posible eta detektaturiko gorputz zatietara egokitutako posturretan kokatzea behar du, aurreko posturak kontutan harturik mugimendu suabeak lortu ahal izateko. Projektu honek gorputz zatiak, enborratik hasita goi eta behe gorputzadarretaraino, sekuentzialki kokatzen dituen alderantzizko zinematika analitiko-iteratibo metodo bat proposatzen du, biomekanikar mugak eta kolisio nabarmenenak kontutan harturik. Bukatzeko, azken faseak mugimenduaren ezaugarri esanguratsuenen analisia behar du, adimen artifizialeko tekniken bidez patroiak interpretatu ahal izateko. Projektu honek datu fluxutik zeinu potentzialak automatikoki antzeman eta klasifikatzeko metodo bat proposatzen du, halaber konbinaturiko akzioen burutzapena onartuz.

ABSTRACT

Typically, people interact with computers by means of devices such as the keyboard and the mouse. Computers can detect the events coming from these devices, such as pushing down or releasing isolated or combined keyboard and mouse buttons, or the mouse motions, and then react according to the interpretation assigned to them. This communication procedure has been satisfactorily used for a wide range of applications. However, this approach lacks naturalness with respect to face-to-face human communication.

This thesis project presents a method for markerless real-time capture and automatic interpretation of full-body human movements for human-computer interaction (HCI).

Three stages can be distinguished in order to reach this objective: (1) the markerless tracking of as many of the user's body parts as possible, (2) the reconstruction of the kinematical 3D skeleton that represents the user's pose from the tracked body parts, and (3) the recognition of the movement patterns in order to make the computer "understand" the user's will and then react according to it. These three processes must be solved in real time in order to attain a satisfactory HCI.

The first stage can be solved by means of cameras focusing on the user and computer vision algorithms that extract and track the user's relevant characteristics from the images. This project proposes a method that combines color probabilities and optical flow. The second one requires to situate the kinematical 3D skeleton in plausible biomechanical poses fitted to the detected body parts, considering previous poses in order to obtain smooth motions. This project proposes an analytic-iterative inverse kinematics method that situates the body parts sequentially, from the torso to the upper and lower limbs, taking into account the biomechanical limits and the most relevant collisions. Finally, the last stage requires analyzing which are the significant features of motion in order to interpret patterns with artificial intelligence techniques. This project proposes a method to automatically extract potential gestures from the data flow and then label them, allowing the performance of combined actions.

CHAPTER 1

INTRODUCTION

1.1 MOTIVATION

For centuries Humanity has been trying to build machines capable of emulating human behaviors in order to make them work, or even fight, instead of real people, or simply for entertainment. Even though some advances have been achieved in the field of artificial intelligence, especially since the appearance of computers, technology is still far away from satisfactorily reproducing human intelligence. This occurs mainly because we still do not really understand what cognition is, nor do we know how to emulate its capabilities.

Interpersonal communication requires the use of human senses in which some regions of the brain take part in order to interpret data coming from sensory organs. Thus, research on communication moves towards the final objective of solving the mysteries of the human mind.

According to the work of Shannon and Weaver (1949) there are six basic elements that conform an act of communication (Figure 1):

- **Information Source:** which emits the message.
- **Transmitter:** which encodes the message into coded signals.
- **Channel:** the physical way in which the signals are transmitted.
- **Noise:** the dysfunctional factor that distorts the signal and therefore may corrupt the message.
- **Receiver:** which decodes the message from the signal.
- **Destination:** which gets the message.

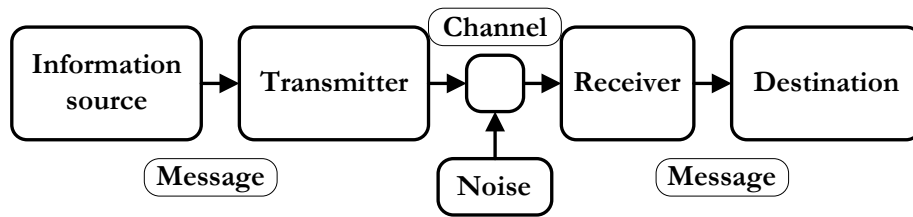


Figure 1: Shannon and Weaver's communication model.

In human-computer interaction (HCI), according to this scheme, the typical ways in which the user communicates his/her will to the computer, and vice versa, the computer communicates "its will" to the user, are the following:

	Human-to-Computer	Computer-to-Human
Source	User	Application software
Transmitter	Keyboard and mouse	Monitor and speakers
Channel	Hardware circuits	Visible light waves and the atmosphere
Noise	From mechanical and electrical devices	From the monitor, eyes, speakers and ears, or other light or sound sources
Receiver	Operating System (OS) and application software	User's sight and hearing senses
Destination	Application software	User

Table 1: Typical Human-to-Computer and Computer-to-Human communication procedures.

This approach has been, and continues to be, widely and satisfactorily used for a huge number of applications, but it is far from the naturalness of face-to-face human communication where human senses are more involved.

Consequently, there has recently been an increased interest in multimodal interaction. Multimodality seamlessly combines multiple modes of interfacing with the computer. Currently these modes go from visual (computer vision), to voice (speech recognition and synthesis), to touch

(haptics), which increase both the level of interactivity and the complexity in their software implementation, with respect to more traditional devices.

Among these modes, nonverbal communication, such as that obtained from facial expressions, body postures and motor actions, represents most of the transferred information in a human face-to-face interaction. Therefore, work on this area is of vital interest for our purpose. In this type of communication we can distinguish four abstraction layers, from lower to higher level of abstraction:

1. **Motion Capture:** This layer refers to the extraction and tracking of the user's features from the images.
2. **Pose Reconstruction:** This refers to the estimation of the user's body configuration from the tracked image-features.
3. **Motor Action Recognition:** This refers to the semantic description of the body pose sequences through time.
4. **Activity Interpretation:** This refers to the semantic description of complex psychomotor tasks involving the recognized actions and the interaction of the user with the context.

The work presented in this thesis project is circumscribed to the first three layers. Therefore, the considered reactions coming from the computer will just be in the form of feedback graphics and messages on the monitor showing what the computer has "understood" with respect to the user's will.

1.1.1 HUMAN MOTION CAPTURE

Human motion can be obtained by means of many different devices. Currently, marker-based systems are the most popular motion capture (mocap) systems. Markers are placed all around the body and the relative positions with respect to the body parts to which they are attached are measured. This way, when markers are tracked while they move, it is possible to determine the motions of the user's different body parts. Captured results are usually displayed with a virtual character that imitates the user's movements.

Depending on the mocap system, the tracked data can be the marker positions, orientations, accelerations, etc. We can distinguish mainly among optical, magnetic, mechanical and inertial mocap systems (Figure 2).



Figure 2: From left to right and top to bottom: Optical [© (Phasespace 2005)], magnetic [© (Ascension 2004)], mechanical [© (DoMotion 2004)] and inertial [© (XSens 2007)] marker-based motion capture systems.

Optical systems use a set of cameras placed around the user which track the positions of reflective, luminous or photosensitive (semi)spherical markers. They are very popular due to they obtain the highest level of accuracy, even with the quickest human movements. Their main drawback comes from the marker occlusions that can occur during the user's motion. This can lead to incorrect posture reconstructions which can be minimized, in certain cases, using extra information such as biomechanical joint rotation limits or estimations of the occluded marker positions. Magnetic systems get the positions and orientations from the relative magnetic flux of three orthogonal coils on both the transmitter and each of the receivers. Their main advantage is that markers cannot be occluded, but they are very sensitive to the presence of metal objects and electrical sources. Mechanical systems correspond to skeletal-like structures that the user wears, from which his/her joint angles can be tracked directly. They are free-of-occlusion and have an unlimited capture volume. Finally, inertial systems are able to capture the positions, orientation and velocities of markers in large capture areas, also free-of-occlusion.

All these approaches can track human movements in real time and therefore are appropriate for HCI, but unfortunately their high price and their cumbersome use makes them prohibitive for home-users. Nevertheless, more recently, new devices that include accelerometers have appeared, such as the Nintendo Wii (2006) video game console remote, and the Apple iPhone (2007) and Sony Ericsson W910i (2007) mobile phones, which can be used to easily obtain the user's simple motor gestures at an affordable cost (Figure 3).



Figure 3: The Nintendo Wii remote controller [© (Nintendo 2006)], the Apple iPhone [© (Apple 2007)] and the Sony Ericsson W910i [© (Sony-Ericsson 2007)] respectively.

But recently there have been remarkable advances in markerless motion capture which, in the near future, may render marker-based systems obsolete for many HCI applications due to their lower cost and easier use. In this approach, human movements are intended to be captured directly from images obtained by cameras. An image can be represented mathematically as a hypermatrix $n_1 \times n_2 \times n_3$, where n_1 and n_2 correspond to its resolution (therefore $n_1 \times n_2$ is the number of pixels), and n_3 is the number of channels corresponding to each pixel. If only a single channel is available it normally corresponds to the intensity in grey-level. If there are three, they usually are color values, which can have different meaning depending on the color space used: RGB, normalized-RGB (nRGB), HSV, HLS, YCrCb, Lab, Luv, XYZ, xyY, etc. (Pascale 2003). Other channels can be added which can have additional information, among which excels the *depth of the pixel* (Figure 4). Depths of the observed scene can be achievable with new technologies that have recently emerged, such as the stereo (Birchfield and Tomasi 1999) and Time-of-Flight (ToF) cameras (May et al. 2006). Computer vision algorithms are intended to make use of the temporal successions of these hypermatrices coming from single or multi-camera systems in order to get the user's movements and map them to a virtual character.



Figure 4: On the left a 2D image and on the right its corresponding depth map using a stereo camera [© (PointGrey 2007)].

1.1.2 HUMAN POSE RECONSTRUCTION

The poses of the virtual character that imitates the user's movements are usually defined with an underlying multibody skeleton conformed of joints and segments (Figure 5).

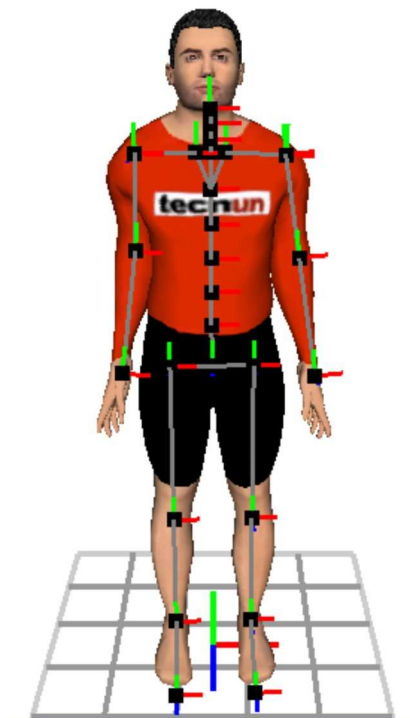


Figure 5: Skeletal multibody structure of a virtual character [©(MIRALab 2008)].

These joints are normally organized as a hierarchical set, where each joint controls the position and orientation of a body segment. The root joint is usually situated in the middle of the pelvis and the rest have only one parent joint. The degrees of freedom (DoF) of this mechanism are normally represented with the *XYZ* displacements of the root joint with respect to the absolute coordinate system and the relative rotation angles of each joint with respect to its parent (absolute coordinate system for the root joint). The number of rotation angles goes from 1 to 3 depending on the biomechanical physiology of the joint.

The problem of human pose reconstruction consists in determining which are the values corresponding to the DoF of this skeleton in order to fit the data obtained by the mocap system. This process is called inverse kinematics and can be classified into *exactly-guided*, *over-guided* and *under-guided*, depending on the level of redundancy of the DoF with respect to the tracked body features (Ausejo 2006). Exactly-guided kinematic problems are those in which the tracked features can determine directly the DoF. Over-guided kinematic problems are those in which there is a redundancy in the tracked features with respect to the DoF. Finally, under-guided kinematic problems refer to the cases in which the mechanism has redundant DoF with respect to the tracked features. The key points of this problem, apart from the fitting, are to obtain biomechanically plausible poses and smooth motions from frame to frame.

1.1.3 HUMAN MOTOR ACTION RECOGNITION

The motion data being captured in real time consists in a continuous data flow evolving in time according to the movements being performed by the user. These data can be the evolution of the reconstructed virtual character's DoF or other possible features of motion. Humans can communicate messages through isolated postures or even through a sequential set of them, with no-meaningful transition movements in the in-betweens.

There are certain issues to be undertaken to fulfill a correct communication through this channel. Firstly, it is necessary to determine which are the motion-features that define a certain semantic motor action. For example, the position of a foot is not meaningful for a "hand shaking" action, while the position of that hand is. If we consider, in this example, also the foot position for a further action classification, the system may fail if the user's foot position is far from the learned pattern, even though the hand traces the same trajectory as in the reference. Therefore the semantic importance of each tracked data must be measured in order to use only the data that is most relevant for recognition.

Along with this meaningful motion-feature extraction, it is necessary to determine which are the starting and ending instants of the semantic gestures. Captured data evolve continuously and the semantic descriptions of motion may vary meanwhile. The data evolution must be analyzed in order to force the system make a decision on which frames compose a potential gesture and which do not. For instance, isolated meaningful postures will be intended to keep static for a certain time while the in-between transitions will not, and the starting instants of meaningful posture sequences may present a distinguishable acceleration variation from the previous motion states.

Finally, the potential gestures must be catalogued according to the system's knowledge. This knowledge is a database composed of a set of learned patterns, where the semantic meaning of these patterns comes through labels describing them. This way, during the capture, new performances can be labeled with the use of statistical classification (Jain et al. 2000) with respect to the database.

Depending on the application, it may be interesting to have several learned performances for each label, as they can vary from person to person or even for the same person trying to perform the same gesture. Thus, the system can try to distinguish the different styles of new gesture performances, or simply make the system more robust for a general gesture classification as it "knows" more variations for the same gesture. On the other hand, having more performances for each gesture implies having a more cumbersome "teaching process".

1.2 CONTRIBUTIONS AND THESIS PROJECT OUTLINE

The goal of this thesis project is to capture and reconstruct full-body human movements, without the use of markers attached to the body, and recognize the undertaken meaningful motor actions for HCI applications. This thesis project presents the following contributions:

- A real-time markerless 2D strategy capable of tracking with low noise specific human body parts in movements performed at high speed. It can handle different skin-colors and clothing. This 2D strategy can be extended to 3D with the use of depth-sensing cameras.
- A real-time markerless strategy to track full-body movements for a pseudo 3D motion reconstruction using a single standard camera considering also the possible depth variation of the tracked body parts, and full 3D using a single depth-sensing camera. The obtained postures

are of sufficient quality for a subsequent motor action recognition procedure.

- A real-time human pose reconstruction procedure for under-guided kinematic problems that considers the possible relative rotations of all vertebrae and the shoulder complex, but ignores those of the fingers and toes. The minimum considered input data are the positions of the pelvis, head, hands and feet, while the maximum are both the positions and orientations of those body parts along with the positions of elbows and knees.
- A simple and efficient method for modeling complex biomechanical joint limits using only a few biomechanical measurements. A real-time strategy for setting joints within biomechanical limits is also presented.
- Real-time collision-avoidance strategies to prevent elbow-torso, wrist-torso and foot-floor interpenetrations. The mesh-surfaces of the graphics are considered for penetration measurement.
- A procedure for the automatic determination of the starting and ending time instants in which a semantic motor action occurs, from a continuous motion data flow while it is being captured in real time.
- A method for the recognition of full-body postures and gestures, in which combined motor actions can also be undertaken having their separate gestures classified accordingly, in real time for HCI applications.

The content of this document is divided into six chapters. This first chapter has presented the motivations for this work. Next chapter presents the most relevant methods used by other authors for vision based human motion reconstruction and motor action recognition. Chapters 3, 4 and 5 correspond to the proposed methods for body parts motion tracking, full-body pose reconstruction and motor action classification respectively, which in conjunction, allow the interaction with the computer through human motion. Chapter 3 and 4 contain experimental results for motion tracking and pose reconstruction, respectively, in which the proposed methods are compared with other alternatives. Chapter 5 contains experimental results on motor gesture and combined motor action labeling in which the proposed methods for tracking, reconstruction and recognition are merged to demonstrate their applicability for HCI. Finally, chapter 6 presents the conclusions derived from this work and proposes future research lines. Additionally, the publications generated during this thesis project work are included. The publication list contains a reference to an Internet video showing the obtained results.

CHAPTER 2

RELATED WORK AND TAXONOMIES

More than one century ago, Étienne-Jules Marey theorized that analyzing the locomotion of animals as mechanisms Humanity would be able to attain goals not known until that date, such as flying (Marey 1873). He opted for the imitation of animal movements such as those of birds to achieve it. Nevertheless, the Wright brothers demonstrated that it is not necessary to follow this strategy, and they even opened up a new research line in human flight more appropriate for the comfort of “flying” humans compared to that of birds.

Similarly, the task of motion observation and recognition has been studied through many different approaches which do not necessarily come from the imitation of the practically unknown human visual perception. Moreover, some of the obtained results transgress the capabilities of human observation. For example, people are not able to measure the anthropometry of the observed people and neither to reconstruct their view-independent 3D shape as computers do. However, human image understanding still outperforms widely that of the computer in many aspects, such as image segmentation and labeling, part differentiation in multibody objects or scene interpretation.

The field of psychology has invested many efforts in order to advance in the human perception understanding. In the 1930s the Gestalt of the Berlin school (Koffka 1935) stated that our senses perceive figures and whole forms instead of a collection of atoms, i.e., the whole is perceived differently from the sum of its parts, and developed six laws that govern human perception:

- **Law of Proximity:** elements that are closer together will be perceived as a coherent object.
- **Law of Similarity:** elements that look similar will be perceived as part of the same form.
- **Law of Good Continuation:** humans tend to continue contours whenever the elements of the pattern establish an implied direction.
- **Law of Closure:** humans tend to enclose a space by completing a contour and ignoring gaps in the figure.
- **Law of Good Form:** elements tend to be grouped together if they are parts of a good form pattern. Here, good means symmetrical, simple, and regular.
- **Law of Figure/Ground:** a stimulus will be perceived as separate from its ground.

More recently, Johansson (1973; 1975) studied how motor actions are recognized by means of moving light displays (MLDs). He attached light spots to the joints of human actors and filmed them moving in the dark. Then, observers who only perceived a few light spots and did not previously know that an actor was wearing them, were able to recognize human figures if lights were in motion, but not when they were still. Two theories arose from these results: (1) observers recognized human figures by previously reconstructing the body structure from the MLDs, and (2) recognition was attained directly from the MLDs. On the other hand, Cutting and Proffitt (1982) found that the relative motion (movement of an element with respect to other elements) is more revealing for the understanding of a motion and the recognition of an object than its common motion (global movement of the object relative to the observer).

All these results obtained from the psychology field can be and have been taken into account for the design of computer vision tools for image understanding. Consequently, in this chapter the most up to date remarkable strategies for the human motion reconstruction (motion capture and pose reconstruction) and motor action recognition, along with their applications besides HCI, are presented. This literature revision derives from the most cited articles and the taxonomies proposed in the surveys (Aggarwal and Cai 1999; Cedras and Shah 1995; Gavril 1999; Moeslund and Granum 2001; Moeslund et al. 2006; Mündermann et al. 2006; Poppe 2007; Wang et al. 2003) and the thesis of González (2004), along with the found most recent techniques and initiatives in these fields.

2.1 GENERAL PROBLEMS AND ASSUMPTIONS

There are several problems to be addressed in order to make the computers “see” and “understand” human motion. The approaches presented in this chapter try to solve them in different ways considering certain assumptions related to the movements, environment and subject being recorded, but some of them are not even feasible yet. The more assumptions are made the easier the tracking procedure will be but the less general will be its applicability. Next, the general problems and the typical assumptions made in the literature methods, are presented:

P1 - Problems in Motion Capture

- Choosing suitable image-features for tracking.
- Discerning multiple subjects and objects moving in a scene.
- Ignoring changeable background and focusing only on the subjects.
- Projecting the image-features obtained from multiple views to the same spatial reference, when they have been recorded in different spatial coordinates.
- Matching the image-features obtained from different views and along time corresponding to the same real object.
- Distinguishing clothes with similar color to the background.
- Extracting body parts from loose-fitting clothes.

P2 - Problems in Pose Reconstruction

- Searching for the optimal posture in the high-dimensional parameter space, knowing that there is a highly nonlinear relationship between the image and model projections similarity and the posture parameters.
- Handling with occlusions (self-occlusions and occlusions with other subjects and objects) and subsequent criteria to stop and restart the tracking of body parts.

P3 - Problems in Motor Action Recognition

- Identifying and matching application independent features of motion along time for motor action recognition.
- Dealing with spatial and time scale variations within classes of movement patterns in both learning and matching methods.

A1 - Assumptions Related to Movements

- The whole body of the subject is in the workspace.
- There is no camera motion or it moves with constant speed.
- There is only one person in the workspace.
- The user faces the camera all the time.
- The movements are parallel to the camera-plane.
- There is no occlusion.
- Movements are slow and continuous.
- Only one or few limbs are being moved.
- The user's motion pattern is already known.
- The user moves on a flat ground plane.

A2 - Assumptions Related to the Environment

- The lighting keeps constant.
- The background keeps static.
- The background is uniform.
- The camera parameters are known.
- Special hardware (for example, a 3D camera or a multi-camera system) is used.

A3 - Assumptions Related to the Subject

- The start pose of the subject is already known.
- The subject's anthropometry and shape is already known.
- Markers are placed on the subject.
- The subject wears special colored clothes.
- The subject wears tight-fitting clothes.

2.2 APPLICATIONS

Besides the HCI applications in which the developments of this thesis project are circumscribed, there are other that can be attained from the research in motion capture and motor action recognition. The *robustness* requirement in these applications refers to the assumptions mentioned in section 2.1. The less assumptions are imposed to the system the more robustness is required. Mainly three major application types can be distinguished:

1. Surveillance

- *Description:* One or more subjects are being tracked over time and possibly monitored for special actions.
- *Requirements:* High robustness and real-time processing in the motion reconstruction and motor action recognition algorithms.
- *Examples:* Access control, parking lots, supermarkets, department stores, vending machines, automated teller machines, traffic, detect possible failures in an automated line, obstacle avoidance of moving objects for robots and satellite monitoring of weather disturbance.

2. Control

- *Description:* Captured motion is used to provide controlling functionalities. Therefore HCI is included here.
- *Requirements:* Median accuracy in motion reconstruction and real-time processing in both motion reconstruction and motor action recognition algorithms. In applications embedded in noisy backgrounds also high robustness is required.
- *Examples:* Interactive virtual worlds, videogames, virtual studios, character animation, teleconferencing, social interfaces, sign-language translation, gesture driven control, signalling in high-noise environments such as airports and factories, virtual clothing, virtual haircuts, virtual shaving, virtual beards, lipreading and visual phones.

3. Analysis

- *Description:* Captured motion is used for biomechanical studies.
- *Requirements:* High accuracy in motion reconstruction.
- *Examples:* Content-based indexing of sports video footage, personalized training in golf, tennis, swimming, etc., choreography of

dance and ballet, clinical studies of orthopaedic patients, study of left ventricular motion, assessment of the locomotion of patients with cerebral palsy, progression of neuromuscular disorders, kinesiological analysis, ergonomic designs and even very low bit-rate video compression.

2.3 CAMERA PARAMETERS AND MULTI-CAMERA SYSTEMS

Observing a scene from multiple perspectives is the best and direct way to solve ambiguities in the matching of the virtual character to subject images, but further challenges arise. Body features are recorded in different spatial coordinates and therefore they must be adjusted to the same spatial reference before matching is performed. This can be solved by calibrating the cameras. Actually, camera calibration establishes a relation between the view and the real 3D world. Therefore the common relation among cameras comes from sharing the same 3D world observed from the different views. Camera calibration consists on establishing the *intrinsic* and *extrinsic camera parameters*. Intrinsic parameters are the attributes that affect the image, whereas extrinsic parameters are the camera position and orientation in the real world.

Usually, cameras used for markerless motion capture are basically composed of a CCD and lens, and are mathematically formulated with the *pinhole camera model* (Figure 6). The intrinsic parameters in this model include the image center or principal point, focal length, aspect ratio or skew, scaling factor, and lens distortion (pin-cushion effect). Some methods to determine intrinsic and extrinsic parameters can be found in (Svoboda et al. 2005; Tsai 1986; Zhang 2000). See (Hartley and Zisserman 2000) for detailed explanations on multi-view geometry.

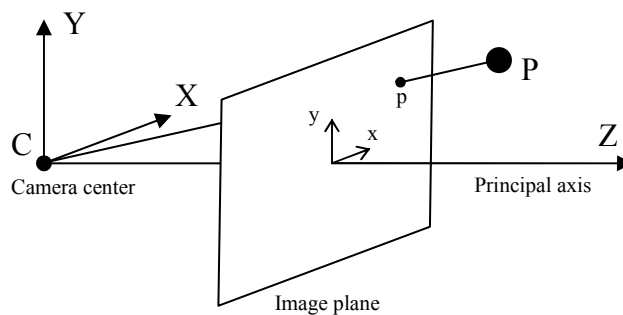


Figure 6: The pinhole camera model geometry.

Apart from calibrating the cameras it is important in multi-camera systems to synchronize the image grabbing, as the body features must be observed at the same time instant in order to obtain a correct matching among views. This is usually achieved by hardware, in which electric pulses emitted by the network or an external trigger specifies when do cameras grab images.

Furthermore, depending on the strategy used for markerless motion capture it may also be important to make all the cameras “see” the scene with the same color temperature, which refers to the relative warmth or coolness of white light. This can be solved with the *white balance* process that consists on focusing the images to a white reference in order to adjust the color temperature. Additionally, some methods, like the *disparity map* for stereo cameras (Birchfield and Tomasi 1999), may require to observe the same grey-level values of the scene in order to estimate correctly the image depths. This can be achieved by adjusting other camera parameters such as the shutter, gain, brightness, sharpness and gamma.

2.4 HUMAN MOTION RECONSTRUCTION SURVEY

The general procedure derived from the literature to solve the problem of reconstructing the human motion (i.e., motion capture + pose reconstruction) without the use of markers attached to the user’s body is the following: (1) extract some features of the images corresponding to the user’s projections, and (2) determine their matching between consecutive frames, along with their mapping to the body parts, or in other words, *track* them.

The optimal procedure for human tracking would be to directly capture the body joint positions and orientations, but this is not feasible in general in the markerless approach, as joints do not have any special visual characteristic that makes them more distinguishable from other body features. Therefore other alternatives have been pursued in the literature as will be shown next.

2.4.1 IMAGE-FEATURES EXTRACTION

As stated in section 1.1.1 an image can be represented mathematically as an hypermatrix of pixel positions containing their grey-level or color values (and even depth in stereo and ToF cameras). These numbers can be processed in order to obtain certain image-features that have certain characteristics, such as brightness, contrast and size, which outline from the rest of pixels. According to the literature these can be (ordered from more simple to more complex): *feature points*, *edges*, *blobs*, *silhouettes* and *visual hull* (Figure 7). In case they correspond to the projections of the subject they can be used for human motion reconstruction.

There is a trade-off between image-feature complexity and tracking efficiency. The simpler the extracted image-features are, the relatively more complex is to track them. For example, it is easier to track blobs than isolated feature points, as they contain more characteristics that define their condition, which can help in motion correspondence. Indeed, these image-features can be combined in order to have more input data for the body parts tracking and consequently for the pose reconstruction.

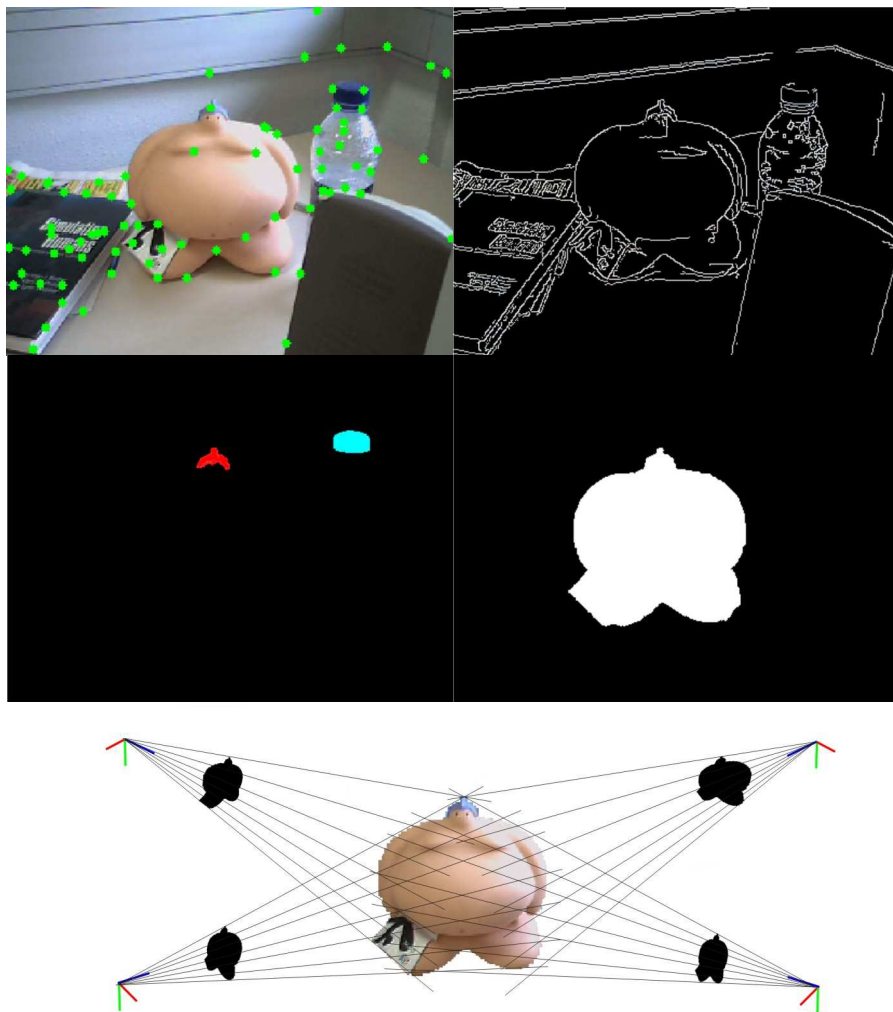


Figure 7: From left to right and top to bottom: (a) Feature points, (b) edges, (c) the blobs of the doll's hair and the bottle's top, (d) the doll's silhouette, and (e) the doll's visual hull obtained from the silhouette backprojections of multiple views.

2.4.1.1 *Feature Points*

Feature points, also known as *interest points* or simply *features*, are points in the image which have a well-defined position, like corners, edge endings, curvature maxima of curved lines or distinguishable isolated points in the observed 3D scene. The grey-level information of the image is used to locate them by evaluating the local image regions with a high degree of variation in all directions. See (Moreels and Perona 2007; Schmid et al. 2000; Zuliani et al. 2004) for further information about existing feature points detection methods and their evaluation.

2.4.1.2 *Edges*

Edges are image-features that reflect inherent properties of the observed 3D scene, such as the discontinuities in its physical, photometrical and geometrical properties. They typically correspond to borders, and are therefore useful for segmentation and identification of objects. In order to locate edges in the images significant variations of the grey-level values are detected. Hence, image derivatives are computed which require the smoothing of images as derivatives are sensitive to their inherent noise. See (Basu 2002; Fernández-García et al. 2008; Ziou and Tabbone 1998) for a survey on edge detection approaches and their comparison.

2.4.1.3 *Blobs*

Blobs refer to compact image regions made of connected pixels that share common characteristics such as color, depth and/or motion, and therefore can be obtained from different sources and criteria. In markerless human motion capture blobs usually correspond to body parts of the subject like, for example, hands, feet and head. There are many approaches that use blobs in order to capture of human motion, e.g., (Argyros and Lourakis 2006; Date et al. 2004; Varona et al. 2005; Wren et al. 1997).

2.4.1.4 *Silhouettes*

Silhouettes can be considered as blobs corresponding to the 2D projection of the subject's full-body shape. Due to the non-rigid nature of human motion, they are usually obtained using background subtraction techniques, in which pixel color values are compared directly with respect to a learned background color model. This background model can be obtained from the observation of the scene without the subject in it, and it can also be updated along time in order to handle background changes. One of the toughest issue at this level, even with static cameras and diffuse illumination, is the suppression of the

shadows cast by the user in order to achieve clean silhouettes. See (Piccardi 2004) for a review in background subtraction techniques.

2.4.1.5 Visual Hull

As stated by Laurentini (1994), the visual hull of an object is the closest approximation of that object that can be obtained with the volume intersection approach. This way, in a multi-camera mocap system the 3D shape of the subject can be reconstructed with the intersection of 3D regions generated by the inverse projections of the subject's silhouettes. The more viewpoints are available the more accurate the subject's inferred visual hull will be. This is usually obtained as a voxel representation which can be colored, and thus resulting in a more realistic reconstruction of the moving subject. Additionally, the voxel coloring can be used to aid in the subject's tracking. See (Cheung et al. 2005; Slabaugh et al. 2001) for a review on these techniques.

2.4.2 HUMAN BODY PARTS AND HUMAN SHAPE TRACKING

Once image-features that correspond to the human body parts or the whole human shape have been detected, in order to establish their displacements it is necessary to match them between consecutive frames. Normally, as many image-features may be obtained at the same time, ambiguities for their matching may arise. Therefore, well defined constraints must be imposed to find a unique match. Typically small image motion between successive frames is required, and hence high-framerate cameras are needed.

According to Yilmaz et al. (2006) the motion correspondence problem of objects can be catalogued into three main approaches (Figure 8): *point tracking*, *kernel tracking* and *silhouette tracking*. In the case of human motion capture, these "objects" may correspond to the projections of separate body parts or even the human figure as a whole (without considering explicitly its body parts).

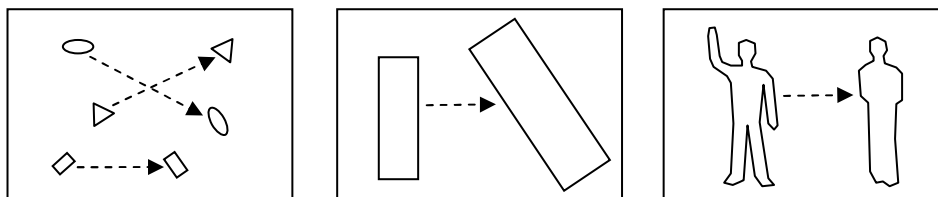


Figure 8: From left to right: point tracking, kernel tracking and silhouette tracking approaches.

Many object tracking methods, independently from the approach of this taxonomy in which they are circumscribed, share a basic low-level motion correspondence technique: the *optical flow*.

2.4.2.1 Optical Flow

This low-level tracking method consists on computing the displacement of pixels between frames in grey-level images (Figure 9). Nevertheless, recently also color images have been used directly to compute optical flow (Andrews and Lovell 2003). In order to determine the pixel displacement, intensity constancy from frame to frame is supposed. Its benefit is that even in the presence of partial occlusion, some of the features being tracked with this technique remain visible. However, in some conditions it only allows the precise computation of the normal flow (parallel to the image gradient). All the same, feature points do not suffer from the aperture problem and therefore are often tracked using optical flow. On the other hand, background pixels around object boundaries might have a non-zero flow value. See (Baker et al. 2007) for a recent revision and comparison of the existing optical flow techniques.

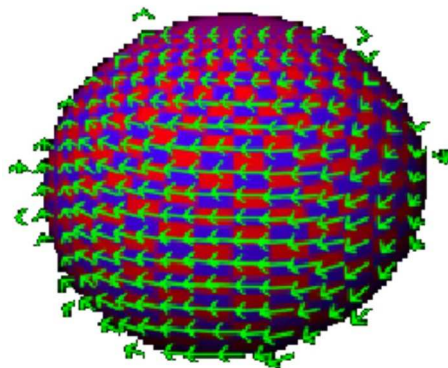


Figure 9: Pyramidal implementation of the Lucas-Kanade optical flow (Bouguet 1999; Lucas and Kanade 1981) applied to a rotating sphere.

2.4.2.2 Point Tracking

In this object tracking approach objects are represented as points across frames (e.g., the centroid of the object). The motion correspondence is based on the previous object state which can include position and motion. Two subcategories can be distinguished:

- **Deterministic methods:** These methods define a cost of associating each object in the image in the previous frame to a single object in the

current frame using a set of motion constraints. These constraints include proximity, maximum velocity, small velocity change, common motion and rigidity of points. Representative works following this approach are the Modified Greedy Exchange (MGE) (Salari and Sethi 1990) and Greedy Optimal Assignment (GOA) (Veenman et al. 2001) trackers.

- **Statistical methods:** These methods take into account the uncertainties coming from the video sensor measurements and the object model properties (such as position, velocity and acceleration) during the object state estimation. This way, random perturbations coming from the video sensor measurements can be handled. Representative works following this approach are the Kalman filter (Broida and Chellappa 1986; Kalman 1960), Joint Probabilistic Data Association Filter (JPDAF) (Bar-Shalom and Foreman 1988), and Probabilistic Multi-Hypothesis Tracking (PMHT) (Streit and Luginbuhl 1994) trackers.

2.4.2.3 Kernel Tracking

In this object tracking approach objects are represented with a kernel that refers to its shape and appearance. The motion correspondence is usually represented as a parametric transformation such as translation, rotation and affine. Two broad subcategories can be distinguished:

- **Template based:** In the case a single object is searched, these methods compute the position of the object template, defined in the previous frame, by a similarity measure, e.g, cross correlation. On the other hand, in the case multiple objects have to be tracked, the background and all the moving objects are explicitly tracked. Representative works following this approach are Mean-shift (Comaniciu et al. 2003), CamShift (Continuously Adaptative Mean-Shift) (Bradski 1998), Kanade-Lucas-Tomasi (KLT) (Shi and Tomasi 1994), and Layering (Tao et al. 2002).
- **Multi-view based:** These methods learn different views of the object offline in order to be able to handle dramatic object view changes, which make the appearance model no longer valid. Representative works following this approach are the Eigentracking (Black and Jepson 1998) and the Support Vector Machine (SVM) tracker (Avidan 2001; Shawe-Taylor and Cristianini 2000).

2.4.2.4 *Silhouette Tracking*

In this object tracking approach objects are represented with their silhouettes. The motion correspondence is obtained by estimating the object region in each frame. Two broad subcategories can be distinguished:

- **Contour evolution:** These methods perform the tracking by evolving an initial contour in the previous frame to its new position in the current frame. This evolution requires that at least some part of the object in the current frame overlap with its occupied region in the previous frame. Representative works following this approach are the state space models, such as the well-known Condensation (CONDitional DENSity propagATIION) or particle filtering algorithm (Isard and Blake 1998), variational methods (Bertalmio et al. 2000) and heuristic methods (Ronfard 1994).
- **Shape matching:** These methods are similar to those of template matching. In this case the search is performed by computing the similarity of the object with respect to that obtained from the hypothesized object silhouette based on previous frame. The silhouette is supposed to only translate from frame to frame, so its nonrigid motion is not handled explicitly. Representative works following this approach are the Hausdorff metric (Huttenlocher et al. 1993), Hough transform (Sato and Aggarwal 2004) and histogram matching (Kang et al. 2004b).

2.4.2.5 *Occlusions*

During the human motion tracking there are many ambiguous situations that may arise because of occlusions. The best way to avoid them is to use more views, but the complexity in the system increases because the multi-view matching of the tracked features must be done, also an appropriate selection of camera positions, and also because of the higher computational cost. Nevertheless, occlusions may still occur and they should be handled. For example, if two hands represented by blobs are being tracked, it may happen that they merge at some time and afterwards they split again into two blobs. In that case the computer will have a problem to discern correctly which blob corresponds to which hand if this situation is not handled specifically. Generally there are three types of occlusions in human motion capture:

- **Self-occlusion:** One body part of the subject occludes another.

- **Inter-object occlusion:** Two objects (or subjects) being tracked in the scene occlude each other.
- **Occlusion with the background:** A structure not of interest, i.e., background, occludes the tracked object (or subject).

In the work of (Gabriel et al. 2003) the techniques that handle occlusions are divided in two categories: *merge-split* and *straight-through* approaches.

- **Merge-split:** These methods handle occlusion situations in which the objects being tracked as N independent blobs encapsulate into a new composite blob which afterwards may split into up to N blobs again. The merged blob is tracked as any other blob in scene, even if it contains more than one object at the same time. When a split occurs the problem is to identify the object splitting from the group. Representative works on this approach are (Haritaoglu et al. 2000; McKenna et al. 2000).
- **Straight-through:** These methods handle occlusion situations in which the N independent blobs being tracked do not merge into a new blob, i.e., each blob always contains one object. Most systems rely on the appearance features of objects to classify any pixel in the vicinity of the occlusion region. The relative depth between occluding objects is a often used feature for this purpose. Representative works on this approach are (Elgammal and Davis 2001; Khan and Shah 2000).

2.4.3 HUMAN POSE RECONSTRUCTION AND MULTIBODY TRACKING

A human body is modeled as a multibody articulated mechanism, and therefore many body parts have to be tracked at the same time, which must maintain coherence according to human motion. As it has been seen in the previous section (2.4.2), there are different ways to track objects from the extracted image-features, but they are not usually applicable to track every single body part, as these are not easily distinguishable from each other. In addition, the *silhouette tracking* is not capable of determining the exact position and orientation of each body part. The process that fulfills this objective is *pose reconstruction* which is directly related to the tracking approach, i.e., the human pose estimation process depends on the considered tracking strategy. Three types of humanoids have been used in the literature for pose reconstruction which vary in their level of complexity (Figure 10):

- **Stick-figure:** It consists only of a combination of line segments linked by joints. Despite its simplicity, it certainly includes structure

information resembling the human skeleton, which can be used to attain biomechanically consistent human poses.

- **Contour-figure:** In this case human body segments are analogous to 2D ribbons or blobs. This way, the projections of the human figure can be exploited for a better pose estimation. Its disadvantage comes from its restriction to the camera's angle.
- **Volumetric-figure:** This model uses from simple shape primitives (generalized cones, elliptical cylinders, spheres, superquadrics) to surfaces (polygonal mesh, sub-division surface) to attempt to describe the human body in 3D. The more complex the 3D model is, the better results may be obtained but the more parameters, and hence computation time, is required during the matching process.

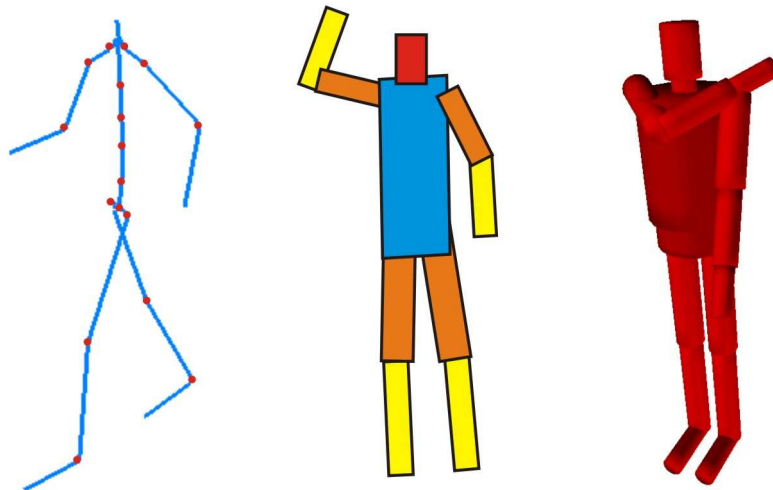


Figure 10: Humanoids in the form of stick-, contour- and volumetric-figures [© (HUMODAN 2005)] respectively.

There are two strategies to solve the image-feature/humanoid mapping problem: *bottom-up* and *top-down*. The former reconstructs human poses trying to deduce the positions and orientations of joints directly from the observed image-features. On the contrary, the latter makes use of a predefined kinematical model explicitly in order to establish the image-feature correspondence between consecutive frames, i.e., the tracking. They can also be combined at various levels in order to verify the correct tracking of body parts between consecutive frames.

The pose reconstruction can be either 2D or 3D achievable with input data coming from single- or multi-view systems. The most difficult and ill-posed problem, due to the perspective ambiguities and self-occlusions, is the estimation of full 3D poses from a single view.

2.4.3.1 Bottom-Up Approach

In this approach, also called *data-driven*, no a priori model is used to aid in the tracking, however it is used to represent the obtained pose. Therefore the employed humanoids do not need to have a similar external shape to that of the real subject, and usually are simpler than those of the top-down approach. In order to establish the correspondence of joints between successive frames heuristic assumptions are usually employed. These impose spatio-temporal constraints to image-feature correspondences, decreasing the search space and therefore allowing a unique match. The validation of the reconstruction is attained by suitable objective functions, correlation or distance measures.

In order to overcome limitations of tracking over long sequences in which the correct correspondence can be lost at some time due to a possible incremental tracking error from frame to frame, the direct pose can be detected on individual image frames. Additionally, the direct pose estimation allows rapid movements. Two bottom-up pose reconstruction alternatives are distinguished in the literature:

- **Probabilistic assemblies of parts:** The most likely body part locations are firstly detected and then assembled to obtain the configuration which best matches with the observations. Physical constraints, such as body part proximity, can be used in the assembling process. Additionally, temporal constraints can also be applied in order to estimate consistent pose configurations over sequences and also to cope with occlusions. Representative works in this area are (Ioffe and Forsyth 1999; Ramanan and Sminchisescu 2006; Ren et al. 2005; Ronfard et al. 2002).
- **Example-based:** These methods initially learn and then apply a mapping from the detected image-features to the 3D pose data, even using only a single view. The ground truth data obtained with marker-based mocap systems is used for training. Then, the mapping is usually obtained by interpolating the candidate poses. It must be taken into account that this mapping, apart from body poses, also contains implicitly information about the body dimensions, viewpoint and appearance. Hence, the system needs to generalize well over the invariant parameters while distinguishing correctly the variant ones.

Representative works in this area are (Agarwal and Triggs 2006; Brand 1999; Elgammal and Lee 2004; Okada et al. 2006; Sminchisescu et al. 2005).

The bottom-up approach has the advantage, with respect to the top-down approach, that the initialization of the human motion capture can be automatized, i.e., it allows starting the reconstruction process without the need of defining manually a certain already known posture and the measurements of the subject's anthropometry and shape. However, this strategy usually produces many false positives in the search of human limbs and it needs to detect most body parts, since missing information results in less accurate pose estimations.

The probabilistic assemblies of parts bottom-up approach has made an important contribution in the 2D pose estimation in single-view cluttered natural scenes, which can be applied to surveillance systems. On the other hand, a limitation of current example-based bottom-up approaches is the restriction to the poses used in training, with relatively small variation in motion and fixed transitions. Another drawback is the large memory amount needed to store the database, along with the computation time needed to process it.

An example of the current state-of-the-art in bottom-up markerless motion capture is that of the commercial system Softkinetic (2008) (Figure 11). This system is capable of identifying and tracking the subject's body parts, without requiring an initialization process, from a depth sensing camera in real-time (about 15 fps). It is mainly aimed to HCI applications in which it obtains robust results but can also be usable for animating industry standard 3D skeletons.

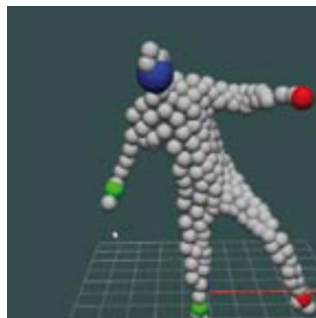


Figure 11: Softkinetic markerless motion capture [© (Softkinetic 2008)].

2.4.3.2 Top-Down Approach

In this approach, also called *model-driven*, a previously defined virtual model that represents the subject is used to match its projections with the image-features, and consequently aid explicitly in the tracking process. Hence, the virtual humanoids in this approach tend to be complex as they intent to match both the anthropometry and the shape of the real human. Thus, the number of parameters in order to describe the humanoid is usually higher than in the case of the bottom-up approach.

Based on the early work of O'Rourke and Badler (1980), the tracking and pose reconstruction procedures in this approach are solved by an *analysis-by-synthesis* strategy. It consists of four main steps:

1. **Prediction:** The model previous poses are used to make a prediction of its current configuration.
2. **Synthesis:** The prediction of the model pose is translated to the measurement level (images).
3. **Image Analysis:** A subset of regions and features are selected in order to evaluate the matching between the model and the images.
4. **State Estimation:** The new model state is computed.

As it can be deduced from this scheme, it is necessary to already have determined both the anthropometry and the shape of the subject, and also the initial posture before tracking starts. It can also be seen that it is important to attain a good accuracy in the model acquisition process for further correct pose estimation. Apart from the a priori knowledge related to the human model itself, it is often used other a priori knowledge related to the images such as the real subject's color appearance, and also maybe the expected motion types. The former must also be obtained during the initialization process, and the latter can be determined with pre-recorded sequences obtained with a marker-based mocap system.

In order to obtain the pose reconstruction from the image-features, a local search is often performed around the initial guess of the model. A brute force local search has an excessive computational cost due to the high number of DoF of the humanoid (normally over 20), their highly nonlinear relationship with respect to the image-features and the high cost of forward rendering the model in order to calculate the matching differences. Hence, constraints, such as biomechanical joint rotation limits or collision avoidance of body segments, are applied on DoF to decrease the search space during the matching process. However, these may introduce additional difficulties in the estimation as they

are simplifications of the biomechanical characteristics of real people. Thus, real biomechanical joint rotation limits are coupled, which means that they vary depending on the orientation of other neighboring joints. This fact is usually not considered in virtual models, which use statistical data (Engin and Chen 1986; Herda et al. 2003; Kapandji 1974; Kapandji 1982; Kapandji 1988; Wang et al. 1998). For example, in the case of a hip joint, that controls the movements of a leg, its rotation range is higher if the corresponding knee is flexed instead of keeping it totally stretched. Nevertheless, these behaviors can be emulated by learned models of joint limits and their correlation (Moeslund et al. 2002; Mulligan 2005).

Apart from this coupling, biomechanical rotation ranges vary from subject to subject and even for the same subject depending on his/her current fitness level. Colliding body parts are usually considered to be totally rigid instead of taking into account the soft-tissue nature of real flesh. This way, errors are propagated through the kinematical chain, which can lead to a loss of motion correspondence.

Human Modeling and Tracking Initialization Process

There are two strategies in the literature for the determination of the kinematic structure, and, if desired also the shape and appearance of the subject:

- **From static poses:** The kinematic structure and shape are estimated from a single or a set of predefined poses. Left-right skeletal symmetry is commonly imposed during estimation. The *T-pose*, in which the subject is standing in a cross-like pose with the legs slightly separated, is typically used as the height and arm span can be easily obtained from it. Representative examples of this approach are (Barrón and Kakadiaris 2003; Carranza et al. 2003; Cheung et al. 2003; Parameswaran and Chellappa 2004; Plänkner and Fua 2001; Starck and Hilton 2003).
- **From motion:** In this case the kinematic structure and shape are directly estimated from sequences of a moving person. The type of movements used in this approach can be predefined in order to reveal the structure (Kakadiaris and Metaxas 1995; Krahnstoeber et al. 2003) or general, which can be applied to surveillance (BenAbdelkader and Davis 2006; Grauman et al. 2003; Mikic et al. 2003; Song et al. 2003).

These techniques can also be applied to start the top-down tracking process once the virtual model that defines the user has been fully determined, as the poses of both are supposed to be aligned.

Tracking and Pose Reconstruction Process

There are many different approaches in the literature that intend to match the predicted humanoid pose to the image-features. Nevertheless, these can be catalogued in two main groups:

- **Multibody Shape Matching:** This strategy iteratively updates the DoF of the humanoid by an optimization technique until a satisfactory matching result is obtained with respect to the silhouette projections (maybe including depth information), contours or the visual hull. This way, the pose estimate is often found by applying a deterministic optimization technique (Bregler et al. 2004; Carranza et al. 2003; Cheung et al. 2003; Delamarre and Faugeras 2001; Gavrilu and Davis 1996; Kakadiaris and Metaxas 2000; Plänklers and Fua 2001; Rohr 1994; Sundaresan and Chellappa 2005; Wachter and Nagel 1999). However, the use of a single pose estimation which is updated at each time step in deterministic optimization approaches may lead to a loss of tracking if there is a rapid movement or visual ambiguities. Alternatively, techniques which employ a stochastic search of the pose estimation achieve more robust tracking, but with a higher computational cost and a jitter which must be smoothed to obtain visually acceptable results (Corazza et al. 2006; Cham and Rehg 1999; Choo and Fleet 2001; Deutscher and Reid 2005; Kehl et al. 2005; Navaratnam et al. 2005).
- **End-Effectors Driven:** In this case only some body parts of the user, which usually correspond to hands, head or feet, i.e., the end-effectors of the multibody mechanism that represents the subject, are directly tracked. Vision based techniques aided by the model are used for their motion correspondence and the rest of the body parts are estimated with robot inverse kinematics applied to the model, such as Cyclic Coordinate Descent (CCD) (Wang and Chen 1991), Jacobian Transpose (Balestrino et al. 1984; Wolovich and Elliot 1984), Pseudoinverse (Whitney 1969), Damped Least Squares (DLS) (Nakamura and Hanafusa 1986; Wampler 1986), DLS with Single Value Decomposition (SVD) (Maciejewski 1990; Maciejewski and Klein 1988), Selectively Damped Least Squares (SDLS) (Buss and Kim 2005) and Prioritized Inverse Kinematics (PIK) (Baerlocher and Boulic 2004). Representative examples following this strategy are (Date et al. 2004; Wren 2000).

Multibody shape matching top-down techniques have a higher computational cost than end-effectors driven ones, but are the markerless

approaches (including bottom-up approaches) that obtain higher accuracies in the reconstructions. Nevertheless, actually there still remains a noticeable gap between the accuracy of commercial optical marker-based and markerless human motion reconstruction, but it is tending to be smaller. This way, in order to help the human motion and pose estimation community to evaluate the accuracies of the markerless approaches, recently, the HumanEva databases (Sigal and Black 2006) have been created, in which the obtained results can be compared with those of a commercial optical marker-based mocap system. In the future markerless approaches can even excel marker-based in accuracy taking into account that the former do not suffer from the existing non-rigid movement of markers placed on the skin.

Occlusions of the tracked individual body parts in the end-effector driven top-down approach can make the reconstruction obtain significantly different poses from the real. On the other hand, in multibody shape matching techniques reconstruction errors are propagated through the kinematic chain, and therefore it may occur, for example, that an inaccurate estimation for the torso induced by self-occlusions, may cause errors in estimating the orientation of body parts lower in the kinematic chain. The side-effects produced by occlusions can be diminished by using learned motion models, which allow the synthesis of natural motions obtained from a database created from a marker-based mocap system. Representative examples of inverse kinematics of human motion based on learned models that have recently been introduced in computer graphics and that can be exploited for motion capture are (Chai and Hodgins 2005; Grochow et al. 2004; Liu et al. 2006).

A remarkable advantage of end-effectors driven top-down approaches with respect to multibody shape matching is that they do not need an accurate human model in order to achieve stable reconstructions, even if the obtained poses are distant from those of the user. This simplifies the initialization procedure and consequently facilitates their usage to HCI applications.

An example of the current state-of-the-art in top-down markerless motion capture is that of the commercial system of Organic Motion (2008) (Figure 12). This system is capable of identifying and tracking with remarkable accuracy and robustness the subject's body parts requiring only a T-pose initialization. It uses 14 grey-level cameras and can process the capture and rendering from 500 to 8000 polygons at 60-120 Hz using the Quad-Core Technology. It is mainly aimed to the animation industry and life science (biomechanical research, sports and clinical applications).

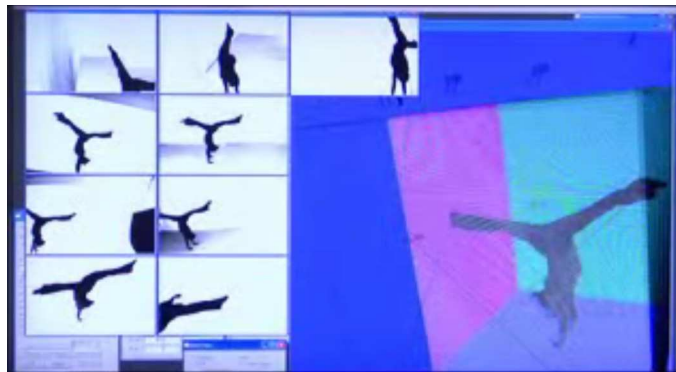


Figure 12: Organic Motion markerless motion capture [© (Organic-Motion 2008)].

2.5 HUMAN MOTOR ACTION RECOGNITION SURVEY

As it has been stated in section 1.1.3 the tracked human motion can have a high level semantic interpretation on its own. In order to make the computer achieve this objective automatically, two main steps can be distinguished in general: (1) the determination of the potentially meaningful motion-features and (2) the extracted motion-features classification process.

2.5.1 POTENTIALLY MEANINGFUL MOTION-FEATURES

Human motion tracking, in the same way in which sections 2.4.2 and 2.4.3 have showed that can be attained for evolving silhouettes, separate body parts and multibody mechanisms, has also been applied accordingly in the literature for motor action interpretation purposes. This way, we can distinguish *holistic* and *non-holistic* motor action recognition approaches. The former uses human figures as a whole, without identifying body parts, while the latter focuses on specific motion-features of human body parts for the semantic interpretation. Therefore, it can be deduced from the literature that it is not strictly necessary to solve the pose reconstruction process for further motor action recognition. Nevertheless, the 3D reconstruction offers the possibility of obtaining viewpoint independent motion-features directly linked to the body pose, which additionally can be expressed in more varied forms: joint angles, joint coordinates, velocities, accelerations, etc. Consequently, more complex actions can be recognized from the reconstructed poses.

Apart from the adopted tracking procedure to get the motion-features for further recognition, it is important to know whether the meaningful actions come through static poses or a time-ordered set of poses. The former only have spatial information while the latter are in *spatio-temporal* (ST) form. The

spatial data recognition procedure needs only to distinguish the isolated meaningful pose from the continuous data flow ignoring the rest, while the ST data needs to determine the starting and ending time instants in which the meaningful motor action is being performed. This process is also known as *gesture spotting* (Mitra and Acharya 2007). Two strategies have been addressed in the literature for representing motion-features for further recognition: *templates* and *state-space models*.

Additionally, a technique commonly used on both approaches in order to determine which among all the tracked motion-features are more important for the classification is the principal component analysis (PCA) (Fukunaga 1990). Using this technique the dimensionality of the motion data is reduced while at the same time containing the core of the performed motor action. This way, those motion-features that may degrade the correct classification rate can be excluded, since they do not have representative information. On the other hand, in the case that the style difference of the performances is intended to be analyzed it might not be interesting to apply this technique.

2.5.1.1 Templates

In this representation strategy an observed motion sequence is converted into a static shape pattern which is compared with other in the knowledge database. Hence, static meaningful poses can be considered as templates with spatial information only (Freeman et al. 1996; Haritaoglu et al. 2000; Jojic et al. 2000). On the other hand, ST motor action templates usually contain the velocities, accelerations, trajectories or even static pose manifolds of the movements intended to be classified. ST templates have been widely used in holistic recognition procedures, in which no pose reconstruction is required, but can also be employed for depicting the spatial evolution of reconstructed joint positions or angles. The main advantage of templates is their low computational complexity. However, they are sensitive to the time variability of the performed actions, and therefore are more aimed to classify simple actions.

In single camera holistic motor action recognition, 2D meshes of subject images have been used as templates, in which the optical flow of the points on the grid contains the motion-features for motor action classification (Efros et al. 2003; Polana and Nelson 1994). Templates with the same flavor to these meshes are *motion-history images* (MHI) (Bobick and Davis 2001), in which pixel intensities are a function of motion recency, and manifolds of *recursively filtered* images (Masoud and Papanikolopoulos 2003), which are groups of sequential images similar to MHI expressed in PCA space. Analogously to the latter, (González 2004) uses manifolds of *key-frames* expressed in PCA space, in which key-frames correspond to the most characteristic poses of a motor action, i.e.,

the locally least likely poses that compose the motor action. In his work, poses are represented with the relative joint angles of the reconstructed skeletons, and therefore are viewpoint independent. However, there is not a distinction on which of these joints do have more importance for defining the meaning of actions. Similarly, (Rahman and Robles-Kelly 2006) use a set of mean poses from different performances of actions in PCA, but in a normalized time scale. Other view independent templates are *motion-history volumes* (MHV) which extend the MHIs to the 3D world using the visual hull of the subject (Weinland et al. 2005).

Alternatively, *scale-space* of ST-curves (Allmen and Dyer 1990), trajectories (Rangarajan et al. 1993) or silhouettes (Roh et al. 2006) have been used. The scale-space representation of a signal is a family of derived signals parameterized by the size of the smoothing kernel used for suppressing fine-scale structures. This way, the scale-space of templates containing similar motions or poses will be similar, and thus, their point by point difference will be small. The curvature scale-space of ST-curves are effective motion-features for the recognition of cyclic actions, such as walking or running, as cycles are position invariant, as well as the curvature scale-space (Allmen and Dyer 1990).

More recently, three-dimensional templates like the ST volumes (STV) (Yilmaz and Shah 2005) have been proposed. These contain information of human silhouettes tracked along a normalized time scale and are treated as solid objects for further comparison to other known “objects” in the database.

2.5.1.2 State-Space Models

This is the most used approach to represent human motion sequences for motor action recognition. State-space models define the considered instantaneous motion-features as a *state*, and therefore a sequence is considered as a tour going through various states. This way, each state is composed of information that may correspond to positions, orientations, velocities, accelerations of reconstructed joints or maybe of certain characteristics of image-features directly, e.g., blob centroids. Additionally, other extra features coming from the context for the interpretation of activities in higher level of abstractions may be included in states.

Connections through these tours are defined, i.e., states are time-ordered, so as to determine the semantic meaning of movements. Consequently, the problem of time interval of movements is not an issue anymore using this approach, but it usually involves expensive iterative computation, as intrinsic nonlinear models are usually applied and do not have a closed-form solution. On the other hand, it is not obvious to select the

proper number of states, apart from the number and type of motion-features that define them, to avoid “underfitting” or “overfitting”. Additionally, as state connections are determined, it is possible to predict next states to the current during the motion capture process from this abstraction layer, which might help in tracking, especially when occlusions occur.

An early work using the state-space approach is that of Yamato et al. (1992), in which state information comes from human silhouettes in a single view system. On the other hand, the translation and angular velocity of blobs representing body parts are used for each state in (Bregler 1997). More recently, in the work of Ahmad and Lee (2006), states combine the cartesian components of the optical flow velocity and the human body shape in PCA extracted from a multicamera system.

Another example of actions expressed by state-space models are the XT-slices from an image space-time volume XYT (Ricquebourg and Bouthemy 2000; Rittscher et al. 2002) where articulated motions can be associated with trajectory patterns. Rittscher et al. (2002) show how XT-slices can be used for enhancing the tracking.

A more sophisticated state-space modeling is used in the works of Ren et al. (2002; 2004), where apart from motion-features of the subject’s body parts, additional context information of surrounding objects are included, attaining this way higher level scene descriptions. Features are weighted in order to establish their importance with respect to the actions to be recognized.

2.5.2 MOTION-FEATURES CLASSIFICATION PROCESS

Once the way in which actions are going to be represented has been defined, a comparison must be made in order to classify the movements being captured. Different performances of the same motor action, depending on how actions have been modeled, may form well defined clusters. Therefore, many statistical classifiers can be applied in order to set the incoming unknown movement into one of the labeled clusters of the database (Jain et al. 2000; Mitchel 1997).

Previously, before the motion recognizer starts labeling new data, it is necessary to train it with predefined data. Hence, *supervised learning* procedures are used. Afterwards, depending on the motor action recognition procedure, the classification will come from the minimum distance to a certain motor action in the database, or from that generating the maximum likelihood. The typical classification procedures found in the literature for motor action recognition are Nearest Neighbors, Dynamic Time Warping (DTW), Hidden Markov Models (HMM), Dynamic Bayesian Networks (DBN), Neural

Networks and kernel methods such as Support Vector Machines (SVM) and Relevance Vector Machines (RVM).

2.5.2.1 Nearest Neighbors

The general nearest neighbors method, called K-Nearest Neighbors (K-NN), stands out by its simplicity. Basically, the method looks for the K samples of the database that are closer to the unknown sample, and then it labels that sample by voting, and may be also taking into account the distance values. Usually the metric used to measure distances between samples is the Euclidean distance, but other can also be used. The nearest neighbor (1-NN) approach is widely used for motor action recognition. Representative works that have used this approach or derived variants are (Bobick and Davis 2001; González 2004; Masoud and Papanikolopoulos 2003).

2.5.2.2 Dynamic Time Warping

DTW is a well known technique to match patterns in which time scales are not perfectly aligned but with the same time ordering constraints. Hence, accelerations and decelerations during the performances with respect to the reference patterns can be managed. It has been widely used as a template-based dynamic programming (Bellman 1957) matching technique to recognize isolated patterns, but it can also be adapted to state-space models. Representative works that have used this approach for motor action recognition are (Darrell and Pentland 1993; Kang et al. 2004a; Takahashi et al. 1994).

2.5.2.3 Hidden Markov Models

The model structure of an HMM can be depicted as hidden Markov chain and a finite set of output probability distributions. Firstly, in the training stage, the number of states must be specified, and the corresponding state transformation and output probabilities are optimized so that the generated symbols correspond to the motion-features within the examples of a specific movement class. Then, in the recognition stage, the probability that a particular HMM generates the symbol sequence corresponding to the observed image features is computed.

HMMs are the most widely used approach for motor action recognition in the literature. Nevertheless, they have relevant limitations like the assumption that successive observations are independent, and also their high computational cost compared to other methods, as they make use of dynamic programming. Representative works that have used this approach for motor

action recognition are (Ahmad and Lee 2006; Bregler 1997; Wren et al. 2000; Yamato et al. 1992). More sophisticated variants of HMM have also been used for motor action recognition, such as Coupled Hidden Markov Models (CHMM) (Brand et al. 1997) and Variable-Length Markov Model (VLMM) (Galata et al. 2001).

2.5.2.4 Dynamic Bayesian Networks

The DBN is a general state-space model to describe a stochastic dynamic system. DBNs represent the hidden and observed states in terms of state variables, which can have complex interdependencies. The HMM can be considered as the most simple DBN. Due to their generality, DBNs have more expressive power and can therefore be used for recognizing more complex actions as they can fuse diverse sources of information, e.g., surrounding context information apart from motion-features. Representative works that have used DBN for motor action recognition are (Luo et al. 2003; Park and Aggarwal 2004; Ren et al. 2004).

2.5.2.5 Neural Networks

Neural networks consist of an ordered sequence of interconnected nodes which are coordinated by temporal and motion events. The detection of an event activates one or more nodes, which might trigger other nodes at higher levels, up to the output level, representing the semantic meaning of the global motion. The main advantage of the connectionist approach is its suitability for handling temporal information. Representative works that have used neural networks are (Guo et al. 1994; Rosenblum et al. 1994; Yu et al. 2005). Variants from neural networks such as Time-Delay Neural Network (TDNN) (Lin et al. 1999) have also been used.

2.5.2.6 Kernel Methods

Kernel methods are a class of algorithms for pattern discovery that map motion-feature vectors into a different dimensional space using some kernel function, and then use a variety of methods to find relations in the data. The most used kernel methods for motor action recognition are SVM and RVM. The former is a machine classification method that minimizes the classification error without requiring a statistical data model. The latter has the same functional form as the SVM but within a Bayesian framework. Examples of works that use SVM for motor action recognition are (Ardizzone et al. 2000; Ramanan and Forsyth 2003), while an example of the use of RVM for the same purpose can be found in (Guo and Qian 2006).

2.5.3 GESTURE SPOTTING

As stated by Lee and Kim (1999), gesture spotting derives from *pattern spotting* (Rose 1992), and consists on segmenting the continuous data flow into relevant gestures. It is a difficult task due to: (a) the *segmentation ambiguity* (Takahashi et al. 1992), and (b) the *ST variability* (Baudel and Beaudouin-Lafon 1993) involved. The segmentation ambiguity refers to the fact that direction, velocity and acceleration changes may occur on both the potentially meaningful gestures and those of the transition and therefore there is no an a priori determined difference in both types of movements. On the other hand, the ST variability refers to the potentially meaningful performance duration differences that will surely occur even for the same subject performing the same gesture continuously. An ideal gesture spotting procedure will extract potentially meaningful gesture segments from the continuous data flow and compare them with the patterns of the knowledge database allowing a wide range of spatio-temporal variability.

For example, in (Takahashi et al. 1994) gesture spotting is solved by using Continuous DTW. In this approach it is assumed that the current observation corresponds to the last state of the gestures in the database, and thus a certain gesture is spotted when its corresponding frame by frame accumulated distance reaches a minimum value. Similarly, in (Roh et al. 2006) gesture spotting is achieved by using the history of matched subject's silhouettes with those of the database, which is a function of time domain. On the other hand, the approach presented in (Lee and Kim 1999) consists in thresholding the probability of the performed movements to be found within the gesture database using HMM. Its main drawback is that the response is given when next gesture is performed and not immediately, which preserves the interface from naturalness. (Kang et al. 2004a) use DTW and distinguish definite gestures from tentative gestures using the recognition values of the tentative gestures in a sliding window, which indicates a certain range in temporal space. In (Yang et al. 2006) non-gesture garbage models are explicitly used in an HMM state-based representation. See (Mitra and Acharya 2007) for a further review in gesture spotting and recognition.

2.5.4 COMBINED MOTOR ACTION RECOGNITION

Humans are capable of distinguishing easily the simultaneous performance of various motor actions such as waving with a hand while walking, running or being seated, for example. However, this is not a straightforward task for a computer by only considering a certain and constant set of motion-features for every motor action to be recognized. Following a holistic procedure it would

be necessary to store in the database all the possible motor action combinations so as to be able to recognize them. This can make the database too cumbersome to be used due to its big size and its possibly confusing sample distribution. Moreover, the variability on the performance of combined actions will surely be even higher than in those being performed isolately. Hence, non-holistic procedures which make use of pose reconstruction are better suited for handling these situations than holistic.

There are a few works that have exploited the pose reconstruction in order to combine the information coming apart from the different body parts for this complex motor action recognition procedure. For example, Emering et al. (1998; 1999) use an action model in which, firstly, the center of mass and end-effector velocities and positions, and finally joint angles are used separately, in a five level hierarchy, for action recognition. The hierarchical approach reduces the database search space at each level in order to accelerate the classification process, but can also result in an early misclassification in the upper levels. Ben-Arie et al. (2002) divide the body in torso, arms and legs, and compute a score to determine the full-body pose with a voting system in which body parts are involved separately. This way, they can recognize activities even when several body parts are occluded. On the other hand, in (Park and Aggarwal 2004) the body is also divided in the same manner, but in order to define a dictionary of interactions between two actors, by relating the different separate limb recognition results.

2.6 DISCUSSION

It has been seen in this chapter that there are many strategies to allow the computer accomplish automatically the objective of “looking at people”. It has also been seen that there is not a definitive way to solve it for every application. The specific application to which the procedure is aimed to will determine the requirements. This way, there are some applications that will only need the use of a single camera while other will need multi-camera systems. Moreover, many applications will need to be in controlled environments as a robust background subtraction and occlusion handling are the toughest tasks to be accomplished by computers, as pixels have the same shape for the areas of interest and those that are not.

In the process of recognizing the semantic meaning of human motion it can be appreciated that the use of reconstructed poses allows to attain the description of more complex motor actions, such as those being performed simultaneously, or even for a higher level activity description where the surrounding context can get involved. Nevertheless, not all the information

coming from the reconstructed poses is valid for the recognition of every motor action. Some of these values may even add confusion to the classifier. Thus, the motion-features to be used will be determined by the motor action database, and consequently by the application.

The work presented in this thesis project is aimed for HCI which includes many types of applications, but concretely it is aimed for those indoor in which only one subject communicates messages to the computer with full-body movements. Therefore, a single camera is enough, and the environment can be controlled. As HCI is the final purpose of this work, gesture spotting and combined motor action recognition will also be explored. Additionally, it must be allowed, for future research, to have the possibility of interacting with a 3D virtual environment, i.e., a higher level of interaction. Therefore, firstly 3D full-body poses must be obtained, which observed from a single view constitute the most challenging reconstruction problem as it has been seen in this review. Finally, it must be remarked that real-time performance is mandatory for all the processes but high reconstruction accuracy is not. Only an accuracy which allows correct motor action recognition is necessary.

CHAPTER 3

HUMAN BODY PART MARKERLESS TRACKING

According to the taxonomy presented in chapter 2, the full-body markerless motion capture presented in this thesis project can be catalogued as: a single-view 3D *top-down end-effector driven* pose reconstruction approach, in which the motion correspondence of hands, head (or face) and feet is achieved by a *template based kernel tracking* combined with a *point tracking statistical method*, while in the case of the pelvis, another *point tracking statistical method* is used. This scope takes advantage of the observation by Liu et al. (2006) that the data obtained from optical marker-based mocap systems exhibit considerable redundancy, and that there is a reduced set of information, the *principal markers*, which retain the essential information concerning movements. Depending on the type of movement, some are more important than others. But in general, we could state that the positions of the end-effectors, i.e., the wrists in the case of the arms, the ankles in the case of the legs, and the pelvis and the head in the case of the torso, are important to situate a human pose. There are exceptions like those in which the size of the captured user is different from the virtual character and, as Shin et al. (2001) state, we may be more interested in preserving the angles of the original pose instead of the positions of the end-effectors because they are not interacting with the environment. We, however, are more interested in end-effector positions since using position data opens up the possibility of interacting with a virtual environment.

In this chapter, real-time markerless 2D strategies capable of tracking these end-effectors, with low noise, in movements performed at high speed are presented. These strategies can also handle different skin-colors and clothing. These tracking methods, combined with (a) the 3D full-body pose

reconstruction procedure presented in chapter 4, and (b) the view depth warping approach based on combined motor action recognition presented in chapter 5, enable a real-time markerless strategy to track full-body movements for a pseudo 3D motion reconstruction using a single standard camera. In addition, a full 3D motion reconstruction can be achieved by using a single depth-sensing camera applying the same tracking methods.

This chapter is organized as follows: firstly, a method for tracking in 2D the hands, head (or face) and feet, called Colored Features Optical Flow (CFOF), is presented. Secondly, a strategy for estimating the pelvis position in 2D, even with severe self-occlusion situations, is explained. Then we explain how the CFOF method can be adapted to self-occlusion situations in HCI applications in which a top-down end-effector driven pose reconstruction approach is used. Next, how full-body motion capture can be initialized from a single predefined pose without the need of manual intervention by anybody other than the user, is explained. Finally, experimental results that evaluate the performance of the CFOF method with respect to alternative approaches are given. The strategies presented here fulfill the objective of tracking, in real-time without markers, the minimum set of body parts for a further pose reconstruction and motor action recognition in HCI applications.

3.1 COLORED FEATURES OPTICAL FLOW

Using only color information may be problematic for the tracking of body parts when there are some (like the hands and face) with similar color characteristics. Furthermore, the difficulty of locating hands is increased when the forearm skin is also visible. Kölsch and Turk (2004) try to solve these problems by means of a *flock of features*. In their approach, initially, feature points of the body part to be tracked (a hand in their case) are extracted from the image using the method of Shi and Tomasi (1994). The flock centroid defines its position. Then, motion is tracked using an algorithm that combines the pyramidal implementation of Lucas and Kanade's (PyrLK) optical flow (Bouguet 1999; Lucas and Kanade 1981) of feature points and color probability. This feature search and tracking by optical flow methods forms the Kanade-Lucas-Tomasi (KLT) template based kernel tracking approach mentioned in chapter 2. During motion, those features that drive away from the flock median position, or are too close to other features, or lie outside the bounding box of the object, or have a low match correlation, are relocated in regions with high color probability based on the normalized-RGB (nRGB) color space. They obtain better results for the tracking of a hand than the well-known CamShift (Continuously Adaptive Mean-Shift) method (Bradski 1998). They use the median for removing distant features because the mean

position can be too influenced by the positions of the most dispersed features while the median is not. On the other hand, the mean or centroid of the feature positions is used for locating the hand position instead of the median because the former changes more smooth during motion.

There are, however, many color spaces apart from nRGB, that distinguish *chrominance* (color characteristics) from *luminance* (luminous intensity): HSV, HLS, YCrCb, Lab, Luv, xyY, etc. Sedláček (2004) found that the HSV model works better than the nRGB for human face detection. Moreover, Santurde et al. (2006) showed that HSV achieves better hand detection results than nRGB. On the other hand, the PyrLK optical flow is very sensitive to noise and can lead to unexpected displacements of features. Consequently, Sánchez and Borro (2007) use the Kalman filter (1960) applied to each feature in order to improve the outliers detection and, consequently, the tracking. If the displacement calculated by the optical flow is too distant from the prediction, it is removed.

Furthermore, in the flock of features tracking there may be some features dragged by the optical flow to regions not corresponding to the body part. This can be avoided by adding the condition of claiming at least a certain color probability to every feature. Taking these results into account the following procedure, called Color Features Optical Flow (CFOF), is proposed for tracking the hands, face and feet:

1. Firstly, the body part (hand, face or foot) is placed in a predefined region of interest (*ROI*) of the image and the system is initialized using Algorithm 1. The *ROI* is simply a rectangular (or maybe squared) subregion of the image.
2. Then, once the system has been initialized, the tracking procedure is applied frame by frame updating the position of the *ROI* using Algorithm 2.

In Algorithm 1, the chrominance of the object is learned, a maximum of $nMaxDetect$ feature points are searched for on the body part and their corresponding Kalman filters (KF) are initialized using the method proposed by Sánchez and Borro (2007). As they state, the Kalman filter is capable of predicting future states of a system described by Equation 1, where \mathbf{x}_k is the state vector at the instant k , \mathbf{A} is the transition matrix of the model and \mathbf{w}_k is a random variable that represents the process noise.

$$\mathbf{x}_k = \mathbf{A}\mathbf{x}_{k-1} + \mathbf{w}_{k-1} \quad (1)$$

The state they use includes the position (u_x, u_y) and the linear velocity (v_x, v_y) of a feature point as shown in Equation 2.

$$\mathbf{x}_k = [u_x \quad u_y \quad v_x \quad v_y]^T \quad (2)$$

This way, the position of the feature in the next frame should be the sum between the current position and the velocity, so the transition matrix \mathbf{A} of the filter is modeled by Equation 3.

$$\mathbf{A} = \begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3)$$

In addition, there is a relationship between the measurements and the process state given by Equation 4, where \mathbf{z}_k is the measurement vector, \mathbf{H} is the measurement matrix (Equation 5) and \mathbf{v}_k is a random matrix that models the measurement noise.

$$\mathbf{z}_k = \mathbf{H}\mathbf{x}_k + \mathbf{v}_k \quad (4)$$

$$\mathbf{z}_k = \begin{bmatrix} u_x \\ u_y \end{bmatrix} \text{ and } \mathbf{H} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \quad (5)$$

According to Sánchez and Borro, if the prediction of the Kalman filter is far from the position calculated by the optical flow algorithm, the feature is an outlier and must be removed from the tracking process. Otherwise, the state of the filter is corrected using the measured position. This distance is parameterized with the filter's prior error covariance.

The chrominance of the method proposed in this thesis project is expressed with the color histograms of Hue and Saturation channels in HSV color space (*bistHS*). Images filtered with the learned H and S histograms separately can be visualized as grey level images, *backprojections*, in which white pixels correspond to a 100% H or S probability from the learned model and black corresponds to 0%. Combining both H and S backprojections with an “&” operation results in another backprojection in which probabilities correspond to the chrominance (Figure 13). These histograms are learned from the central subregion of the ROI as it is intended to learn only the color characteristics of the object itself.

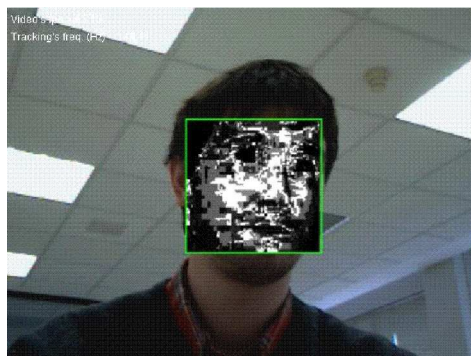


Figure 13: The backprojection of a face obtained from the corresponding learned skin-color HS chrominance histograms.

Then, after learning the image characteristics of the body part, the position of the *ROI* is relocated using a feature selection obtained from the initial feature flock. This feature selection is made applying a procedure that follows the same flavor of (Kölsch and Turk 2004) of maintaining features within a certain distance from the median (*maxDist*) and with a minimal color probability (*minColor*). We call this procedure Colored Maximum Distance Filter (CMDF) which is shown in Algorithm 3. Both the Kalman filter and CMDF provide a probabilistic reinforcement to the KLT procedure, which adds the component of a point tracking statistical approach to the template based kernel tracking procedure.

As can be observed in Algorithm 2, in the case that all feature points have been lost during tracking, the *ROI* may stay still until the number of pixels with at least a minimal color probability, according to the learned model, is sufficient to restart the process. This way, an online tracking recovery is possible.

Algorithm 1 Initialization Procedure of CFOF

```

1: procedure INITIALIZATIONCFOF(ROI, nMaxDetect, nSelect, maxDist,
   minColor)
2:   Learn the histHS of the central subregion of ROI
3:   Search features in the ROI using (Shi and Tomasi 1994)  $\rightarrow$  nFeat =
   nDetect features (normally  $nDetect \leq nMaxDetect$ )
4:   Initialize KF (one for each detected feature)
5:   Apply CMDF to the nFeat features
6:   Situate ROI center in the feature selection centroid (using the
   first nSelect features of the nFeat)
7: end procedure

```

Algorithm 2 Tracking Procedure of CFOF

```

1:  procedure TRACKINGCFOF(ROI, nSelect, minNColorPixels, maxDist,
    minColor, histHS, KF, features)
2:      nColoredPixels = getColorPixels(ROI, histHS)
3:      if nFeat = 0 & nColoredPixels ≥ minNColorPixels then
4:          Search features in ROI using (Shi and Tomasi 1994) → nFeat =
            nDetect features (typically nDetect ≤ nMaxDetect)
5:      end if
6:      if nFeatures > 0 then
7:          Update feature positions with PyrLK optical flow → nFeat =
            nTemp1 features (typically nTemp1 ≥ nSelect)
8:          Estimate feature positions from previous frame with KF (to the
            nFeat features)
9:          Correct or remove feature positions obtained with PyrLK using
            the estimations of KF → nFeat = nTemp2 (nTemp2 ≤ nTemp1
            features)
10:         Situate ROI in estimated features centroid (using the first
            nSelect features of the nFeat)
11:         Apply C MDF to the nFeat features
12:         Correct or remove features using the estimations of KF (to the
            nFeat features)
13:         Situate ROI center in the feature selection centroid (using the
            first nSelect features of the nFeat)
14:     end if
15: end procedure

```

Algorithm 3 Colored Maximum Distance Filter (C MDF)

```

1:  procedure C MDF(ROI, nSelect, maxDist, minColor, histHS, features)
2:      Calculate median of the nFeat features
3:      for each feature do
4:          Calculate distance to median distToMedian
5:          Calculate HS color probability colorProb with histHS inside the
            ROI
6:          if distToMedian ≤ maxDist & colorProb ≥ minColor then
7:              Store feature
8:          end if
9:      end for
10:     if Number of stored features ≥ nSelect then
11:         Set stored first nSelect features in the selection
12:     else
13:         Set stored features in the selection

```

```
14:     for every pixel in ROI do
15:         Calculate HS color probability colorProb with histHS inside
           the ROI
16:         if colorProb  $\geq$  minColor then
17:             Store pixel
18:         end if
19:     end for
20:     Sort stored pixels from higher to lower color probability
21:     Fill the gaps of the feature selection with colored pixels in
           ROI to get a selection of at most nSelect features
22: end if
23:     Overwrite the first elements of the nFeat features with the
           obtained feature selection
24: end procedure
```

Figure 14 shows some images from tracking a hand with the proposed approach. Despite their similarity, this procedure enhances that of Kölsch and Turk (2004) in the following aspects:

- The HSV model is used instead of nRGB, which works better for modeling chrominances of body parts.
- The tracking procedure is less noisy. This occurs due to the following factors:
 - The CMDF does not contain the condition of minimal distance among features. This accelerates the procedure and decreases the number of feature position “corrections”. In addition, body parts are observed from a higher distance which makes the minimal distance among features constraint during the tracking disturbing. The minimal distance is only used for the initial feature search procedure.
 - The optical flow is applied to a higher number of features than those that are finally used for situating the *ROI* in order to increase the probability of existing features that meet the conditions of the CMDF, and hence, decrease the number of corrections.
 - The KLT tracking procedure is reinforced by means of Kalman filters which, apart from removing outliers, can also provide the possibility of estimating feature positions while occlusions occur.

- The system is more robust for the tracking of more than one body part with similar color characteristics (hands and face) in the same image. This is done by only searching for color probabilities inside the *ROI* instead of in the entire image. This requires a proper estimation of the *ROI* position before applying the CMDF, which is done by both the optical flow and Kalman filter. Additionally, this local search allows recovering from the loss of tracking by merely passing the body part through it again.

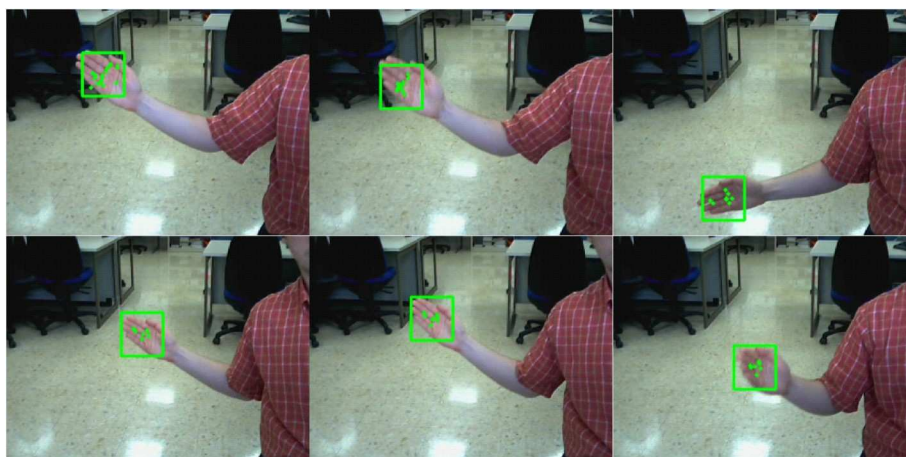


Figure 14: Some samples of the tracking of a hand using CFOF. The forearm skin is also visible but it poses no problem for the system.

3.2 PELVIS POSITION TRACKING

Apart from the tracking of the hands, head and feet with the CFOF method presented in section 3.1, it is also necessary to track the pelvis in order to obtain the minimal information required for the pose reconstruction to be explained in chapter 4. It would also be possible to apply the CFOF approach to track it, but it is preferable to use an alternative because its position lies in the neighborhood of the user's torso and legs, which involves the following issues:

- The *ROI* may not have an homogeneous color distribution to be learned. For example, it may contain the colors of the shirt, belt and trousers altogether, which may each be different from one another.
- The *ROI* may contain a smooth surface (for example, the trousers) from which it is difficult to extract features of sufficient quality for further optical flow tracking.

- Even though the CFOF is capable of handling different occlusion situations, as shown in sections 3.3 and 3.5, like the rest of the tracking methods in the literature, it can not handle severe occlusions. Specifically, the pelvis has a higher tendency to be unconsciously occluded with other body parts during motion.

Thus, as the pelvis position is located around the centroid of the user, its silhouette may be exploited. In order to extract the silhouette background subtraction techniques can be applied (Piccardi 2004). Two approaches that are appropriate for indoor HCI applications are proposed:

- **Gaussian RGB Model:** This strategy supposes stationary background and illumination. The procedure is the following: (1) initially, some time is required to learn the RGB color values of background pixels, (2) the mean (BG_{mean}) and standard deviation (BG_{stdDev}) backgrounds are calculated, and (3) once the user is in the image (I), the background is subtracted with Algorithm 4 (Figure 15). A threshold can be used for the conversion to binary in order to improve results.
- **Chroma-key:** This strategy makes use of a green or blue colored backdrop behind the subject that, taking advantage of the RGB color model, can easily remove those pixels with a color similar to that of the backdrop. The same effects would also be obtained by simply painting the room with one of these colors, but we will refer to backdrops, as they are of common use for chroma-key. The red tone is not used because skin-color, as shown in section 3.5, derives from this hue. Algorithm 5 shows the procedure when a green backdrop is used (Figure 16). Again, a threshold can be used for the conversion to binary in order to improve results.

Algorithm 4 Gaussian RGB Model Background Subtraction

```

1: procedure GAUSSIANRGBMODELBS( $I$ ,  $BG_{\text{mean}}$ ,  $BG_{\text{stdDev}}$ ,  $colorThr$ )
2:    $T = (I - BG_{\text{mean}}) \cup (BG_{\text{mean}} - I) - BG_{\text{stdDev}}$ 
3:   Convert  $T$  channels to binary separately (if pixel of  $T$  channel  $\geq$ 
      $colorThr$  then pixel = white else pixel = black)  $\rightarrow$   $TBinary_R$ ,
      $TBinary_G$  and  $TBinary_B$ 
4:    $SilhouetteMask = TBinary_R \cup TBinary_G \cup TBinary_B$ 
5: end procedure

```

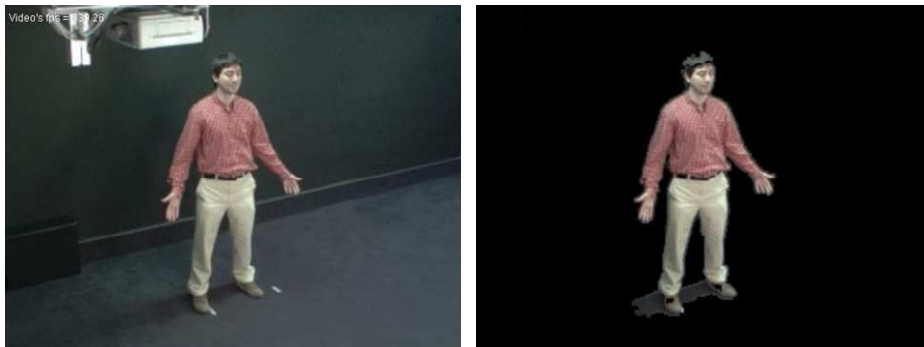


Figure 15: Background subtraction by Gaussian RGB model of pixels.

Algorithm 5 Background Subtraction with a Green Backdrop

```

1: procedure GREENCHROMAKEYBS( $I$ ,  $colorThr$ )
2:   Split  $I$  into separate color channels  $\rightarrow I_R, I_G$  and  $I_B$ 
3:    $T_1 = I_G - I_R$ 
4:   Convert  $T_1$  to binary (if pixel of  $T_1 \geq colorThr$  then pixel = white
   else pixel = black)  $\rightarrow TBinary_1$ 
5:    $T_2 = I_G - I_B$ 
6:   Convert  $T_2$  to binary (if pixel of  $T_2 \geq colorThr$  then pixel = white
   else pixel = black)  $\rightarrow TBinary_2$ 
7:    $TBinary_3 = TBinary_1 \cap TBinary_2$ 
8:    $SilhouetteMask =$  inverted binary image of  $TBinary_3$ 
9: end procedure

```



Figure 16: Background subtraction with a green backdrop behind the subject.

The output of these two algorithms is in the form of a binary image, the *silhouette mask*, which combined with the incoming RGB image results in an image where the user is visible while the background is set to black. The main advantage of the Gaussian RGB model is that it does not require having and installing any backdrop, but on the other hand, casted shadows are present. It

is not strictly necessary to erase them for the estimation of the pelvis position in a case where their size and shape are similar to that shown in Figure 15. Nevertheless, in case they are disturbing there are several strategies to remove them (Prati et al. 2003). In our case, as the background colors are darker than those of the user's clothes and skin, a threshold can be used in the grey-level image to remove them because their intensity values are lower than those of the non-shadow regions. On the other hand, it can be observed in Figure 16 that since the backdrop tone is very similar to that of the RGB green channel, using Algorithm 5 it is possible to attain clean silhouettes without casted shadows, even those coming from backdrop folds, even if they change during the user's performances. This excels the classical chroma-key technique, which directly removes the G or B channel (depending on the backdrop color) in RGB color space, and hence may be affected by shadows and backdrop folds. Moreover, the use of a backdrop allows a wider range of clothes and skin colors. Apart from these two methods, a depth threshold can be used to ignore elements behind the user in a situation where a depth-sensing camera is used. This approach is also appropriate for background subtraction for indoor HCI applications, but is not included in the results of this project thesis.

The desired pelvis position, estimated through the silhouette's centroid, can be disturbed by the limb movements. For instance, the position of the pelvis should not be affected by the movements of arms if the rest of the body is still. In an HCI application it may be supposed that the user's movements will mainly be circumscribed to vertical or slightly bent poses of the body, but not horizontal poses. Taking advantage of this situation, it is possible to estimate the approximate position of the pelvis with a higher stability than directly using the silhouette centroid, by applying Algorithm 6. Here, computational calculations are alleviated by increasing the grid size of the processed pixels, i.e., not processing every pixel but some separated by a step in both X and Y coordinates of the image. Again, the Kalman filter is used to smooth the estimated pelvis position because the silhouette extraction procedure relies on noisy color (mainly due to fluorescent light flickering and the gain applied to the images of the camera to get brighter observations). The median is used instead of the mean for pixel discrimination for the same reason as in the CMDF. The mean or centroid of the selected pixels are again used instead of the median because it moves more smoothly.

Figure 17 shows two examples of the image regions used for this procedure. It can be seen that arms are not included in the pixel selection and thus do not affect to the pelvis position computation. The left image corresponds to the use of a Gaussian RGB background model, and it can be observed that the casted shadows are ignored using the grey-level threshold.

Meanwhile, the right image corresponds to the chroma-key approach, and it can be observed that only the region within the backdrop is used for the calculation of the pelvis position. This is achieved by initially learning (before the user comes to the scene) where the backdrop is and then applying Algorithm 6 only to that region of the image, ignoring the rest.

Algorithm 6 Pelvis Position Tracking

```

1:  procedure PELVISPOSITIONTRACKING(SilhouetteMask, gridSize, maxDist)
2:    for pixels of SilhouetteMask on the grid do
3:      if pixel intensity > 0 then
4:        Store pixel in buffer1
5:      end if
6:    end for
7:    if buffer1 size = 0 then
8:      Estimate pelvisPosition from previous frame using Kalman filter
9:    else
10:     Calculate the median X coordinate of buffer1 pixels → xMedian
11:     for stored pixels do
12:       if xPixel ≥ (xMedian - maxDist) & xPixel ≤ (xMedian +
13:         maxDist) then
14:         Store pixel in buffer2
15:       end if
16:     end for
17:     if buffer2 size = 0 then
18:       Estimate pelvisPosition from previous frame using Kalman
19:       filter
20:     else
21:       Calculate the mean position of buffer2 pixels →
22:       pelvisPosition
23:       Correct current pelvisPosition using Kalman filter
24:     end if
25:   end if
26: end procedure

```

It must be stated that, depending on the anthropometry of the humanoid used for mapping the movements of the user, it may be interesting to experimentally tune the pelvis position obtained with this method. This can easily be done by applying an offset in *X* and *Y* to the estimation.

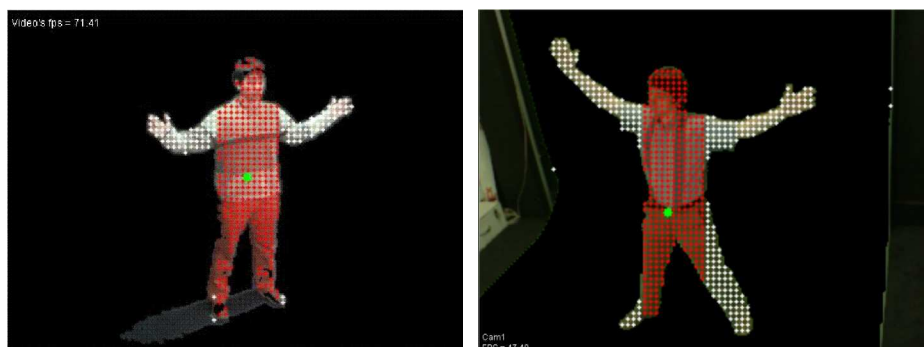


Figure 17: The estimated pelvis position (green dot) is calculated from the red pixels, which lie in the neighborhood of the median x coordinate of the highlighted pixels.

3.3 SELF-OCCLUSIONS

While the user is performing movements, some of the tracked body parts may be occluded to the cameras. In the case of the CFOF procedure, when the tracked body part is occluded by another one, the movement of that frontal body part “steals” its features due to optical flow. In other words, features that should continue tracking the body part will now be tracking the body part that occluded it. We propose overcoming this problem with the aid of a 3D human model that represents the user and the color distribution of the tracked region. Hence, we distinguish two cases: (a) the overlapping of two tracked ROIs and (b) the excessive color change of the tracked ROI.

3.3.1 OVERLAPPING OF TWO TRACKED REGIONS OF INTEREST

This case is solved relying on the underlying 3D human model, i.e., with a top-down approach. As explained in section 2.3, once we have calibrated the camera, we have the relationship between the 3D world and the projection. Hence, the projected 3D human model can be fitted to the images (using the reconstruction method to be presented in chapter 4) according to the tracked body features in 2D by maintaining their depth constant with respect to the view (Figure 18). This way the tracked ROIs can each control the motion of a humanoid end-effector (hands, face or feet). The system may be visualized as six mouse devices controlled by the user at the same time with full-body movements, where each end-effector moves in 2D in its own plane parallel to the view. We can use their depths with respect to the camera to handle occlusions.

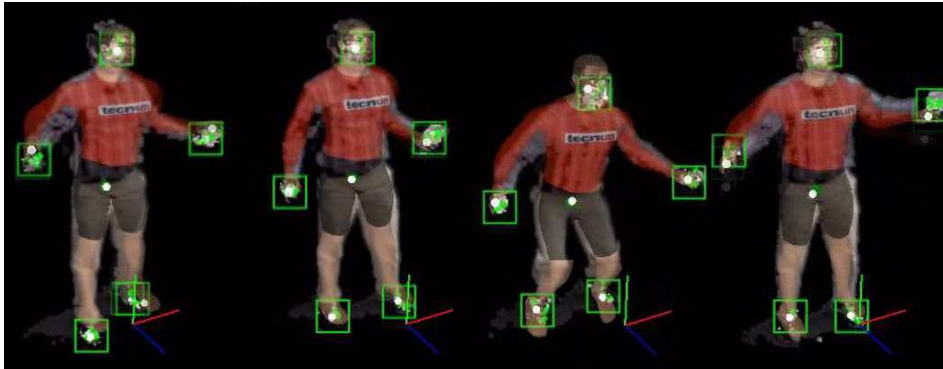


Figure 18: Markerless full-body motion capture samples maintaining the depths of the tracked body parts constant with respect to the view.

For instance, if we have two overlapping *ROIs*, the one with the least depth corresponds to the frontal end-effector in the image while the other is the occluded one. In Figure 19 it can be seen how the *ROI* corresponding to the right hand occludes the left hand *ROI*. Note that the occluded *ROI* boundary turns from green to red.

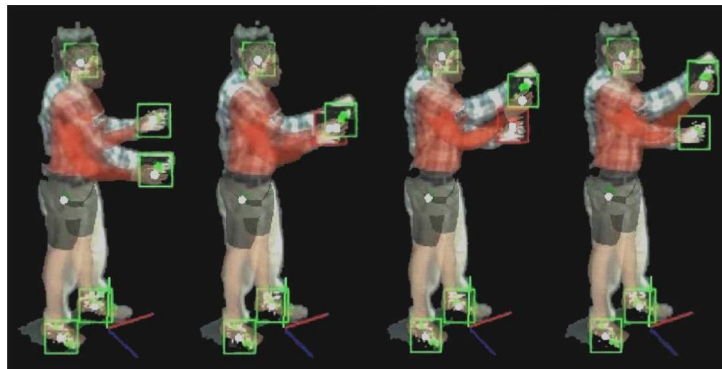


Figure 19: Left hand occluded by right hand. The occluded *ROI* boundary turns from green to red.

The adopted strategy is to stop the optical flow in the back *ROI* and continue with it in the frontal one. The left hand *ROI* cannot rely on optical flow while *ROIs* are overlapping because the right hand will “steal” the features that initially were attached to the left hand. The only possibility is to predict its motion while occlusion is occurring and retake the optical flow once the camera sees it again. It is impossible to predict every movement a human could perform because we cannot read his/her mind, so we can suppose that:

- The occluded end-effector will not move. In HCI applications this option may be sufficient as the user can have the option of recovering consciously the *ROI* of the occluded body part once it is visible again.
- The occluded end-effector will continue the trajectory it was performing without any sudden change, which can be predicted, e.g., with the mentioned Kalman filter, which is limited to linear assumptions, or other more elaborated versions, e.g., the Extended Kalman Filter (EKF) (Sorenson 1985; Uhlmann 1992) or the Unscented Kalman Filter (UKF) (Julier and Uhlmann 1997), applicable to non-linear systems.

3.3.2 EXCESSIVE COLOR CHANGE OF TRACKED REGIONS OF INTEREST

Another circumstance that may occur is that a body part which is not an end-effector occludes the tracked *ROI*. To solve this case we can rely on the learned color distribution of the *ROI*. In this case, we consider that the frontal body part color is different from the tracked end-effector, so if color changes excessively in the *ROI* it is assumed to be occluded. Consequently, we can stop the optical flow until the original color model is detected again in that *ROI*. Meanwhile, we can adopt one of the two strategies explained in section 3.3.1, i.e., (a) the occluded end-effector will not move and (b) the occluded end-effector will continue the trajectory it was performing without any sudden change.

In order to detect the color change, firstly we count the number of non-zero pixels in the backprojection of the image used during the initialization of the CFOF procedure, and secondly, while we are tracking, we make the same computation for each frame. In a case where the number of pixels is less than a threshold in relation to the initial number of pixels, it is considered that the tracked *ROI* is occluded. The greater the number of minimal pixels for considering excessive color change the more sensitive the system will be to color changes. This threshold must be defined depending on the application. In Figure 20 we have an example of excessive color change in which the tracking box corresponds to the right hand that is occluded by the shirt. As the shirt color is different from the skin-color of the example, its backprojection is totally black, so the optical flow is stopped (*ROI* boundary turns from green to red).

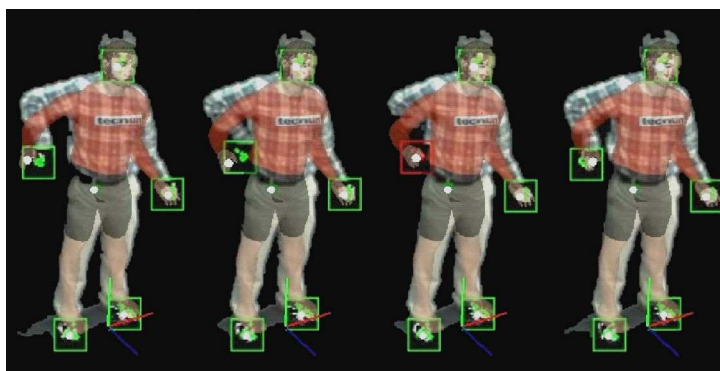


Figure 20: Occlusion due to the excessive color change on the tracked right hand *ROI*. The occluded *ROI* boundary turns from green to red.

3.4 FULL-BODY CAPTURE PROCESS INITIALIZATION

As stated in section 3.1, it is necessary to start the CFOF tracker in a known position of the user to initialize the color models of the end-effectors, to find the feature points to be tracked, and also, in case it is necessary for the full-body capture, to get the anthropometry of the user for the reconstruction of the postures. In HCI applications it is intended to make the initialization process as “user friendly” as possible. This implies that the system should be fully usable by only one person, the user, without the need for collaboration from any one else, e.g., for manually selecting the regions to be tracked, and without the need to perform a series of different predefined postures for anthropometry estimation. In our case, we are not especially interested in extracting the shape of the user as we only want to obtain postures similar to those of the user which we estimate from inverse kinematics, and for this procedure, only the lengths of the body parts to be measured are needed, i.e., the information provided by a stick-figure. Furthermore, as only visually apparent movements are needed for further motor action recognition, it is not necessary to calculate very precise results for the anthropometry measurements, so we propose a method that emphasizes the ease of use rather than precision.

Firstly, it is necessary activate the background subtraction since the user is the only visible figure that appears in the image. The user must focus on the view in which he/she will see four highlighted *ROIs* where he/she must set his/her hands and feet (Figure 21). It is recommendable to mark on the floor the positions where the feet stand to ease the process. Then, the user must step on the marks and then set his/her hands on the upper *ROIs*. This way the system is aware that the *ROIs* are filled with “something”, that means that since the background subtraction is active, the *ROIs* are empty while the user does

not actively fill them. The system will wait a certain number of frames and if the *ROIs* are still full at that point in time it is assumed that the user has consciously adopted the initialization posture. Then, the *ROIs* are initialized as explained in section 3.1 for finding feature points and for creating the color models.

Meanwhile, if background subtraction is active it is also possible to calculate the body's pelvis position as explained in section 3.2. The remaining *ROI*, the one corresponding to the face, is initialized by, firstly, calculating the arithmetic mean *X* coordinate obtained from the hands' *ROIs* and, secondly, by searching from top to bottom the first non-black pixel corresponding to that mean *X* coordinate. The face *ROI* is supposed to be located below that spotted pixel. Bearing this in mind, the exact position of the face *ROI* can be tuned experimentally to obtain the color characteristics of the face.

Regarding the anthropometry initialization, as the lengths measured between the pelvis and the rest of the end-effector positions will surely be different from those of the projected humanoid, its anthropometry can be readapted according to the observations. For instance, statistical human proportions (Pheasant 1986) can be used to modify the body part lengths fitting the measures obtained from the initialization posture.

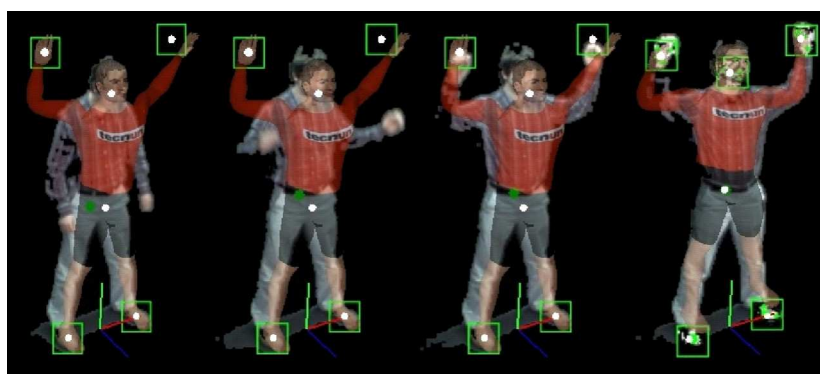


Figure 21: Full-body markerless capture initialization process samples.

3.5 EXPERIMENTAL RESULTS

The behavior of the CFOF 2D tracking method presented in this chapter is analyzed, with a benchmark, under typical HCI circumstances. These may be the usage of the system by people with different skin-color, fast movements, partial occlusions or backgrounds that have similar color to the user. Along with these, the computation time, and the accuracy and noise of the tracking are also evaluated, as they are also determinant for a satisfactory HCI. Different

alternatives for the tracking process are also analyzed and compared to the proposed method. The computer vision algorithms for optical flow and color probability calculations have been implemented using the OpenCV (Intel 2001) library for C++ and have been tested in a 2.66-GHz Pentium 4 with 512 MB RAM and 320×240 images.

The benchmark consist of a 2D scenario where a face image to be tracked performs a predefined non-linear trajectory in which some obstacles are placed. Other end-effectors (hands or feet) apart from faces could be used for the test, but this is not relevant for the evaluation since the approach works in the same way for any object to be tracked. The tracking procedure is evaluated by taking the following measures from this benchmark:

- The differences between the real trajectory of the face centroid and the tracked one. These are measured in a “clean” path and also with obstacles that partially occlude the face along the path.
- The maximum speed at which the face can be stably tracked. This means that the face can be tracked for a long period of time while moving at that speed.
- The number of false positives obtained with the color model. False positives refer to the number of pixels from non-face objects bearing a similarity with the tracked face color.
- The distribution of the learned color probability throughout the entire face.
- The computation time needed to process the tracking algorithm.

More precisely, the face to be tracked traces a circular trajectory, and the obstacles that can be found along the path are six masks with holes for the eyes, nose and mouth. The six masks are painted with gradual variations, each of a different hue, corresponding to the principal tones existing in nature (red, yellow, green, cyan, blue and magenta), which also contain a white color trimming so that a wide range of colors is considered (Table 2). The face and mask images are static photos to which artificial random noise is added to pixel colors in order to emulate the noise coming from a real camera. Four different face samples are considered which, according to the United States FBI (1908) nomenclature, correspond to *Asian*, *Black*, *Hispanic* and *White* human races (Table 3). It must be stated that these photos have been taken under different lighting conditions, but nevertheless differences in the skin-color can be appreciated.

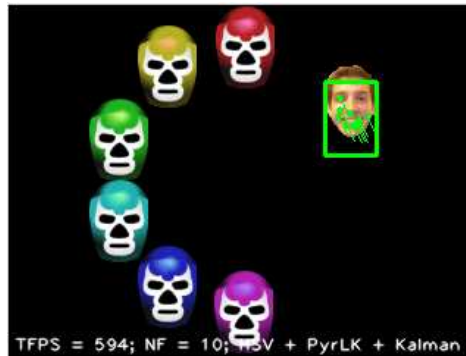


Figure 22: Benchmark for evaluating 2D object tracking procedures.


Hue	Red	Yellow	Green	Cyan	Blue	Magenta
Mask						

Table 2: Masks used as path obstacles for the evaluation of tracking procedures.





Race	Asian	Black	Hispanic	White
Face Sample				

Table 3: Face samples considered for the evaluation of tracking procedures.

Table 4 shows the face chrominances in HSV color space obtained by processing the central subregions of the images. They present a Hue channel histogram tending to a value between red and orange in all cases, while the Saturation channel differs slightly on the histogram distribution having their maxima around the total midrange. This means that, as they have similar color characteristics, the color probability distributions of the non-face objects observed with the learned model will also be similar. Besides, it is possible to predefine the skin-color models for the tracking of hand and faces for different human races without the need to specifically learn the face chrominance. Nevertheless, the system still requires initializing the feature point search.

Table 5 shows the backprojections of the different faces and masks based on the learned skin chrominances in nRGB, HSV, HLS, YCrCb, Luv and xyY color spaces, while Figures 23-28 show their probability distribution.

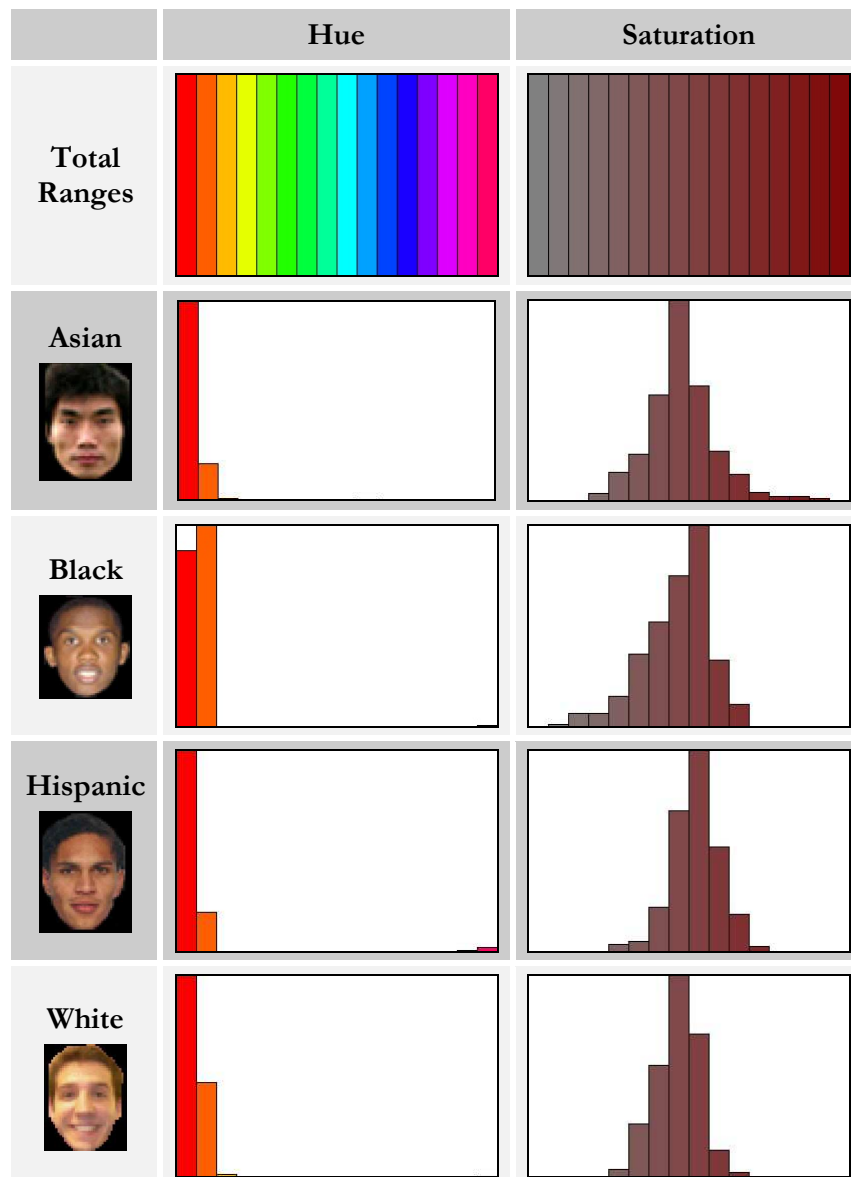


Table 4: Hue and saturation channel histograms in HSV color space for different skin-color face samples.



Table 5: Human face and mask color model backprojections obtained from learning the chrominances of the faces' central subregion.

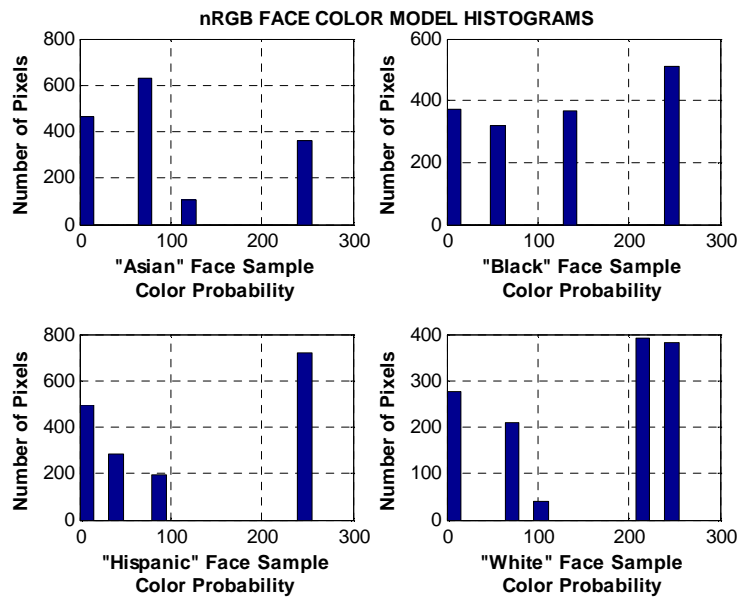


Figure 23: Histograms of the nRGB chrominance probability distributions in the face samples (values from 0 to 255).

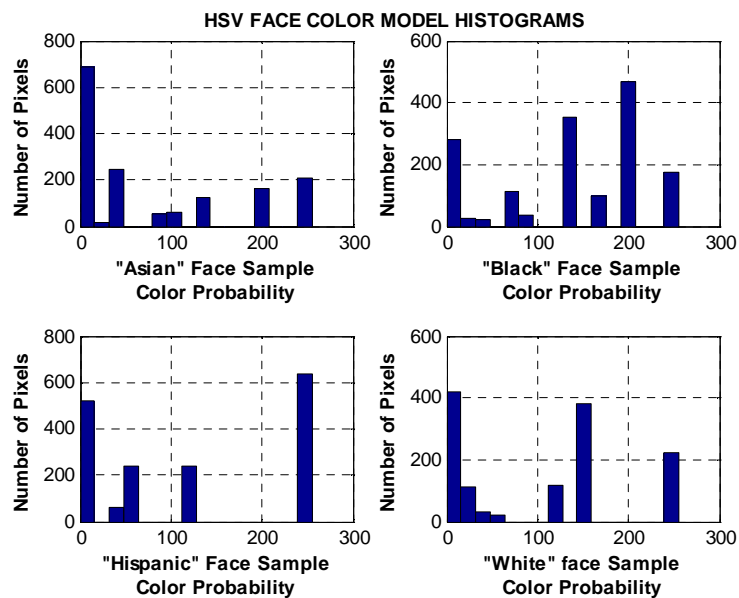


Figure 24: Histograms of the HSV chrominance probability distributions in the face samples (values from 0 to 255).

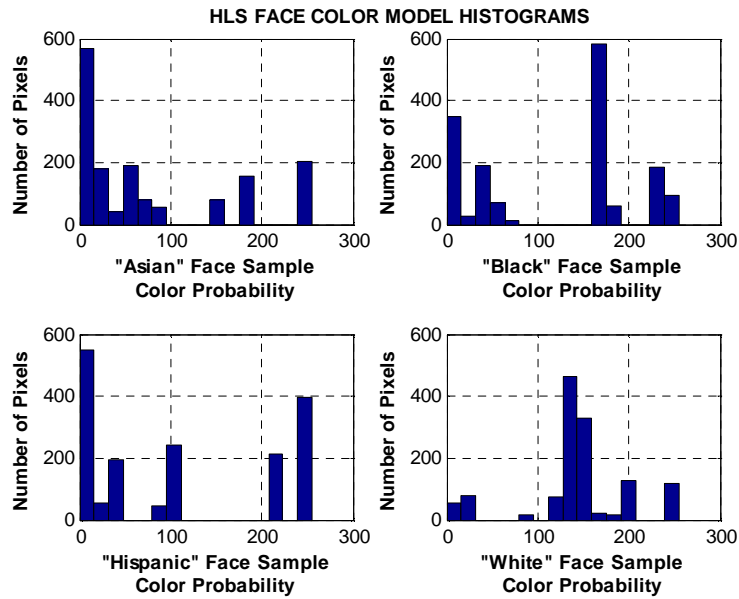


Figure 25: Histograms of the HLS chrominance probability distributions in the face samples (values from 0 to 255).

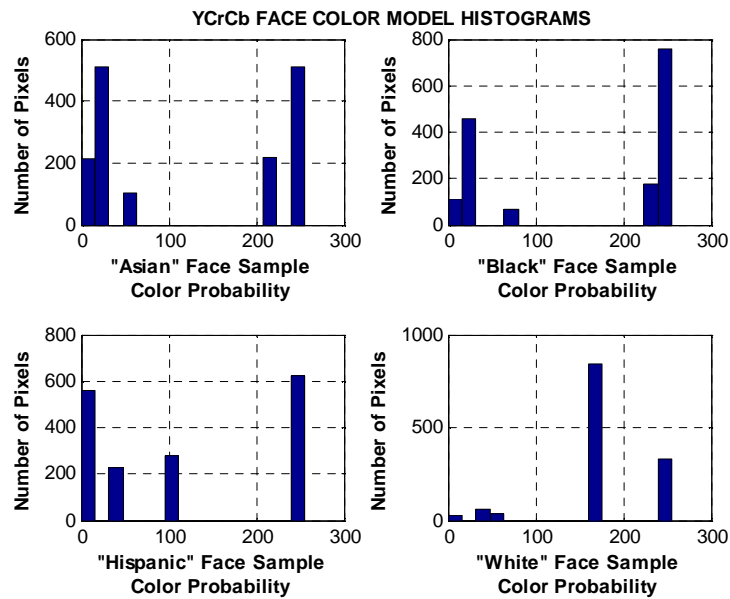


Figure 26: Histograms of the YCrCb chrominance probability distributions in the face samples (values from 0 to 255).

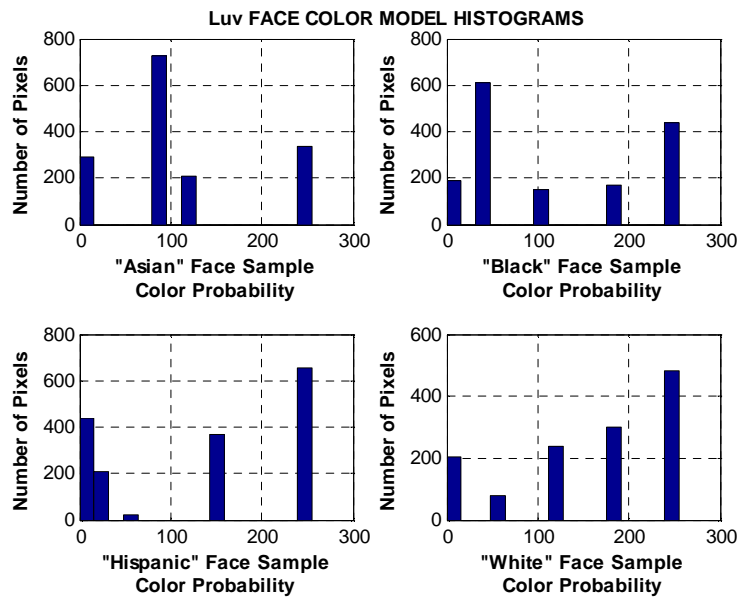


Figure 27: Histograms of the Luv chrominance probability distributions in the face samples (values from 0 to 255).

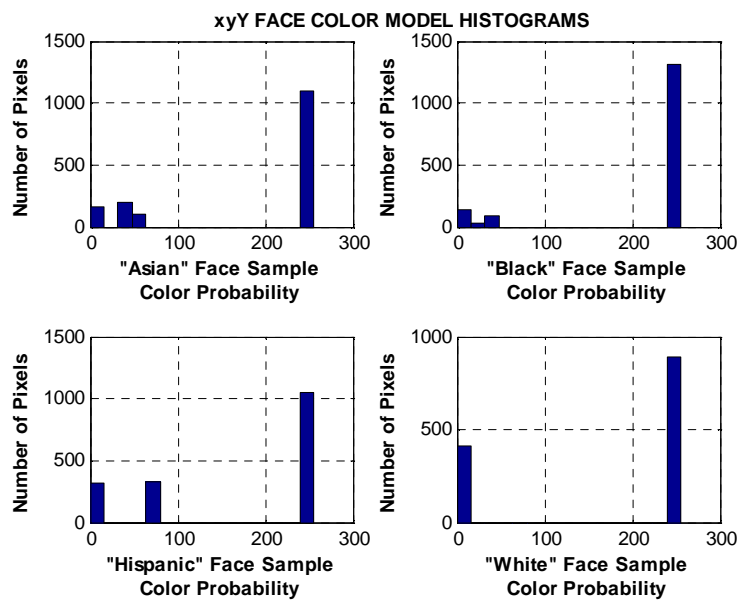


Figure 28: Histograms of the xyY chrominance probability distributions in the face samples (values from 0 to 255).

The following observations can be attained from these figures:

- Hair can clearly be seen in the backprojections of HLS (*White* sample), YCrCb (*White* sample), Luv (*Asian* and *White* samples) and xyY (*Asian*, *Black* and *Hispanic* samples), while it can also be seen, but with less intensity, in nRGB (*Asian* and *White* samples). This is not a desired characteristic as hair is not the face from which color has been learned.
- HLS obtains the darkest face backprojections. The brighter the better for correcting feature positions.
- The most uniform and the brightest face probability distributions are those of xyY, Luv and YCrCb, which is a desired characteristic for correcting feature positions.
- The white color trimming of masks is clearly seen in the xyY backprojections, which leads to improper feature corrections as they do not correspond to the face but rather to other objects.
- Red and yellow masks are the most visible ones, especially in HLS, YCrCb, Luv and nRGB backprojections, which was foreseeable taking into account the face hues observed in Table 4. Again, this leads to improper feature corrections as they do not correspond to the face.

Additionally, Table 6 shows the false positives obtained in the mask backprojections for the color models expressed by different color spaces by counting the median number of pixels with a minimal probability of corresponding to the learned color distribution. It is shown that HSV obtains the least amount of false positives.

Color Space	nRGB	HSV	HLS	YCrCb	Luv	xyY
Median Number of Pixels	231	165	1023	1747	501	3789

Table 6: False positives in the scene chrominance spotting (number of pixels in the backprojections of masks). The median is obtained from a set of frames with artificial random noise added to pixel colors.

Taking into account all these observations, it can be deduced that HSV is the most proper color space for correcting the feature positions of the CFOF procedure, as it leads to the least amount of false positives, including hair, and it also leads to an acceptable color distribution, even though it is not the best distribution from among those analyzed.

The next step is to evaluate the face tracking performance of CFOF. It is done by comparing CFOF with other alternative tracking approaches: (a) by replacing the PyrLK optical flow with those of Lucas-Kanade (LK) (Lucas and Kanade 1981), Horn-Schunck (HS) (Horn and Schunck 1981) and Block Matching (BM) (Gyaourova et al. 2003), (b) by skipping the Kalman filter (KF), (c) by ignoring the optical flow, i.e., by combining only the CMDF and the KF, (d) by removing the minimal color constraint for features, which is the same as transforming the CMDF into a Maximum Distance Filter (MDF), (e) by using a modified version of the well-known Pfinder (mPfinder) (Wren et al. 1997), and (f) with the also well-known Condensation (CONDitional DENSity propagATIOn) or particle filtering algorithm (Isard and Blake 1998).

The Pfinder approach tracks a blob in four steps:

1. Predict its appearance in the new image using the current state of the blob model. Simple Newtonian dynamics are assumed.
2. Predict for each image pixel measure the likelihood that it is a member of the blob model. The likelihood comprises both spatial and color information in which YCrCb (they call it YUV) color space is used. In order to avoid self-shadowing only the chrominance (CrCb) is used.
3. Resolve these pixel-by-pixel likelihoods into a support map. Spatial priors and pixel connectivity constraints are used.
4. Update the statistical model for the blob.

The modified version of the Pfinder (mPfinder) directly uses the previous state as the prediction instead of using Newtonian dynamics and also replaces the YCrCb color space with HSV for the obtention of the color probability. The first modification is made because it is intended to show the advantage of prediction (using the KF as all the other of the approaches) with respect to non-prediction, especially in the case of partial occlusions, and the second because of the conclusion reached previously regarding color spaces. Nevertheless, the used implementation of the KF works in a similar way as the Newtonian dynamics of the Pfinder.

In chapter 2, the Condensation algorithm was catalogued as a *silhouette tracking contour evolution* method, because it was originally used that way. However, it has also become a popular method for tracking body parts using other image-features, such as blob color and spatial cues (Pantrigo et al. 2005; Raducanu and Vitrià 2006; Shan et al. 2004), which handle shape changes more efficiently. This way, the particle filter performs a random search guided by a stochastic model in order to obtain an estimate of the posterior distribution describing the body part's position. Hence, as stated by Raducanu and Vitrià

(2006), its strength comes from its ability to simultaneously track non-linear/non-Gaussian multi-modal distributions, thereby solving the limitations imposed by the KF. Similarly, the state of a particle includes the position (u_x, u_y) and the linear velocity (v_x, v_y) of the pixel where it lies as shown in Equation 2. Equation 6 shows the likelihood function used in this test, which is obtained as the weighted sum of two probabilities: (1) the spatial probability of a particle calculated as its distance with respect to the center position of the ROI based on the normal distribution probability density function (Equation 7), and (2) its color probability in HSV color space. Attention must be paid to the parameter values taken by the former in order to make the spatial probability be between 0 and 1. The latter can take two values: 0, when its HS chrominance does not reach the considered minimal value, and 1 otherwise.

$$\hat{p}_{particle} = (1 - w_c) \hat{p}_{space} + w_c \hat{p}_{color} \quad (0 \leq w_c \leq 1) \quad (6)$$

$$\hat{p}_{space} = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(x-ROI_x)^2 + (y-ROI_y)^2}{2\sigma^2}} \quad (7)$$

Table 7 shows the parameter values used for the algorithms mentioned in the tracking comparison. The minimal color probability is set low (only 1%) as it is intended to make the system more sensitive to false color positives in order to evaluate the different approaches in noisy environments. In the case of the Condensation algorithm, the search window size represents the upper and lower bounds around the ROI position and velocity, where the particles are randomly spread. If the area where particles may lie is too high, i.e., considerably higher than the face size, it may occur that if another blob with similar color characteristics is around the face, a set of particles may lie on it and degrade the tracking. For instance, Figure 29 shows how, if the tracked face passes near another face, the particles can lie on it and the ROI fails to track its objective. Therefore, Condensation parameters are adjusted for this not to occur in that case.

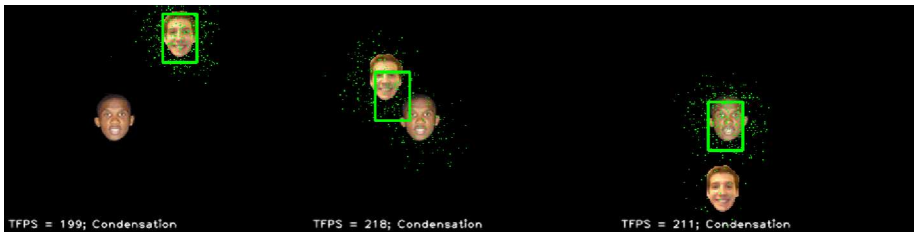


Figure 29: An example of how too great a search area in Condensation may degrade the tracking if there is more than one blob with similar color characteristics.

Common Parameters	Color Space	HSV
	Minimal Color Probability (%)	1
CFOF (CMDf) General Parameters	Number of Flock Features	10
	Maximum Distance to Median (pixels)	50
LK Parameters	Window Size (pixels)	5
HS Parameters	Lambda Lagrangian Multiplier	0.1
BM Parameters	Block Size (pixels)	10
	Shift Size (pixels)	1
	Maximum Range (pixels)	2
PyrLK Parameters	Window Size (pixels)	10
	Number of Pyramid Levels	3
KF Parameters	Process Noise Covariance (pixels)	2
	Measurement Noise Covariance (pixels)	1
Condensation Parameters	Number of Particles	400
	Search Window Size (pixels)	20
	Position Noise Amplitude (pixels)	10
	Velocity Noise Amplitude (pixels/frame)	5
	Distance Dispersion from <i>ROI</i> Center, σ (pixels)	20
	Color Probability Weight (w_c)	0.5
	Minimal p_{particle} for <i>ROI</i> Center Calculation (%)	70

Table 7: Parameters considered for the tracking performance comparison.

Thus, Table 8 shows the face tracking results for the tracking approaches considered in both free path and path with obstacles (Figure 30) scenarios. The attained maximum speeds have been measured for the four face samples, *Asian* (A), *Black* (B), *Hispanic* (H) and *White* (W), and the mean values derived from them have also been included. These maximum speeds correspond to those in which the tracking is done with stability, i.e, for several turns, and are given in pixels/frame, which means that the face has moved from frame to frame the stated Euclidean distance in pixels. If a result is marked with a hyphen "-", it means the tracking, for some reason, got lost during the test. It normally occurs in the case of the paths with occlusions because the tracker has not been able to move across the masks. In these cases, the tendency in the mPfinder + KF approach is to lose tracking by tracing the tangent to the circle, especially in those regions with more false positives, such as the red and yellow masks. This

occurs because the KF is limited to linear assumptions, and as the traced path is not, and the mPfinder relies on spatial and color configurations, the false positives make them trace tangents to the path. The tendency in the CMDF variants is to get stuck in these regions. Meanwhile, the Condensation algorithm may fail in both ways. The mPfinder + KF is more prone to failure in the masks scenario than the rest, and therefore the speeds have been marked with an asterisk “*” to reflect that.

Tracking Method	Filters	Maximum Speed in Free Path (pixels/frame)					Maximum Speed During Partial Occlusions (pixels/frame)					Median Time per Frame (ms)
		A	B	H	W	Mean	A	B	H	W	Mean	
CFOF with LK	MDF (+ KF)	1 (1)	2 (2)	2 (2)	2 (2)	2 (2)	- (-)	- (-)	- (-)	- (-)	- (-)	12.66 (12.66)
	CMDF (+ KF)	16 (19)	22 (22)	18 (20)	14 (17)	18 (20)	6 (8)	- (-)	10 (12)	6 (7)	6 (7)	12.66 (12.82)
CFOF with HS	MDF (+ KF)	1 (1)	1 (1)	1 (2)	1 (1)	1 (1)	- (-)	- (-)	- (-)	- (-)	- (-)	76.92 (83.33)
	CMDF (+ KF)	14 (18)	22 (22)	17 (19)	10 (16)	16 (19)	5 (7)	- (-)	10 (12)	5 (7)	5 (7)	83.33 (83.33)
CFOF with BM	MDF (+ KF)	2 (2)	2 (2)	2 (2)	2 (2)	2 (2)	- (-)	- (-)	- (-)	- (-)	- (-)	1.08 <i>(1.30)</i>
	CMDF (+ KF)	17 (19)	20 (22)	19 (20)	17 (18)	18 (20)	2 (9)	- (-)	9 (14)	8 (10)	5 (8)	<i>1.36</i> (1.58)
CFOF with PyrLK	MDF (+ KF)	49 (51)	50 (49)	48 (48)	43 (42)	48 <i>(48)</i>	- (-)	- (-)	- (-)	- (-)	- (-)	1.64 (2)
	CMDF (+ KF)	50 (51)	50 (50)	48 (48)	43 (43)	48 <i>(48)</i>	7 (13)	- (-)	13 (17)	12 (15)	8 (11)	1.63 (2.06)
No Flow	CMDF + KF	26	27	26	25	26	22	23	22	20	22	1.89
mPfinder	-	52	54	53	52	53	-	-	-	-	-	6.67
mPfinder	KF	46	47	48	47	47	6*	-	8*	8*	6*	6.76
Condensation		16	17	16	16	16	10	7	9	10	9	4.90

Table 8: Face tracking results. Highlighted values correspond to the best three mean maximum speeds and computation times (the best in bold font and the other two in italics).

It can be seen that the best optical flow method for our purpose is the PyrLK as it is capable of working at much higher speeds than the rest. It can also be seen that the inclusion of the KF is beneficial for the trackers, especially in the mask scenario, even though the top speed among the fastest methods is only slightly decreased. Apart from the KF, the minimal color constraint included in the CMDF has also proven to be beneficial for the tracking along the obstacle path. Moreover, the combination of the KF and the CMDF without optical flow obtains the highest speeds during occlusions.

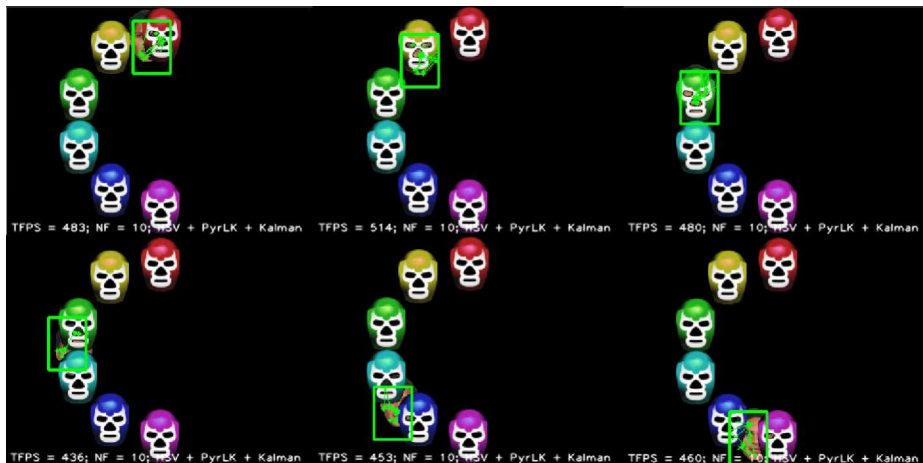


Figure 30: Face tracking (*Hispanic* sample) during partial occlusions using CFOF with PyrLK + CMDF + KF.

The overall fastest methods in the free path scenario correspond to the mPfinder + KF and the CFOF with PyrLK + CMDF + KF approaches. Both approaches have a low computational cost, even though the CFOF is faster. This is an important issue taking into account that five body parts must be tracked with this method simultaneously, and also that a further full-body pose reconstruction and eventual motor action recognition must also be computed for HCI.

Figure 31 shows the trajectories of these two methods running at a high speed in the free path scenario. It can be observed that both methods track the face with little difference and low noise with respect to the real path. The offsets between the real and the tracked positions come mainly because the center of the face does not necessarily need to be the centroid of the skin color (in the case of mPfinder + KF) and the centroid of the feature flock. Nevertheless, the traced trajectories are sufficiently similar to those of the real paths for further motor action recognition.

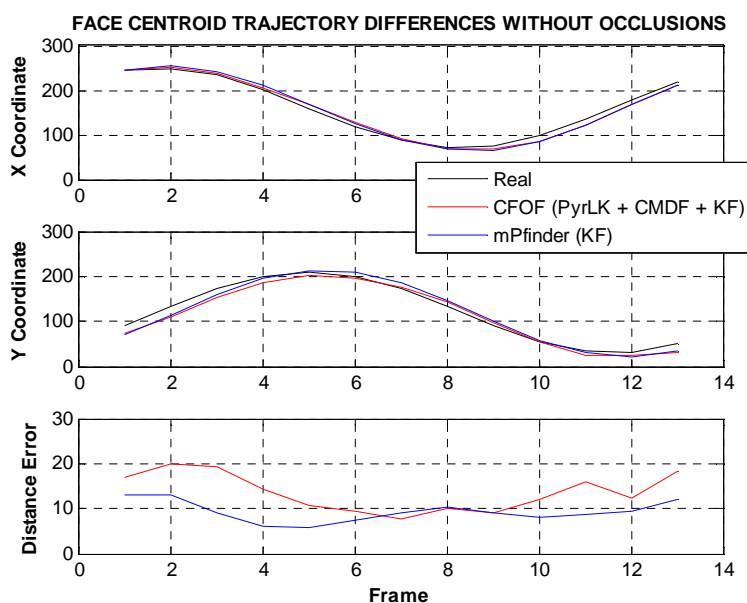


Figure 31: Face (*White* sample) trajectory differences running at 43 pixels/frame without occluding masks using CFOF (PyrLK + CMDF + KF) and the mPfinder (KF).

On the other hand, the overall fastest methods in the path with obstacle scenario correspond to the CFOF with PyrLK + CMDF + KF and the Condensation approaches. The latter algorithm is especially interesting when there are no other regions in the screen with color characteristics similar to those of the tracked object, because it is possible to increase the random search area and hence attain the highest tracking velocities even with severe occlusions. However, this is not the case for the full-body tracking proposed in this thesis project, as hands and face share similar color characteristics, as also the feet do (because the user will probably wear shoes with the same color). Additionally, due to the random search, the *ROI* position obtained with Condensation is much noisier than when using CFOF with PyrLK + CMDF + KF, and this constitutes a problem for further motor action recognition.

Figure 32 shows the trajectories of these two methods running at top speed in the scenario with obstacles. It can be observed, as is the case with the free path scenario, that both methods track the face with little difference with respect to the real path. But in this case, there are more disturbances due to the partial occlusions that prevent the face from being totally visible along the path with masks. Again, the traced trajectories are sufficiently similar to those of the real paths for further motor action recognition.

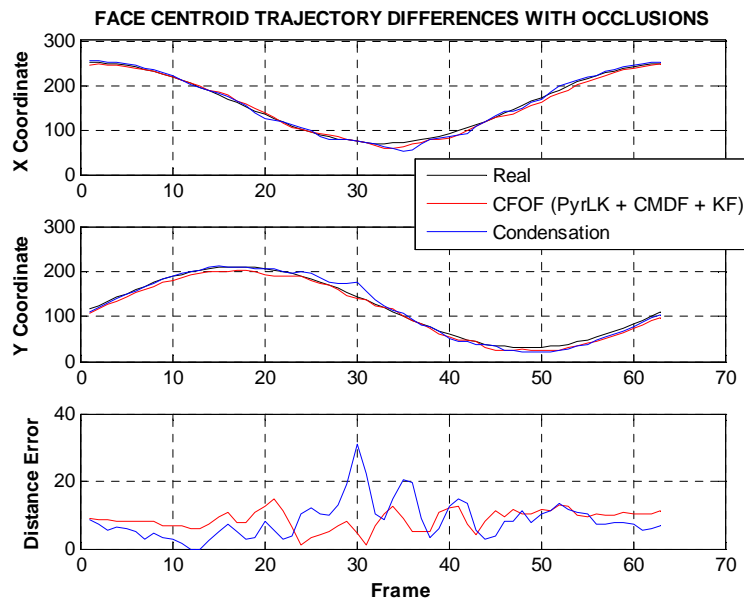


Figure 32: Face (*White* sample) trajectory differences running at 9 pixels/frame with occluding masks using CFOF (PyrLK + CMDF + KF) and Condensation.

It must be stated that the CFOF and Condensation approaches work with more stability through obstacles than the mPfinder, even though the Condensation approach is noisier. On the other hand, the CFOF approach is capable of tracking body parts attached to other body parts with similar color characteristics, such as hands when forearm skin is also visible, unlike the mPfinder and the Condensation algorithms (Figure 14). Therefore, since the CFOF with PyrLK + CMDF + KF approach obtains fast tracking results allowing the tracking through partial occlusions with low computational cost, it can be stated that it is a satisfactory method for tracking body parts in HCI applications.

CHAPTER 4

HUMAN FULL-BODY POSE RECONSTRUCTION

Human motion is typically represented as a series of different configurations of a rigid multibody mechanism consisting of a set of segments connected by joints. Segments correspond to body parts such as the thighs, shanks, upper arms, forearms, etc., and joints correspond to articulations such as hips, knees, shoulders, elbows, etc. These joints are hierarchically ordered and have one or more Degrees of Freedom (DoF) which represent the rotations relative to their parent joints. There is a root joint of which the position and orientation are represented with respect to the absolute coordinate system. A good example of this way of representing humanoid characters is the H-Anim standard (H-Anim 2008). This standard places the root joint at the pelvis, and defines a standard name for each joint, as well as a standard reference, or *neutral*, posture. This thesis project adheres to this standard.

In the 3D *top-down end-effector driven* pose reconstruction approach presented in this work, as stated in chapter 3, the known data correspond to the tracked positions of hands, head, feet and pelvis, i.e., end-effectors. Nevertheless, the pose reconstruction approach presented in this chapter also considers the possibility of adding more input data, such as the orientations of hands, head, feet and pelvis, and also the positions of elbows and knees, in case these are available. Thus, this method can also be used for the control of avatars by animators, which can easily generate humanoid pose databases for their use as a basis in: (a) static motor action recognition and (b) depth warping

of tracked body parts instead of maintaining their depth constant in a single-view motion capture, as will be shown in chapter 5.

It is often too cumbersome and time-consuming for an animator to manually set all the DoFs of a virtual character. This is solved by Inverse Kinematics (IK) techniques, in which only the positions (or sometimes also the orientations) of certain joints, usually the end-effectors, must be specified by the animator or by the motion capture system. The remaining DoF are automatically determined according to different criteria that depend on the IK variant one employs. End-effector positions can be modeled as a function of the DoFs, leading to formal definition of the IK problem as $f(\mathbf{q}) = \mathbf{G}$, where \mathbf{q} is the vector of DoFs and \mathbf{G} is a vector that gathers all the desired end-effector positions. This problem is highly under-constrained as \mathbf{q} usually has a much larger dimension than \mathbf{G} . In addition, it is a nonlinear problem as f involves complex combinations of trigonometric functions.

This chapter is organized as follows: firstly, a method for the human 3D full-body pose reconstruction in real time is presented. In order to achieve the reconstruction of this mechanism an approach, called Sequential Inverse Kinematics (SIK), is presented in which IK methods are applied to the spine and the upper and lower limbs. Secondly, a method to define and apply the biomechanical rotation restrictions to reduce the search space of solutions to the biomechanically plausible ones is described. Then, techniques designed for collision avoidance between the humanoid and the environment and for handling with self-collisions are presented. Finally, the performance of SIK is evaluated by comparing it to other well-known robot IK methods, with satisfactory results. Additionally, examples of the use of SIK in marker-based systems, in which the rotations of the end-effectors and the positions of elbows and knees are available, are reported. The reduced dimension of the input data and the low computation cost make the approach interesting for applications that require low-cost performance animation capabilities. Also, as no pre-recorded motion database is used, the memory footprint of the method is minimal.

4.1 SEQUENTIAL INVERSE KINEMATICS

There have been different approaches to solve the IK problem that can be distinguished as: *analytical*, *numerical* and *hybrid* methods. Analytical methods find all possible solutions as a function of the lengths of the mechanism, its starting posture and the rotation constraints. Their advantages are their low computational cost and good accuracy compared to other methods and their

drawbacks are that they can only be used for low DoF mechanisms and that they are not feasible when the system is ill-posed.

Some examples of analytical solutions of multibody mechanisms are the ones proposed by Zoppi (2002), Wu et al. (2004) and Gan et al. (2005). On the other hand, numerical methods cover those that require a set of iterations to achieve a satisfactory solution. In this case we can find methods such as the work of Zhao and Badler (1994) in which they propose to search for a plausible solution by solving a constrained nonlinear optimization. Another numerical approach is that in which the nonlinear problem is linearized using the Jacobian matrix where at each iteration an update of the DoF is obtained. There are different strategies for this update like the Jacobian Transpose method (Balestrino et al. 1984; Wolovich and Elliot 1984), the Pseudoinverse method (Whitney 1969), the Damped Least-Squares (DLS) (Nakamura and Hanafusa 1986; Wampler 1986), the DLS with Single Value Decomposition (SVD) (Maciejewski 1990; Maciejewski and Klein 1988) and the Selectively Damped Least Squares (SDLS) (Buss and Kim 2005). A review on these strategies can be found in (Buss and Kim 2005). The numerical approach can be enhanced by enforcing priorities to arbitrate the fulfillment of conflicting constraints such as in the Online Motion Retargeting (OMR) (Choi and Ko 1999) and the Prioritized IK (PIK) (Baerlocher and Boulic 2004). The main difference between OMR and PIK is that OMR has only two levels of priority while PIK can have any number making the latter more suited to our purpose. The potential of PIK for full-body motion capture was explored by Peinado et al. (2004). Raunhardt and Boulic (2007) use PIK for the reconstruction of human spines. They use both equality and inequality constraints to model the coupling behavior of the spine and reduce the search-space, achieving natural spine shapes. An alternative to this approach for reconstructing the spine is the work of Boulic et al. (2004). It is an approximate method for distributing a relative thorax/pelvis orientation on a set of vertebrae according to their anatomic behavior. It is intended to provide fast qualitative results, which suffice in the synthesis of walking animations.

Other numerical methods are based on neural nets and artificial intelligence such as those proposed by Oyama et al. (2001) and D'Souza et al. (2001) in which the mechanism rearranges learned movements to reach the targets. A related approach is to reconstruct postures based on a database of prerecorded motions, as in the recent works of Grochow et al. (2004), Chai and Hodgins (2005) and Liu et al. (2006). The drawback of methods that rely on databases is that if desired postures are too distant from those of the database, odd results are obtained. Another approach is the Cyclic Coordinate Descent (CCD) algorithm (Wang and Chen 1991) where the joints of a kinematic chain

are rotated one by one starting from the root a certain step-angle reducing the difference between the end-effector's current position and orientation and a full iteration is performed when all the joints have been rotated.

Finally, hybrid methods are those that combine both analytical and numerical algorithms such as those proposed by Tolani et al. (2000) for upper and lower limbs, which we will refer to as the TGB (Tolani-Goswami-Badler) method, and for the human full-body reconstruction such as the method proposed by Shin et al. (2001). Kulpa et al. (2005) use CCD and TGB to readapt, in real-time, pre-recorded animations to certain constraints such as feet-ground contact. They apply these algorithms separately in the different body parts in which they subdivide the humanoid; the head, the two arms, the two legs and the trunk. Their work can also be used for full-body reconstruction if, instead of using the postures of an animation for their readjustment, a neutral posture such as a standing pose is used as a starting posture. We will refer to this approach as the KMA (Kulpa-Multon-Arnaldi) method.

There have been previous studies in which known global orientations are used for motion reconstruction such as (Molet et al. 1999; Monheit and Badler 1991) for the spine, (Zordan and Hodgins 1999) for the upper-body, and (Badler et al. 1993a; Semwal et al. 1998) for the full-body. Nevertheless, we are more interested in adjusting the poses to known positions instead of only orientations, since it is more intuitive for an animator to situate them or for a mocap system to track them. This opens up new possibilities for the simple control of avatars and for markerless human motion capture based in computer vision.

Thus, the main idea of the pose reconstruction approach presented in this thesis project is that the reconstruction is solved sequentially, using simple analytical-iterative IK algorithms in different parts of the body in a specific order. No pre-recorded motion database is necessary, thereby avoiding the need for extra memory. First the orientation of the root joint is estimated from the known positions. The configuration of the spine is found using a hybrid IK method that combines this estimated orientation with the positions of the root and head markers. Then the orientations of clavicles are determined with the positions of their corresponding known end-effector positions and the already positioned spine. Finally, each of the limbs is situated according to their known end-effector positions with an analytic IK method. Complex biomechanical rotation limits are modeled from only a few known anatomical data to constrain the joint orientations and prevent elbows from penetrating the torso in order to obtain visually plausible human poses. Algorithm 7 shows the general procedure of this approach. Its novel aspects are described in the

following sections. Note that *wrists* and *ankles* are mentioned in this algorithm instead of hands and feet, because in most humanoids these joints are the true end-effectors of their skeleton's arms and legs, which can be controlled for IK purposes. Nevertheless, there may be cases, such as those in which the mesh of the body segments are included, where the geometry of hands and feet can be used to match the user's tracked real hand and feet positions. These cases are handled in the same way as when wrists and ankles are used directly for IK, so in the rest of the chapter they will be treated equally. Besides, for HCI applications the difference between, e.g., hand and wrist positions is not significant, as it could be in the case of a biomechanical analysis of an arm's movements for sports, or for medical purposes.

Algorithm 7 Sequential Inverse Kinematics

```

1: procedure SIK( $pos_{pelvis}$ ,  $pos_{head}$ ,  $pos_{wrists}$ ,  $pos_{ankles}$ )
2:   Estimate  $rot_{pelvis}$  from  $pos_{pelvis}$ ,  $pos_{head}$ ,  $pos_{wrists}$  and  $pos_{ankles}$ 
3:   Reconstruct the spine from  $rot_{pelvis}$ ,  $pos_{pelvis}$  and  $pos_{head}$ 
4:   Reconstruct the legs from  $pos_{ankles}$ 
5:   Reconstruct the clavicles from  $pos_{wrists}$ 
6:   Reconstruct the arms from  $pos_{wrists}$ 
7: end procedure

```

4.2 SPINE RECONSTRUCTION

The reconstruction starts with the torso and involves two steps. In the first one we estimate the orientation of the pelvis, which we do not know since we only consider the positional information of the end-effectors and pelvis. This differs from (Monheit and Badler 1991) where the known data are the orientations instead of the positions. In the second step we find a suitable configuration for the spine from the knowledge of both end points.

For the first step, taking into account the H-Anim specification for axes definition, we propose to estimate the pelvis orientation as follows:

- Y direction defined by the vector that goes from the pelvis to the head which is subsequently normalized.
- X direction defined by a weighted average of three vectors, the first one from the right wrist to the left one, the second one from the right ankle to the left one, and the third one, the X axis of the previous frame pose, all projected and normalized in the plane whose normal is the Y direction.
- Z calculated as the cross-product of X and Y .

This novel approach to estimate the global orientation of the body is applicable to a broad range of different movements with visually satisfactory results. The weights used for the calculation of the X axis (called respectively m_1 , m_2 and m_3) control the dependency level of the torso's axial rotation with respect to the positions of the upper and lower end-effectors. The third vector is used to increase the rigidity of its movements. Once the orientation of the pelvis (root joint) is established, one can proceed to solve the reconstruction of the spine bearing in mind that the objective is to achieve a visually acceptable result, not an exact solution.

As a simple example to visualize the philosophy of this approach, consider a totally straight spine composed of 5 equally spaced joints and where only the positions of the end-effectors are known. A visually acceptable and reasonable solution for the reconstruction of a human-like spine of this type would be the one shown in Figure 33 where both end-effectors get closer to each other. In this 2D solution, the root joint or pelvis is rotated by φ_{root} and the other joints are rotated in the opposite direction with an equal amplitude φ so that the last segment orientation is $-\varphi_{\text{root}}$. This solution can be represented analytically for n joints with Equations 8 and 9. Due to the non-linearity of the latter equation the value of φ is obtained iteratively.

$$\varphi = \frac{2\varphi_{\text{root}}}{n-2} \quad (8)$$

$$D = L[\cos \varphi_{\text{root}} + \cos(\varphi_{\text{root}} - \varphi) + \cos(\varphi_{\text{root}} - 2\varphi) + \dots \\ \dots + \cos(\varphi_{\text{root}} - (n-2)\varphi)] \quad (9)$$

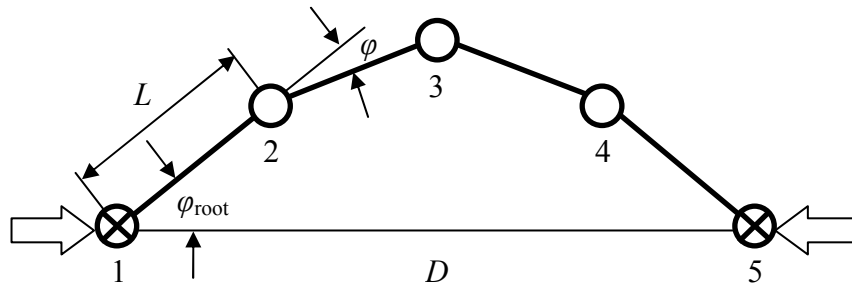


Figure 33: The readjustment of an equally distributed straight spine with 5 vertebrae.

However, real human spines do not have the same length for all vertebrae and they are not fully extended in their resting posture. Despite these facts experiments show that we are able to obtain visually acceptable postures of n vertebrae spines with the procedure that appears in Algorithm 8, which is

based on the above mentioned solution. The plane in which the spine would be bent is that composed of the known (or goal) and current (or previous frame's) positions of the pelvis and the head, i.e., the normal of this plane would be that obtained from the cross product of two vectors; one formed with the known positions of the pelvis and head and the other with its current positions. The plane calculated this way allows rotating the spine in any direction, not only forward-backward. In case of alignment, the previous frame's bending plane is used. In this algorithm, L_{obj} is the distance between the known positions corresponding to the root joint and the head joint. $L_{current}$ is the length measured between these joints of the virtual humanoid for each iteration. $L_{current}$ is intended to reach L_{obj} . Finally, $L_{neutral}$ is the same measure but when the joints are in their neutral or resting posture.

In human-like spines there is a curvature on its neutral posture so if our known end-effector positions are too distant the spine must be stretched out completely. The process to calculate these stretching postures is the same as the one presented (by means of Equations 8 and 9), where the joint rotation limits are those in which their corresponding segments are aligned with the vector that goes from the pelvis to the head. This process could result in excessive a computational cost compared with the rest of the posture calculations. This occurs because the stretching postures are close to the singular configuration of the spine, and adjusting the angles is not very efficient for lengthening the spine. Therefore, we can use a set of precomputed stretching postures to speed up this transition and so that there is a unique way of stretching the spine. Then, if after stretching completely the spine we come back to reachable positions, the spine would not recover its original shape again for $L_{current} = L_{neutral}$. For this reason, when $L_{current} > L_{neutral}$ we must set the spine in the neutral posture for further calculations. The iterative process to get a satisfactory value of $L_{current}$ when $L_{neutral} > L_{obj}$ is shown in Algorithm 9.

Algorithm 8 Spine Reconstruction Procedure

```

1:  procedure SPINERECONST( $pos_{pelvis}$ ,  $pos_{head}$ ,  $rot_{pelvis}$ )
2:     $PelvisJointPosition \leftarrow pos_{pelvis}$ 
3:     $L_{obj} \leftarrow |pos_{head} - pos_{pelvis}|$ 
4:    if  $L_{neutral} > L_{obj}$  then
5:      if  $L_{current} > L_{neutral}$  then
6:         $AllJointOrientations \leftarrow IdentityRot$ 
7:      end if
8:      Rotate joints until satisfactory  $L_{current}$  is obtained with
        SATISFACTORYLENGTHSEARCH( $L_{current}$ ,  $L_{obj}$ ,  $L_{neutral}$ )
9:    else
10:     if Spine not totally stretched then

```

```

11:         Get the corresponding pre-recorded pose
12:     end if
13: end if
14: Rotate pelvis to align its X axis with that of  $rot_{pelvis}$ 
15: Set the spine joints within biomechanical limits if necessary
16: Rotate pelvis to align the vector that connects head and pelvis
    joints with vector  $pos_{head} - pos_{pelvis}$ 
17: Translate pelvis along vector  $pos_{head} - pos_{pelvis}$  to distribute the
    positional error of the end
18: end procedure

```

This method is employed because it is impossible to express in simple terms the explicit equations in the 3D spine for every bending plane. If the step angle is too small the process could become too slow. For this reason the iterative process is done in two phases. Firstly, it is solved by a coarse step angle until it "overshoots" a bit and then we iterate "back the other way" with a refined step angle until it also "overshoots" a tinier bit. An example of the readjustment of a complete human spine can be seen in Figure 34.

Algorithm 9 Satisfactory $L_{current}$ Search

```

1: procedure SATISFACTORYLENGTHSEARCH( $L_{current}$ ,  $L_{obj}$ ,  $L_{neutral}$ )
2:   if  $L_{current} > L_{obj}$  then
3:      $sign = +1$ 
4:   else
5:      $sign = -1$ 
6:   end if
7:   while ( $sign \times L_{current}$ ) > ( $sign \times L_{obj}$ ) do
8:     Rotate pelvis the corresponding step angle ( $\approx -sign \times 0.1$  rad)
9:     Rotate remaining joints according Equation 8
10:    Update  $L_{current}$ 
11:   end while
12:   while ( $sign \times L_{current}$ ) < ( $sign \times L_{obj}$ ) do
13:     Rotate pelvis the corresponding step angle ( $\approx sign \times 0.01$  rad)
14:     Rotate the rest of the joints according to Equation 8
15:     Update  $L_{current}$ 
16:   end while
17: end procedure

```

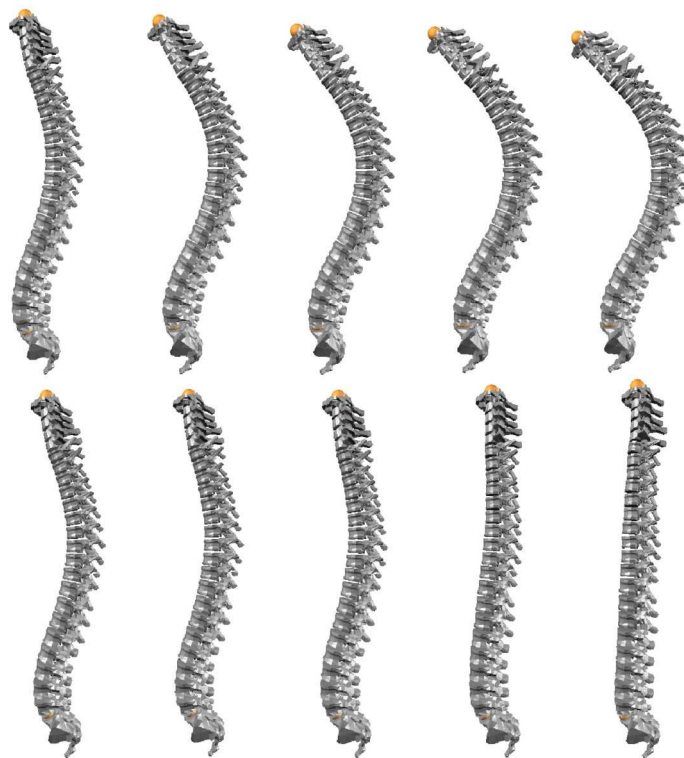


Figure 34: The resulting readjustment of a complete spine moving on its sagittal plane from the known positions of head and pelvis. On the top row, a bending movement, and on the bottom row, a stretching movement.

Finally, it must be stressed that this methodology focuses on contexts in which only one position is known for the pelvis and one for the head (this is case of the markerless approach presented in chapter 3). As a consequence, there is no reliable means for evaluating the relative twist between the pelvis and the head. For this reason such angular twist is not handled in Algorithms 8 and 9. However, in case we can also measure the pelvis and head orientations (and also maybe the neck basis orientation, concretely $w7$ joint) we can distribute the relative twist quantity along the spine as shown in Algorithms 10 and 11, which must be applied after Algorithms 8 and 9. If the neck basis orientation measurement is not available it is considered to be the same as that of the head. At this point it must be remarked that at every iteration the joints are set within biomechanical limits in case it is necessary.

Due to their very limited mobility along the twist axis, the lumbar vertebrae would probably transgress the biomechanical limits at every iteration. So the algorithm could be adapted to take advantage of such a priori

knowledge by distributing the twist rotation only in the thoracic and cervical regions. To conclude this discussion, we have observed that the visual influence of this lack of relative twist is not very critical compared to other features of the pose as evaluated in section 4.7.1. Figure 35 shows an outline of a spine region twisting procedure on the left, and on the right, a full-spine twisting example having as known data the orientations of its end-effectors. The red, green and blue facets of the cubes placed on end-effectors of the right image represent their X , Y and Z axes orientations. It can be observed how vertebrae are gradually twisted in order to make the spine's end-effectors fit the known data.

Algorithm 10 Spine Twisting Procedure

```

1:  procedure SPINETWISTING( $pos_{pelvis}$ ,  $pos_{head}$ ,  $rot_{pelvis}$ ,  $rot_{head}$ ,  $rot_{neck}$ )
2:    Get known axial status vectors:
       $U_{spine} \leftarrow (pos_{head} - pos_{pelvis}) / |pos_{head} - pos_{pelvis}|$ 
       $X_{pelvis} \leftarrow$  first column of  $rot_{pelvis}$ 
       $X_{neck} \leftarrow$  first column of  $rot_{neck}$ 
       $X_{head} \leftarrow$  first column of  $rot_{head}$ 
       $projUX_{pelvis} \leftarrow U_{spine} \times (X_{pelvis} \times U_{spine}) / |U_{spine} \times (X_{pelvis} \times U_{spine})|$ 
       $projUX_{neck} \leftarrow U_{spine} \times (X_{neck} \times U_{spine}) / |U_{spine} \times (X_{neck} \times U_{spine})|$ 
       $projUX_{head} \leftarrow U_{spine} \times (X_{head} \times U_{spine}) / |U_{spine} \times (X_{head} \times U_{spine})|$ 

3:    Get current pelvis axial status vector:
       $X_{pJoint} \leftarrow$  first column of  $PelvisJointOrientation$ 
       $projX_{pJoint} \leftarrow U_{spine} \times (X_{pJoint} \times U_{spine})$ 

4:    if  $|projX_{pJoint}| > \epsilon$  then ( $\epsilon \approx 0.01$ )
5:       $projUX_{pJoint} \leftarrow projX_{pJoint} / |projX_{pJoint}|$ 
6:      Rotate pelvis around  $u_{spine}$  to align its  $projUX_{pJoint}$  vector with
          its corresponding known  $projUX_{pelvis}$ 
7:    end if

8:    Get current neck axial status vector:
       $X_{nJoint} \leftarrow$  first column of  $NeckJointOrientation$ 
       $projX_{nJoint} \leftarrow U_{spine} \times (X_{nJoint} \times U_{spine})$ 

9:    Twist vertebrae from the joint next to pelvis up to the neck with
      SPINEREGIONTWISTING( $projUX_{neck}$ ,  $projX_{nJoint}$ )

10:   Get current head axial status vector:
       $X_{hJoint} \leftarrow$  first column of  $HeadJointOrientation$ 
       $projX_{hJoint} \leftarrow U_{spine} \times (X_{hJoint} \times U_{spine})$ 

11:   Twist vertebrae from the joint next to neck up to the head with
      SPINEREGIONTWISTING( $projUX_{head}$ ,  $projX_{hJoint}$ )

12:    $HeadJointOrientation \leftarrow rot_{head}$ 

13:  end procedure

```

Algorithm 11 Spine Region Twisting Procedure

```

1:  procedure SPINEREGIONTWISTING( $projUX_{end}$ ,  $projUX_{endJoint}$ )
2:    if  $|projX_{hJoint}| > eps$  then ( $eps \approx 0.0001$ )
3:       $projUX_{endJoint} \leftarrow projX_{endJoint} / |projX_{endJoint}|$ 
4:       $EndAng \leftarrow \cos^{-1}(projUX_{endJoint} \cdot projUX_{end})$ 
5:       $nIter = 0$ 
6:      while  $EndAng > twistError$  &  $nIter < nIterMax$  do ( $twistError \approx$ 
7:         $0.1$ ,  $nIterMax \approx 20$ )
8:        for spine region vertebrae do
9:           $U_{axis} \leftarrow$  unitary vector from vertebra to region's end
10:         joint
11:         Rotate vertebra around  $U_{axis}$  the angle
12:          $EndAng / (numberOfRegionVertebrae)$ 
13:         Set the vertebra within biomechanical limits if necessary
14:       end for
15:       Update  $EndAng$ 
16:        $nIter = nIter + 1$ 
17:     end while
18:   end if
19: end procedure

```

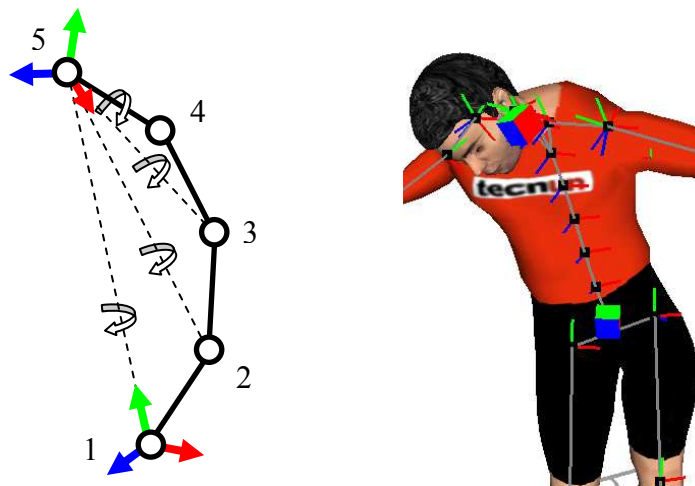


Figure 35: On the left, an outline of a spine region twisting procedure, and on the right, a full-spine twisting example where it can be seen how the vertebrae are gradually twisted in order to fit the end-effector orientations.

4.3 CLAVICLE RECONSTRUCTION

Once the spine is set, the positions of the sternoclavicular joints are also defined. We simplify the shoulder complex in a similar manner as Badler et al. (1993b), i.e., we consider that it is composed of only two joints, which, adhering to H-Anim terminology, are called sternoclavicular and shoulder. These joints are connected by the clavicle segment, resulting in a shoulder model that ignores the scapulothoracic articulation. In Badler’s book, the shoulder model of Otani (1989) is exploited in order to simulate the observed synergies between the sternoclavicular and the shoulder joints. Because this model expresses the relative distribution of arm elevation and abduction between these joints, it can only be implemented using the concept of joint group.

In our IK solver joint groups are avoided for efficiency reasons, and thus, instead of trying to apply Otani’s model we have chosen to apply a heuristic criterion of cost minimization: when the wrist “marker” can be reached using motion of the shoulder joint alone, the sternoclavicular joint remains in its current configuration. If, on the other hand, after positioning the shoulder joint the target remains unreachable, we change the configuration of the sternoclavicular joint so that the target is reached. This is consistent with the way the shoulder complex behaves, as clavicles tend to remain still unless (a) their motion is required for reaching the end-effector’s goal, or (b) they are deliberately actuated, which is out of our control given the scarcity of our input data. This approach for the reconstruction of a clavicle is given in Figure 36.

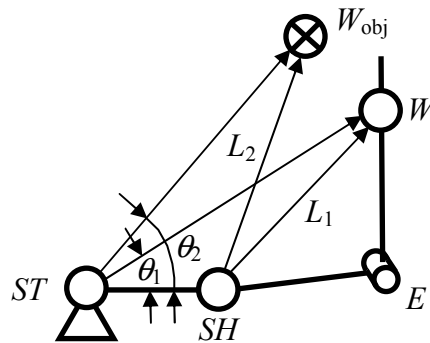


Figure 36: Diagram of the clavicle readjustment.

ST , SH , E and W are the sternoclavicular, shoulder, elbow and wrist joints, L_1 and L_2 are the lengths from SH to W and from SH to W_{obj} , respectively and W_{obj} is the end-effector’s known position. θ_1 and θ_2 are the angles between the clavicle segment and the vector that goes from ST to W and

between the clavicle and the vector that goes from ST to W_{obj} , respectively. The procedure for readjusting the clavicle is shown in Algorithm 12. An example is given in Figure 37.

Algorithm 12 Clavicle Reconstruction Procedure

```

1: procedure CLAVICLERECONST( $W_{obj}$ )
2:    $L_1 \leftarrow |W - SH|$ 
3:    $L_2 \leftarrow |W - ST|$ 
4:   if  $L_2 > L_1$  then
5:     Calculate  $\theta_1$  and  $\theta_2$  with the cosine theorem
6:     if  $\theta_2 > \theta_1$  then
7:        $V \leftarrow$  axis perpendicular to the  $ST\_SH\_W_{obj}$  plane
8:        $\alpha \leftarrow \theta_2 - \theta_1$ 
9:       Rotate  $ST$   $\alpha$  radians about axis  $V$ 
10:      Set  $ST$  within biomechanical limits if necessary
11:    end if
12:  end if
13: end procedure

```

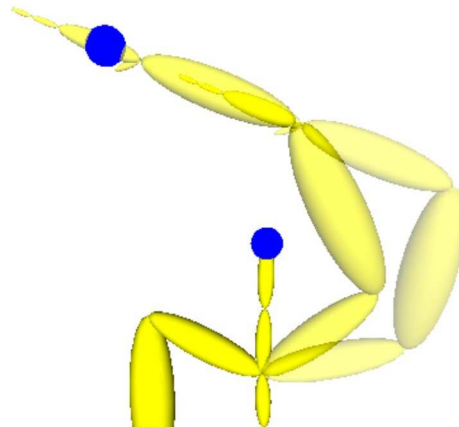


Figure 37: The resulting readjustment of a clavicle when its corresponding arm cannot reach the known wrist position by itself.

4.4 UPPER AND LOWER LIMB RECONSTRUCTION

Now that the postures of the central parts of the body have been determined the arms and legs can be adjusted to their corresponding end-effectors. It can be seen in the left image of Figure 38 how to calculate the ψ and φ angles analytically, in a similar way as in the TGB method. Regarding the swivel angle (right image of Figure 38), it is undetermined, but a reasonable value can be set

in the context of human pose reconstruction by, e.g., minimizing a cost function in order to attract joint A towards the mid-range of its biomechanical Euler angle limits. This procedure was done in the publications generated during this thesis project (Unzueta et al. 2005; Boulic et al. 2006). Using a similar approach, Kang et al. (2003) calculated a swivel angle that would minimize the torque. Due to the nonlinearity of its parameterization, these approaches require solving numerically an optimization problem.

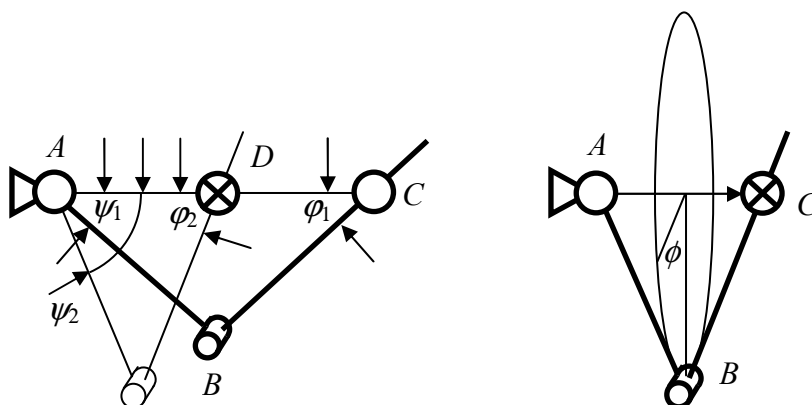


Figure 38: Diagram of the readjustment of a limb. On the left, the analytically calculable angles, and on the right, the undetermined swivel angle.

To find suitable configurations of the limbs we use Algorithm 13, which exploits biomechanical articular limits, prevents self-collisions and deals with non-reachable end-effector positions. Initially joint A is oriented analytically taking into account biomechanical limits and, in the case of the arms, the fact that the elbow cannot penetrate the torso. Then joint B is oriented, also analytically, to get the least possible error between the limb's end-effector's position and its known one. In this readjustment, apart from the biomechanical limits it is taken into account that the wrist cannot penetrate the torso, in the case of the arms, or that the foot cannot penetrate the floor, in the case of the legs. To avoid the computational burden of performing a numerical optimization to solve for the swivel angle, in our method we always start from the neutral configuration of the limb, i.e., totally stretched with joint A aligned to its parent. This yields reasonable postures, with the additional benefit of making the swivel angle independent of the postures in previous frames.

Algorithm 13 Limb Reconstruction Procedure

```

1:  procedure LIMBRECONST( $D$ )
2:      Rotate  $A$  to align  $AC$  and  $AD$  vectors
3:      if  $|AB| + |BC| > |AC|$  then
4:          Rotate  $A$  around  $X$  axis of  $B$  to match with  $\psi_2$ 
5:      end if
6:      Set  $A$  within biomechanical limits if necessary
7:      Prevent  $B$  interpenetration with the torso (only for the elbow)
8:       $V \leftarrow$  projection of  $BD$  vector in the  $XZ$  plane of  $B$ 
9:      Rotate axially  $A$  to align  $Z$  axis of  $B$  with  $V$ 
10:     Set  $A$  within biomechanical limits if necessary
11:     Rotate  $B$  around its  $X$  axis to align the  $BC$  and  $BD$  vectors
12:     Set  $B$  within biomechanical limits if necessary
13:     Prevent  $C$  interpenetration with the torso (only for the wrist) or
        with the floor (only for the ankle)
14:  end procedure

```

On contexts in which the position of joint B is known it is possible to determine the swivel angle of the limbs. It can be done by simply rotating the limb around AC vector, the angle necessary to align the projections of AB vector and the one that goes from A to the known position of B on the plane perpendicular to AC . This process should be done before step 6 of Algorithm 13.

On the other hand, if the orientation of joint C is available a further readjustment of the limb can be done. In the case of the arms, the wrist joint has two DoF corresponding to *flexion-extension* and *radial-ulnar deviations*, while the axial rotation of the hand (*pronation-supination*) comes from the forearm bones that orbit one another (Table 9). This latter rotation can be modeled in a simplified way by considering the axial rotation of elbow joint. Thus, the known wrist orientation can be set to the arm by: (1) calculating the XYZ follower Euler angles of the known wrist orientation referred to wrist joint system, (2) setting to the wrist joint an orientation made from those Euler_X and Euler_Z angles but with Euler_Y = 0, again in the form of XYZ follower Euler angles, (3) referring both the known and the current wrist orientations to the elbow joint system, and (4) rotating the elbow joint around the local Y axis of these matrices the angle required to align the current wrist local X axis with that of the known orientation. Meanwhile, ankles are considered to handle the three rotations on their own, so the known orientations can be applied directly

to them. Likewise previous cases, after rotating each of these joints they must be set within biomechanical limits.






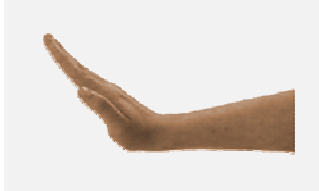
Ulnar Deviation (+ X)	Pronation (+ Y)	Flexion (+ Z)
		
Radial Deviation (- X)	Supination (- Y)	Extension (- Z)
		

Table 9: Right wrist rotations. Note that the pronation and supination movements come from the forearm bones that orbit one another, and not the wrist joint itself. Axes are modeled according to H-Anim specification (H-Anim 2008).

4.5 BIOMECHANICAL LIMITS

Grassia (1998) compared different parameterizations of rotations and concluded that, in general, no single parameterization is best. The use of a particular parameterization depends on its performance in the application. In principle, it is possible to limit the rotations regardless of the employed parameterization. For example, it could be possible to limit directly the Euler angles, but the achieved workspace would not be realistic for modeling the behavior of a human joint such as the shoulder. For this reason we propose a method to limit the orientation of the ball-and-socket joints similar to the spherical polygons used by Korein (1985) and reviewed by Baerlocher and Boulic (2000).

Three types of rotation ranges are distinguished: *swing*, *twist* and *simple flexion-extension* (Figure 39). Simple flexion-extension occurs only for the elbow and the knee joints. It is not difficult to limit this single DoF. On the other hand limiting swing and twist is not a straightforward task. Our purpose is to get visually pleasing results so joint couplings are not considered in order to simplify calculations.

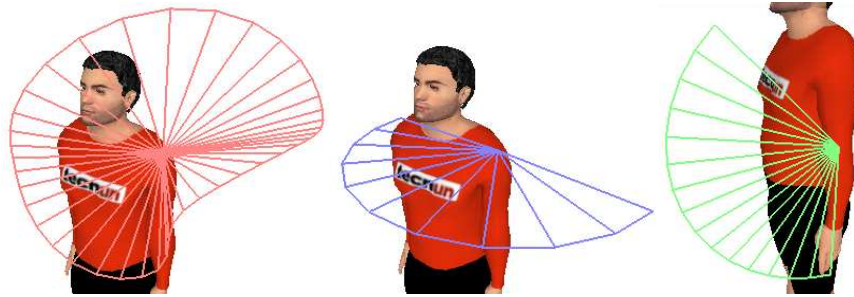


Figure 39: From left to right: swing and twist limits of a shoulder and flexion-extension limits of an elbow.

We model swing movements using a spherical parameterization of orientations. The segment that depends directly on the orientation of the joint has a constant size so the radius employed in the spherical coordinates is not taken into account. Only the other two angles, which we will call *circumduction angle* or θ , and *swing amplitude* or ψ , are considered.

The range of θ goes from -180° to $+180^\circ$, and for each value there is a corresponding biomechanical limit of ψ . In the spherical polygon representation the workspace boundary of the joint is defined by a set of vertices situated on the surface of a unit sphere connected by great arcs which would be the shortest path that bind two points on the sphere. The more vertices we have, the smoother and more realistic the boundary is, but more measurements of the swing limits will be needed. There are some studies of these measurements such as (Engin and Chen 1986; Herda et al. 2003).

Our alternative is to use a reduced set of biomechanical limits that can be obtained from books such as Kapandji's (1974; 1982; 1988); for example the maximal vertical flexion and extension of the shoulder, etc. and apply a cubic spline to these points to get a visually reasonable and smooth limitation of the swing movements (left image of Figure 39 and Figure 40). The first derivative of the spline at its starting and ending points ($\theta = -180^\circ$ and $\theta = 180^\circ$ respectively) is the same to get a smooth boundary over the entire circumduction movement. Its value is estimated with Equation 10,

$$\left(\frac{d\psi}{d\theta}\right)_1 = \left(\frac{d\psi}{d\theta}\right)_n = \frac{1}{2} \left(\frac{\psi_2 - \psi_1}{\theta_2 - \theta_1} + \frac{\psi_n - \psi_{n-1}}{\theta_n - \theta_{n-1}} \right) \quad (10)$$

where the subindices refer to the indices of a vector containing the known θ values, i.e., subindex 1 refers to the starting point and subindex n to the ending point. So to limit the swing amplitude of a joint we only have to calculate its current θ and ψ values. If ψ is above its higher bound, we simply

rotate the joint maintaining the same θ , until the maximum value of ψ is reached. In order to limit the twist or axial rotation we need a reference orientation to which the current orientation is compared. We propose to define this reference orientation by, first, considering the orientation of the parent joint as the neutral orientation of the current joint and, second, rotating it with the θ and ψ values corresponding to the current orientation. This way the reference orientation differs from the current one only on the twist rotation, so that the current orientation can be easily set within axial biomechanical limits. These limits vary depending on the swing orientation of the limb as was demonstrated by Wang et al. in (1998) for the case of the shoulders. For a visually acceptable result, though, the average values provided in the above mentioned books of Kapandji would be sufficient.

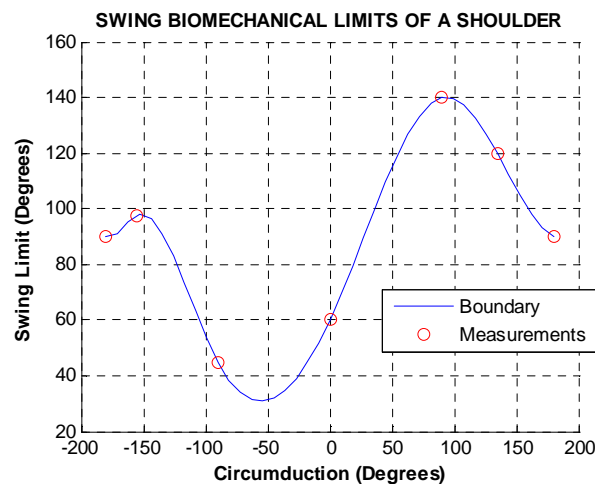


Figure 40: Modeling of the swing biomechanical limits of a shoulder with a cubic spline.

4.6 COLLISION AVOIDANCE

Biomechanical limits prevent the humanoid from performing many unnatural poses, but self-collisions and collisions with the environment can still occur. This is a complex problem with many different peculiarities. In this section three strategies are presented, which prevent three of the most common self-collisions and collisions with the environment in posture reconstruction; the penetration of (a) the elbows in the torso, (b) the wrists in the torso and (c) the feet in the floor. First the *penetration depth* (PD) must be estimated and then the proper response must be applied to resituate the colliding body parts.

There are many studies on PD calculations of colliding figures formed by meshes such as the different body parts of the humanoid that appears in Figure 39. Zhang et al. (2006) review the different approaches to achieve this objective and present a method to compute the PD taking into account possible rotations throughout the path in order to separate the overlapping objects. Nevertheless, these methods do not take into consideration that the limb constraints a multibody system subject to for repositioning in case of collision. The strategy to avoid undesired collisions must be integrated within the IK algorithm as it is shown in the works of Kallmann (2005) and Peinado et al. (2006). In the present study we integrate the strategies to reorient the limbs in the SIK method as shown in Algorithm 13.

4.6.1 TORSO-ELBOW COLLISIONS

In this case, we focus our interest on a region comprising the upper arm and forearm mesh vertices that are near the elbow joint. For simplicity in Figure 41 we represent this section as a circle. We conceptualize the meshes of the segments that compose the torso as prismatic bodies, which is not far from reality in most human-like figures. Let \mathbf{p}_t be the outer vertex of the colliding torso segment, \mathbf{p}_e the deepest vertex of the region of interest of the arm, and P the XZ plane of the spine joint that contains the colliding torso segment. This way we define the depth of the elbow within the torso as the distance from \mathbf{p}_t to \mathbf{p}_e , in the direction that goes from \mathbf{p}_e to \mathbf{p}_t projected in P , and that passes through the elbow joint (see Figure 41 and left image of Figure 42). We not only obtain the depth value but also two points that represent the deepest point of the elbow (\mathbf{p}_e) and the outer point of the colliding torso segment (\mathbf{p}_t) which define the penetration direction.

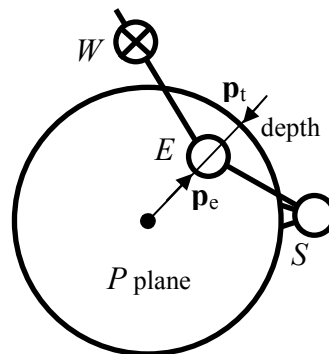


Figure 41: Diagram of the right elbow penetration depth estimation.

Once we have these measurements we are able to make the upper arm amend its orientation while we readjust the arm so that its end-effector matches the known position. We distinguish two cases. In the first one, the arm must be almost or fully stretched to reach the end-effector. Let v be vector that goes from the shoulder to the deepest point of the elbow, and w the one goes from the shoulder to the outer point of the colliding torso segment. These vectors are assumed to be normalized to length 1. In this case, we solve the elbow collision by rotating the upper arm about the axis defined by the cross product of v and w . This rotation is done iteratively. The angle applied on each iteration is given by $\arccos(v \cdot w)$. This value is a lower bound of the one that might be necessary to solve the collision, thus the need for performing several iterations.

In the second case, the one in which the arm is not stretched, we make the upper arm rotate around the swivel angle axis shown in the right image of Figure 38. Then as in the previous case we rotate iteratively until the elbow exits the torso. In this case it is not easy to determine a lower bound to the optimal angle. We propose rotating an angle which is k times the ratio between the depth and the length of the upper arm. We adjust the proportionality factor k empirically to a value of 1. In Figure 42 an example of a corrected arm posture following this strategy is given.

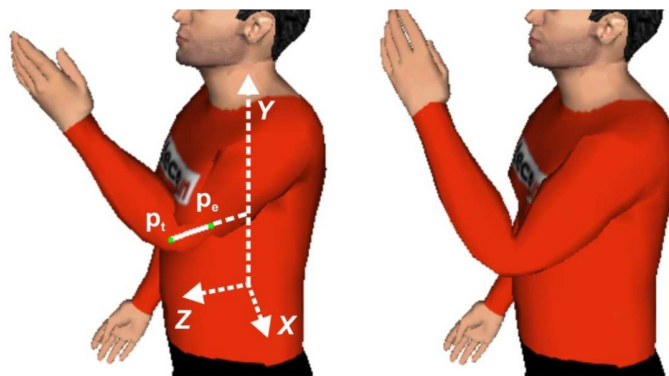


Figure 42: Left elbow torso penetration: on the left the estimated penetration depth and on the right the penetration avoidance response posture.

4.6.2 TORSO-WRIST COLLISIONS

The procedure to calculate the PD of the wrist within the torso is the same as in the case of the elbow, but focusing on the hand and forearm mesh vertices that are near the wrist joint (Figure 43). However, the response is more intricate as more DoF are involved in the repositioning of the arm. Again, an iterative process is applied until the wrist gets out of the torso, in which two

steps are undertaken: (1) a “correct” wrist position is estimated by adding the vector going from \mathbf{p}_e to \mathbf{p}_t to its current position, which will become the “objective” position of the iteration, (2) the shoulder joint is rotated in the same way as in the step 2 of Algorithm 13 towards the “objective”. Afterwards, in the case that the resulting elbow position makes it collide with the torso, the upper arm is reoriented as explained in section 4.6.1.

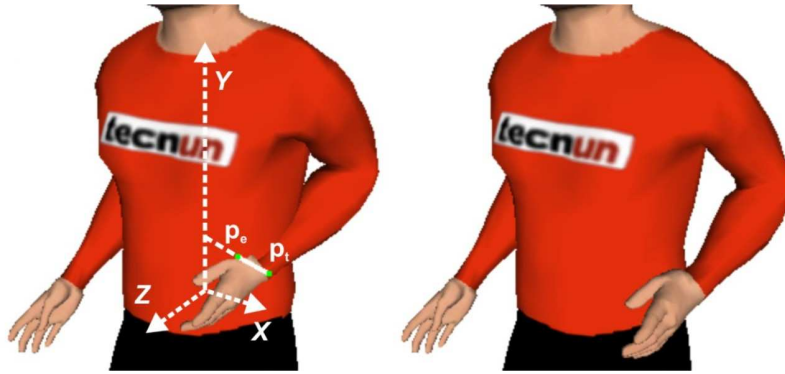


Figure 43: Left wrist-torso penetration: on the left the estimated penetration depth and on the right the penetration avoidance response posture.

4.6.3 FOOT-FLOOR COLLISIONS

Finally, the PD of the foot within the floor corresponds to the Y absolute coordinate of the deepest vertex of the colliding foot (Figure 44). The response to this collision is solved with one iteration by: (1) setting as the “objective” foot position the result from adding its current one plus the vector going from \mathbf{p}_e to \mathbf{p}_t , and (2) repositioning the leg with this “objective” using Algorithm 13, ignoring the biomechanical limits of the hip and knee joints.

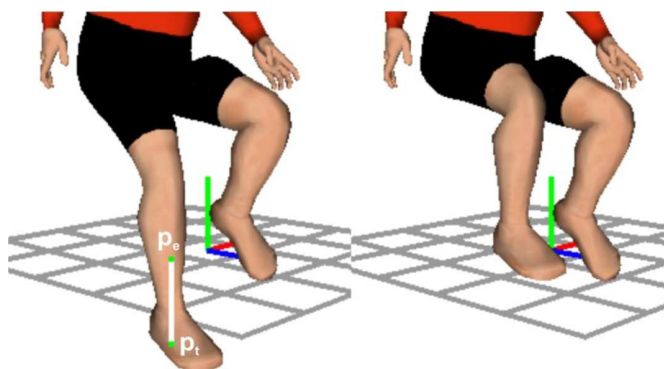


Figure 44: Right foot-floor penetration: on the left the estimated penetration depth and on the right the penetration avoidance response posture.

4.7 EXPERIMENTAL RESULTS

4.7.1 RECONSTRUCTION PERFORMANCE EVALUATION

In this section we compare the SIK method with other well-known approaches, specifically the KMA (Kulpa et al. 2005), CCD (Wang and Chen 1991), Jacobian Transpose (Balestrino et al. 1984; Wolovich and Elliot 1984), Pseudoinverse (Whitney 1969), DLS (Nakamura and Hanafusa 1986; Wampler 1986), DLS with SVD (Maciejewski 1990; Maciejewski and Klein 1988), SDLS (Buss and Kim 2005), PIK (Baerlocher and Boulic 2004; Peinado et al. 2004), and the TGB method for anthropomorphic limbs (Tolani et al. 2000) combined with the reconstruction methods presented here for the spine and clavicles. We call the latter TGBSIK. The test consists of reconstructing three sequences with different classes of motions in order to have available different swivel angles of the upper and lower limbs and axial orientations of the pelvis (root joint): the first one with 2,783 frames from typical boxing movements (Figure 45), the second one with 513 frames from a jump kick motor action (Figure 46), and the third one with 4,510 frames from a playground activity, which involves walking, climbing, hanging and swinging actions (Figure 47). These animations have been obtained from a marker-based mocap system (CMU 2008) and have not been filtered. This way robustness of the algorithms to noisy movements can be analyzed. This noise is more noticeable in the *Playground* animation.

First, the positions of the wrists, ankles, head and root joints are extracted, then the orientation of the root joint is estimated as explained in the spine reconstruction section and finally the different reconstruction methods are applied to a humanoid identical to the original. The weights employed for

the estimation of the root joint in the first two animations are $w_1 = 1$, $w_2 = 1$ and $w_3 = 0$, while for the third one are $w_1 = 0.1$, $w_2 = 0.1$ and $w_3 = 1$. This way the reconstructed walking motor action embedded in the *Playground* animation contains more realistic movements of the torso as its axial rigidity is increased with respect to the wrist and ankle vectors' oscillations.

In order to make the anthropometry configurable we distinguish the following parameters: *foot length*, *calf length*, *thigh length*, *pelvis length*, *pelvis width*, *torso length*, *torso width*, *neck length*, *upper arm length*, *forearm length* and *hand length*. Based on these parameters we define the height of the humanoid with the sum of the calf, thigh, pelvis, torso and neck lengths and we rescale it for the humanoids of the three sequences so that they are 1.86, 1.82 and 1.83 meters tall respectively.

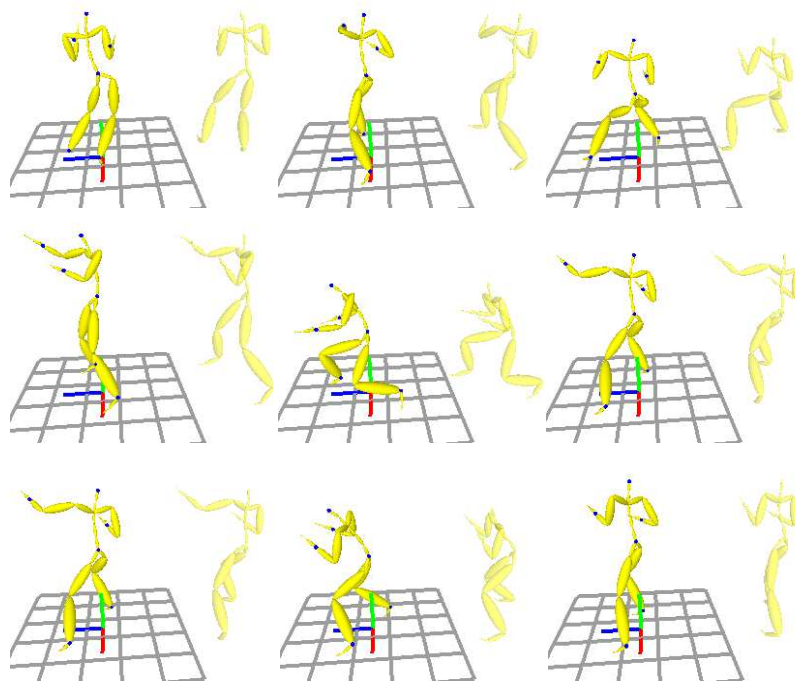


Figure 45: Samples of the *Boxing* animation: on the left of each sample the SIK reconstructions and on the right the original postures.

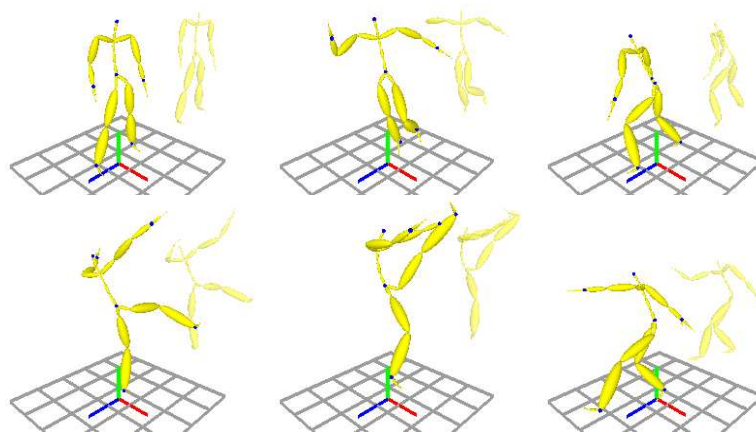


Figure 46: Samples of the *Jump Kick* animation: on the left of each sample the SIK reconstructions and on the right the original postures.

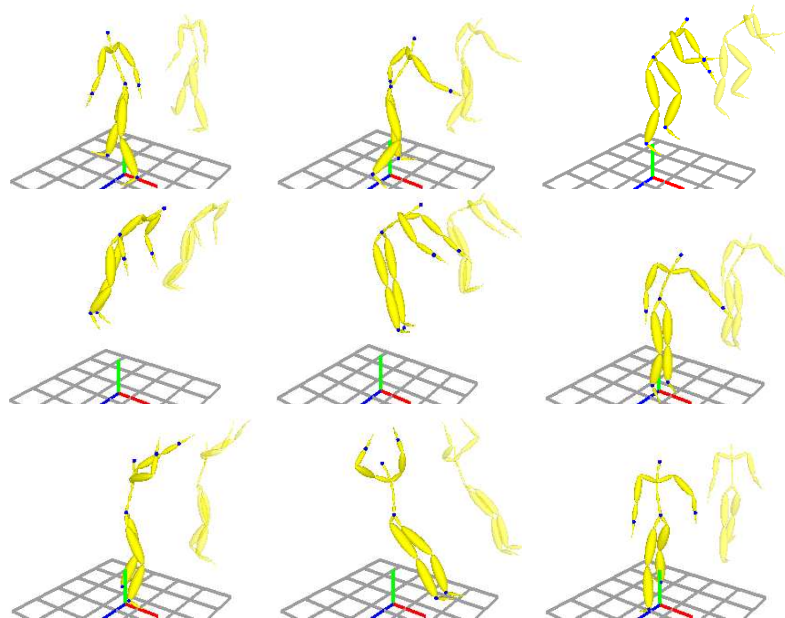


Figure 47: Samples of the *Playground* animation: on the left of each sample the SIK reconstructions and on the right the original postures.

All three humanoids have 34 joints, according to the H-Anim specification (Figure 48), and for this analysis, the relative movements of all the joints except those that derive from wrists and ankles are considered. In this

hierarchy *HumanoidRoot*, *sacroiliac* and *vl5* joints are distinguished in order to be compatible with the specification. But in reality, these joints are placed in the same position and they do not have relative motion. These humanoids do not have a mesh that defines their external shape so self-collisions are not considered in these reconstructions.

In the KMA method a constraint is associated to each known position or marker and their priority goes from top to bottom like this: root marker, head marker, ankle markers and wrist markers. The reconstruction of each frame starts always from the standing pose. First the root joint is placed. Then the CCD algorithm is applied to the spine. The joints are set within biomechanical limits at each step if necessary with the method presented here and the positional error of the end-effectors of the spine is evenly distributed as was explained in the spine reconstruction section. The iterative process stops when the difference between the end-effector's known and current positions are less than 1 cm or when that error does not improve from the previous iteration in more than 1 mm or when a maximum number of iterations, which we set at 20, has been reached. The clavicles do not perform any relative movement and the upper and lower limbs are situated using the TGB method with swivel angles equal to zero. Finally the joints of the limbs are set within biomechanical limits if necessary. The references for the swivel angles are:

- For the arms the $-Y$ axis of the estimated pelvis orientation. In case of alignment with the axis that connects the shoulder with the wrist it is changed to the $-Z$ axis of the estimated pelvis orientation.
- For the legs the $+Z$ axis of the estimated pelvis orientation. In case of alignment with the axis that connects the hip with the ankle it is changed to the $+Y$ axis of the estimated pelvis orientation.

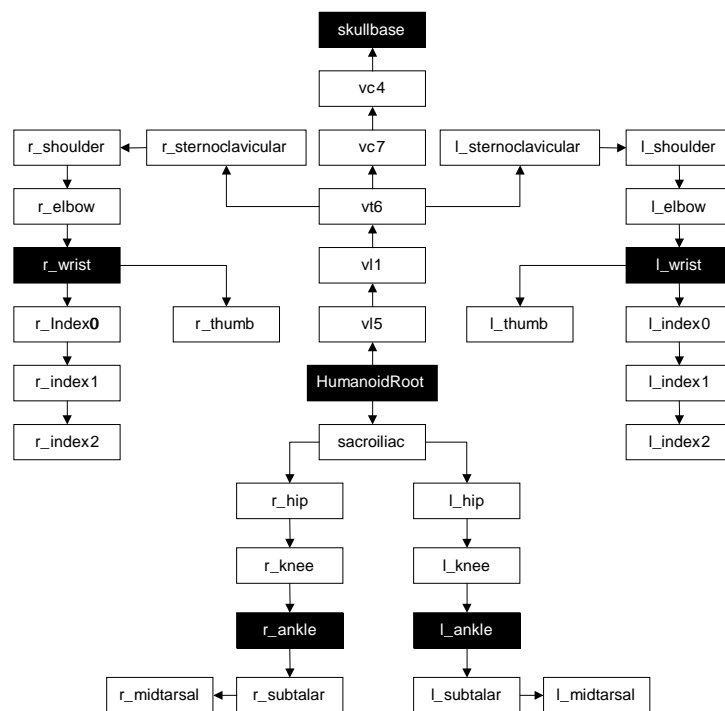


Figure 48: H-Anim structure (H-Anim 2008) of the humanoids of *Boxing*, *Jump Kick* and *Playground* animations. Highlighted joints are those whose positions are known.

There is a slight difference between the original KMA algorithm and this adaptation regarding the reconstruction of the spine. In the original KMA method the spine is subdivided into three segments (neck, torso and abdomen) to which the CCD algorithm is applied. The positions of the remaining vertebrae are then sampled from a spline that passes through the joints of these three segments. This procedure has been avoided in our test in order to simplify calculations; the reconstruction result does not differ significantly from the original KMA approach.

In the CCD method first the root joint is placed, then the CCD algorithm is applied separately, first to the spine, and then to each clavicle and arm together, and to the legs. The process is the same as in the spine of the KMA method. In this case the starting posture is the one of the previous frame.

For the Jacobian Transpose, Pseudoinverse, DLS, DLS with SVD and SDLS methods, the humanoid is modeled by decomposing the joints in 1 DoF articulations as if they represented the orientation like XYZ follower Euler angles. As in CCD, the biomechanical limits are checked at each iteration and the stop criteria is the same. Using these methods, the root is initially placed at its known position and after making each iteration the error in the end-effectors of the spine is redistributed as stated before. The damping factor of the DLS method is set to 75 to get stable reconstructions near singular poses. This factor has been determined experimentally and lower values give the humanoid jerky movements.

With the PIK method, priorities go from top to bottom like this: root marker, ankle markers, wrist markers and head marker. The root position is the most important because the remaining joints depend on it. The priority of ankles and wrists is changeable and the head is the least important in order to obtain the smallest degree of error since the mobility of arms and legs is higher than that of the spine. With this method the joint limits are solved by inequality constraints i.e limits are checked and if any have been transgressed, an equality constraint is added and the iteration is recalculated. Once all constraints have been accomplished, the swivel angles of arms and legs are determined by optimizing a cost function expressed in the joint space in order to attract the obtained posture to a standing pose in which the arms and legs are slightly bent; the elbows backwards and the knees forward. This way one can attain better convergence and a more relaxed posture, distant from singular configurations like those in which the limbs are totally stretched.

In the case of the TGBSIK method the swivel angle is optimized by minimizing the equation $f(\varphi) = swing^2 + weight \times twist^2$, where φ is the swivel angle and both swing and twist values of shoulders and hips are intended to be minimized to obtain the biomechanical configuration as far as possible from the joint limits. We use a weight of 1 in this function so both twist and swing have the same importance. As this function is non-linear and it is impossible to express the explicit equations in simple terms for each posture, we use the same optimization method as in the case of the spine, but in this case we look for the global minimum throughout the entire space, i.e., the whole 2π rotation of φ since there may be more than one local minima.

The performance of the IK methods is measured by comparing the differences in the positions and orientations of the joints of the reconstructions with respect to the original sequences, and by measuring the median computation time of one frame on the animations. The joints that derive from wrists and ankles are not considered because they are not controlled by IK algorithms. Therefore 22 joints are used for the error computations. The

position error is the Root Mean Square (RMS) value of the sum of the differences between the positions of the reconstructed and the original joints divided by the height of the humanoid in order to have an adimensional value independent from the subject's dimensions, and also divided by the number of joints to get an average error per joint. The orientation error is the RMS value of the sum of exponential map modules of the reconstructed joints orientation relative to the originals divided again by the number of joints. These orientations are computed with respect to the parent joint's coordinate system. The position error is more reliable than the orientation error because a little difference in angles of interior joints might lead to considerable visual differences but we also include the latter because focusing only on positions does not provide information about the differences in the orientations of the root and end-effector joints and the twist of the interior joints. RMS values are adequate because they compute the standard deviation of the distribution of the error around the desired value. An alternative to these measurements could be the cloud of points used by Kovar et al. (2002). The median computation time is computed instead of the mean value because it is more robust to outliers.

Figure 49 shows that the Jacobian Transpose method gives poor results due to the obtained jerky movements and because there is a significant difference between the end-effector positions with respect to their known values. This happens because this method is not appropriate to handle unreachable end-effector positions and because there is more than one target at the same time which is not suitable for this approach. For this reason this method is excluded for the remaining comparisons.

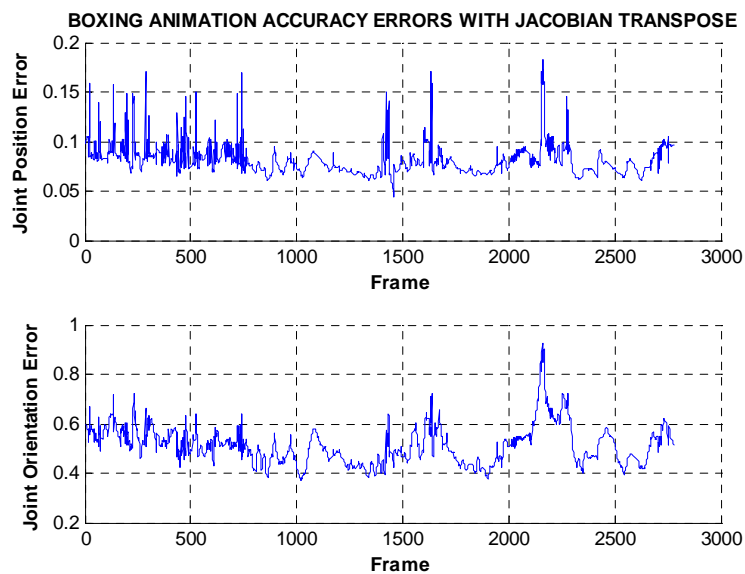


Figure 49: *Boxing* animation accuracy errors with Jacobian Transpose.

Table 10 shows the RMS values in the reconstruction quality and computation time averaged per joint. The system was implemented using C++, and tested on a 2.66 GHz Pentium 4 with 512 MB RAM. This table shows that there is a considerable difference of computation time between the SIK, KMA, CCD and TGBSIK with respect to others. This demonstrates that the processes to be undertaken by these methods are much faster at obtaining a satisfactory solution. With respect to the quality of the reconstruction, SIK, TGBSIK, PIK and KMA obtain better results. Their main differences, compared with the original sequences, come from the swivel angles of the limbs which is not dramatic since they still get natural poses for the movements considered. In terms of reconstruction quality, CCD presents two main drawbacks. First, the swivel angle of the upper limbs is biomechanically correct but becomes unnatural as the motion progresses. The second drawback is in the motion of the spine, which resembles that of a snake rather than a human spine. This is because bending starts from the head, so joints closer to that end-effector in the hierarchy are moved to a greater extent than those further away. Figure 50 shows that quality errors tend to increase as the motion advances in the *Boxing* animation. This occurs because bending in the iterations starts from the end-effectors of the chains so the motions of the joints nearest to them are greater.

IK Methods		KMA	CCD	PsInv	DLS	DLS _{SVD}	SDLS	PIK	SIK	TGBSIK
Joint Average RMS Position Error (normalized)	<i>Boxing</i>	<i>0.0281</i>	0.0316	0.0303	0.0325	0.0325	0.0293	<i>0.0283</i>	0.0265	0.0312
	<i>Jump Kick</i>	0.0256	0.0292	0.0260	0.0291	0.0293	<i>0.0245</i>	0.0247	<i>0.0213</i>	0.0208
	<i>Playground</i>	0.0295	0.0275	0.0275	<i>0.0256</i>	<i>0.0257</i>	0.0265	0.0240	0.0261	0.0266
	Average	0.0277	0.0294	0.0279	0.0291	0.0292	0.0268	<i>0.0264</i>	0.0246	<i>0.0262</i>
Joint Average RMS Orientation Error (radians)	<i>Boxing</i>	<i>0.2524</i>	0.4376	0.3157	0.3115	0.3113	0.3060	0.3256	0.2508	<i>0.2937</i>
	<i>Jump Kick</i>	0.3175	0.4651	0.3413	0.3331	0.3336	0.3119	0.3447	0.2849	<i>0.2912</i>
	<i>Playground</i>	0.3452	0.4606	0.3786	0.3492	0.3495	0.3626	<i>0.3472</i>	<i>0.3469</i>	0.3475
	Average	<i>0.3050</i>	0.4544	0.3452	0.3313	0.3315	0.3268	0.3392	0.2942	<i>0.3108</i>
Median Time per Frame (milisec)	<i>Boxing</i>	<i>1.11</i>	<i>0.87</i>	17.46	11.45	20.63	15.38	28.56	0.44	1.89
	<i>Jump Kick</i>	<i>1.02</i>	<i>1.02</i>	22.36	8.31	15.71	17.09	39.54	0.45	1.92
	<i>Playground</i>	<i>1.13</i>	<i>0.78</i>	14.87	8.28	14.42	12.82	28.40	0.43	1.90
	Average	<i>1.09</i>	<i>0.89</i>	18.23	9.35	16.92	15.10	32.17	0.44	1.90

Table 10: Full-body reconstruction methods comparison. Highlighted values correspond to the best three for each row (the best in bold font and the other two in italics).

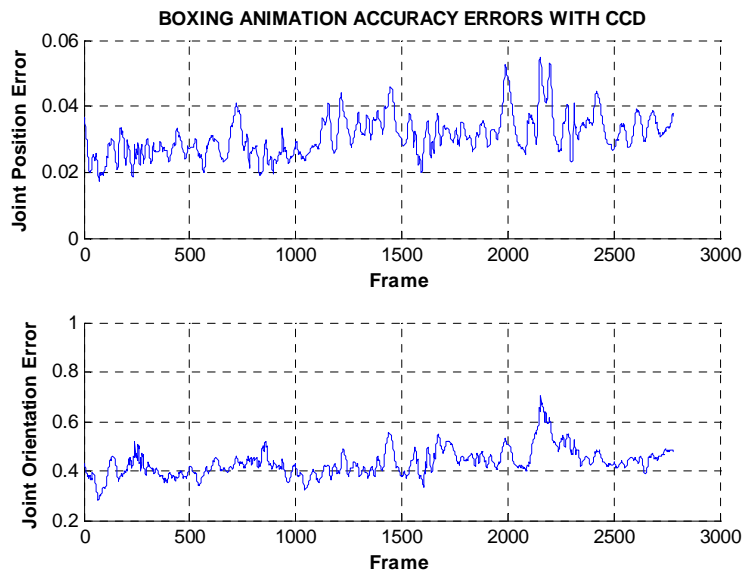


Figure 50: *Boxing* animation accuracy errors with CCD.

The comments on CCD for the spine are partly applicable to the KMA approach because the latter is based on the CCD. Nevertheless, the well known "snake-like" artifact is not so noticeable since the KMA reconstruction always exploits the resting posture instead of the previous frame posture. Regarding the limbs, the KMA approach uses the TGB method, but without performing any optimization procedure, unlike TGBSIK. This is why it is faster than TGBSIK.

In the Pseudoinverse, DLS, DLS with SVD and SDLS methods the main drawback comes from the orientation of the clavicles, which tend to bend more than would appear natural. This is because in each iteration the angle of each articulation is updated simultaneously using the same rigidity to rotation. The main difference between them comes from the jerkiness when the limbs are near singular postures, which is more noticeable in the *Playground* animation due to its inherent noise. Among these, the DLS with SVD and the SDLS methods give the most stable results. The PIK method, which relies on the DLS with SVD, obtains more natural orientations of the clavicles because the pose is attracted to the relaxed standing posture mentioned above. The TGBSIK method is a bit slower than SIK and KMA in positioning the anthropomorphic limbs because an optimization is being performed. But the reconstruction is also visually acceptable even though the swivel angles are more distant from the original movements. Its main drawback is that there are some frames in which the swivel angle changes abruptly because the minimization based on biomechanical limits does not yield totally smooth results, mainly due to the fact that for a given joint the global minimum is not continuous throughout the workspace. This type of abrupt changes also occur with the KMA method in the *Playground* animation because of its high variety of movements and the discontinuity in the swivel angle references.

Figure 51 shows the quality errors in the *Boxing* animation for all the considered IK methods except for the Jacobian Transpose from frames 1,500 to 2,500, Figure 52 shows similar results for the *Jump Kick* animation from frames 100 to 500, and Figure 53 the same for the *Playground* animation from frames 1,500 to 2,500. The maximal errors in the first two cases occur when the humanoid performs a quick turn-around movement, where one of the limbs is totally stretched trying to reach its known end-effector position (a punch in the *Boxing* animation and the kick in the *Jump Kick* animation). This maximal value is caused by the differences between the actual pelvis orientation and its estimation, and also because the end-effector is out of range for its corresponding limb. In the third animation maximal errors happens again while the humanoid performs turnaround movements while walking. These facts

show that a satisfactory spine positioning is decisive for an appropriate full-body reconstruction quality.

Finally, Figures 54-56 present excerpts of the reconstructions in which the differences mentioned above can be appreciated. All these results show that the SIK method is a fast and reliable approach for our purpose, yielding satisfactory results both in terms of quality and computation time, which make it appropriate for HCI applications.

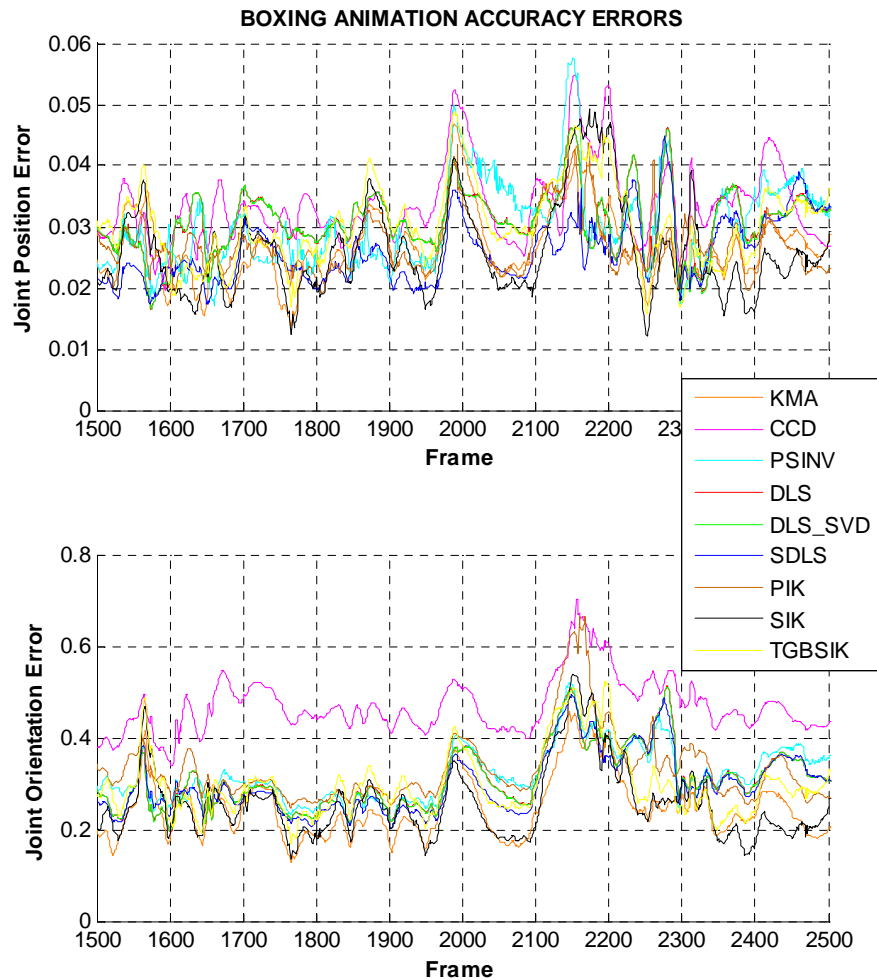


Figure 51: *Boxing* animation accuracy errors.

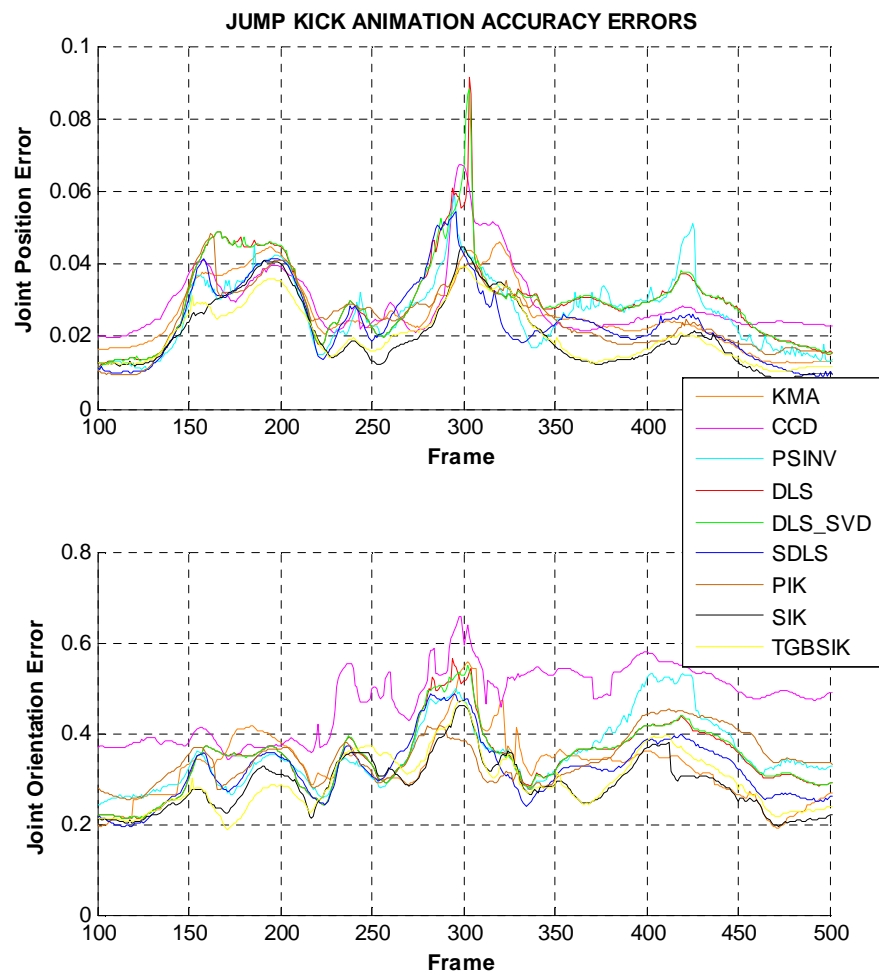


Figure 52: *Jump Kick* animation accuracy errors.

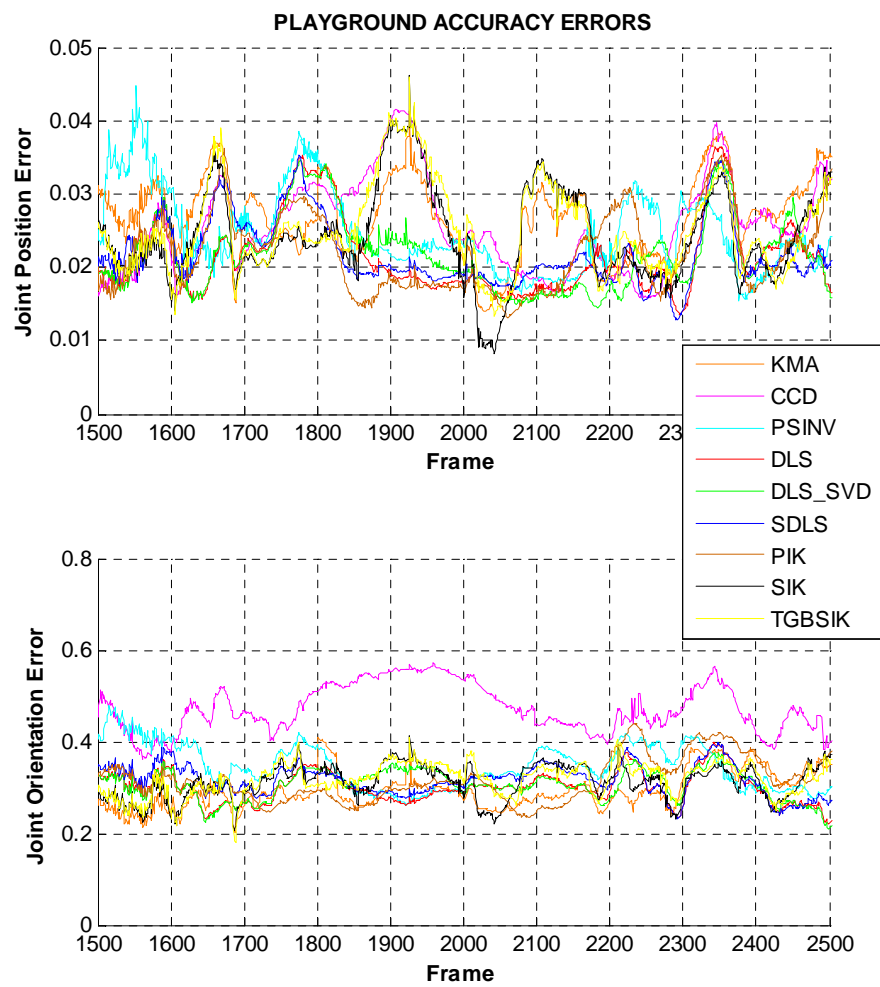


Figure 53: *Playground* animation accuracy errors.

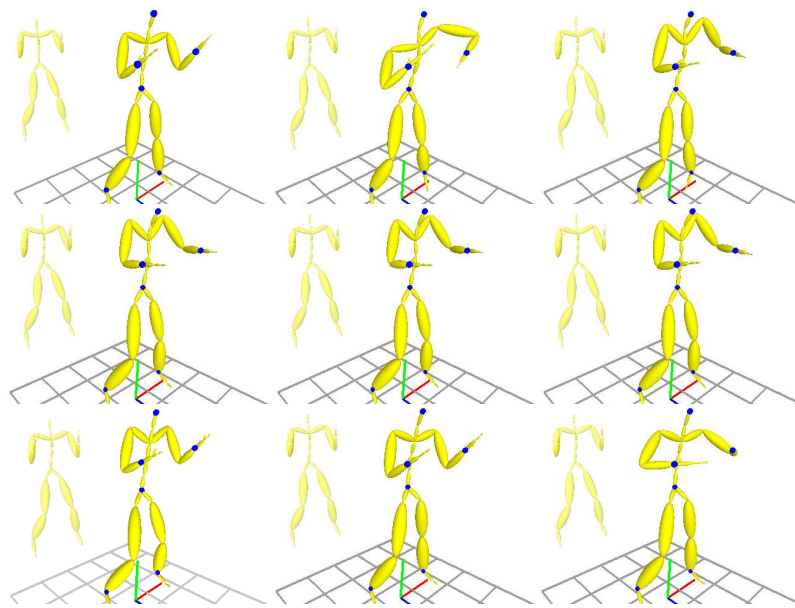


Figure 54: From left to right and top to bottom: KMA, CCD, Pseudoinverse, DLS, DLS with SVD, SDLS, PIK, SIK and TGBSIK reconstructions in frame 2,507 of the *Boxing* animation.

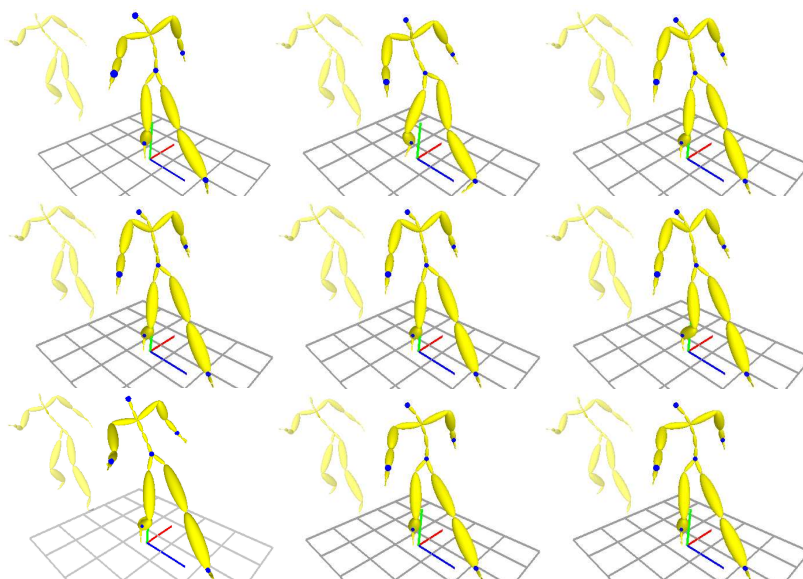


Figure 55: From left to right and top to bottom: KMA, CCD, Pseudoinverse, DLS, DLS with SVD, SDLS, PIK, SIK and TGBSIK reconstructions in frame 327 of the *Jump Kick* animation.

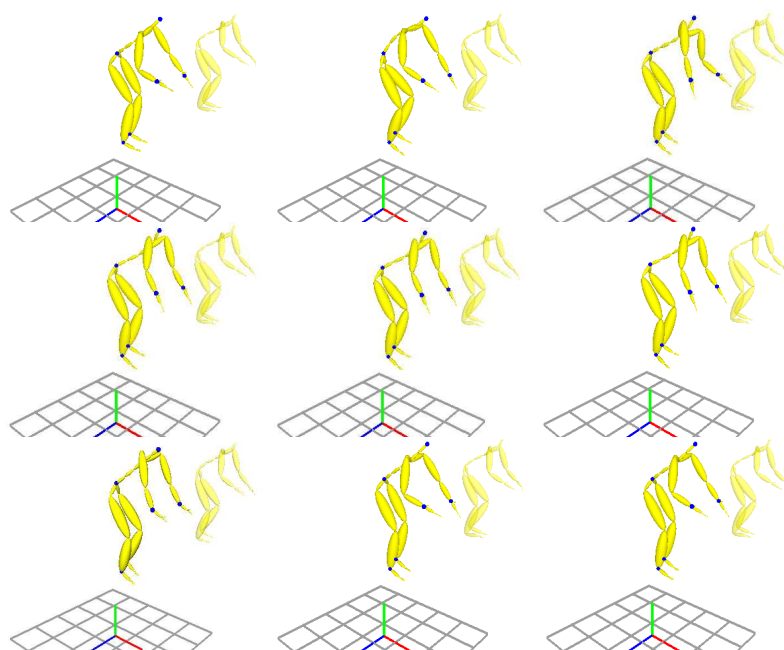


Figure 56: From left to right and top to bottom: KMA, CCD, Pseudoinverse, DLS, DLS with SVD, SDLS, PIK, SIK and TGSIK reconstructions in frame 675 of the *Playground* animation.

4.7.2 SEQUENTIAL INVERSE KINEMATICS APPLIED ON A MARKER-BASED MOCAP SYSTEM

In this section it is described how to use the SIK method, having as input data the 3D positions of a set of markers obtained with the IMPULSE optical mocap system (Phasespace 2005), which allow to estimate the orientations of end-effectors and the positions of elbows and knees. This experiment is part of the results published in (ENACTIVE 2007).

In this test the user wears a total of 24 markers distributed along the body as shown in Figure 57. Five markers correspond to the back, three to the head, three to each arm, two to each leg and three to each foot. As long as IMPULSE uses active markers they are directly labeled without the need of further marker matching strategies. Therefore, the only handicap that can arise during capture is when markers are occluded. In case of occlusion the last known position of the marker is considered for this experiment.

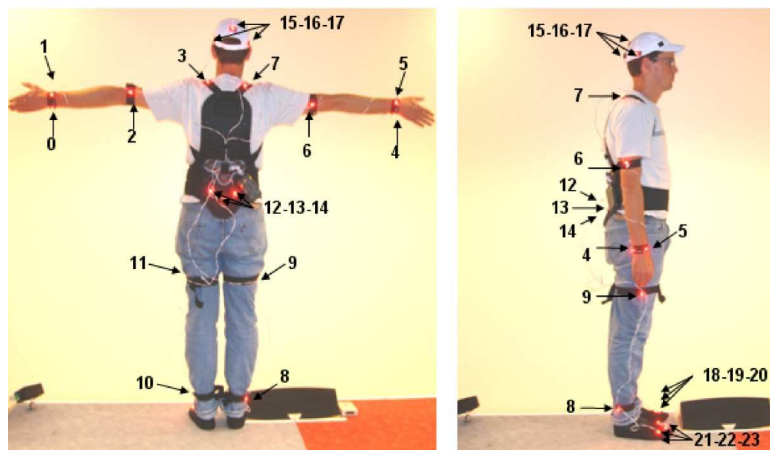


Figure 57: The IMPULSE (Phasespace 2005) mocap system's marker configuration used for the experiment (Peinado et al. 2007).

The humanoid that represents the user is shown in Figure 58. It has 86 joints and its kinematical model is specified following the H-Anim standard. For the full-body reconstruction the relative movements of the fingers, face, acromioclavicular and feet joints are not considered. The relative movements of the rest of the joints are.

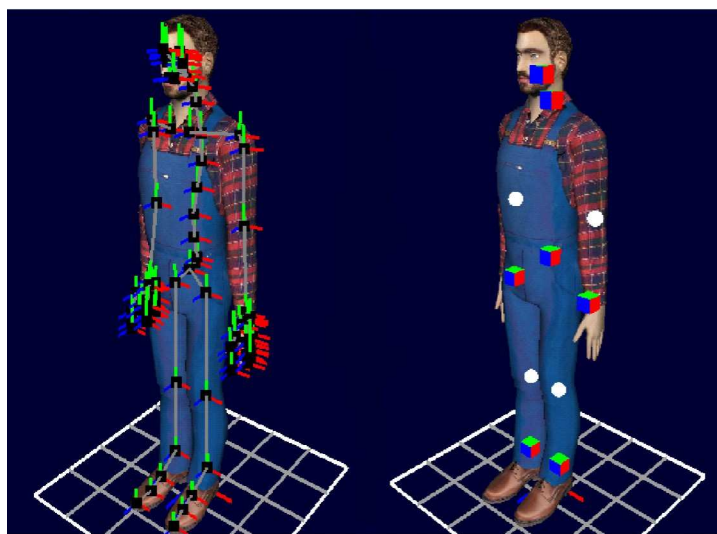


Figure 58: On the left the virtual humanoid kinematical model and on the right input features for SIK [©(Vrlab 2008)].

In the current experiment more data than those of the markerless mocap system proposed in this thesis project can be obtained from the considered marker set. The current development status of the SIK algorithm can afford the following input data (right image of Figure 58):

- Pelvis position and orientation.
- Neck (concretely *vc7* joint) axial orientation.
- Head position and orientation.
- Hand positions and orientations.
- Feet positions and orientations.
- Elbow and knee positions but not in a strict way. They are used only to determine the swivel angles of the limbs.

The marker positions must be adapted to the known data affordable by the SIK algorithm. Table 11 shows this correspondence. Apart from these feature-markers relations, for the current experiment the bending plane of the spine is defined by the back's markers (3-7-12-13-14) instead of the way explained in section 4.2.

SIK Controllable Features	Involved Markers
Pelvis position and orientation	12-13-14
Neck axial orientation	3-7
Head position and orientation	15-16-17
Left arm swivel angle	2
Left hand position and orientation	0-1
Right arm swivel angle	6
Right hand position and orientation	4-5
Left leg swivel angle	11
Left foot position and orientation	10-18-19-20
Right leg swivel angle	9
Right foot position and orientation	8-21-22-23

Table 11: Relation between SIK controllable features and markers.

At least three markers are needed to define the position and orientation of a limb. In the considered marker set this is the case of the pelvis, head and feet, but not for the hands. In the case of the hands only their positions and

their axial orientations can be obtained from the markers as long as there are only two markers.

For the human calibration it is considered that both the user and the humanoid have the same anthropometry. In order to calibrate the feature-markers matching, the user stands in the neutral pose defined in the H-Anim standard as shown in Figure 59.

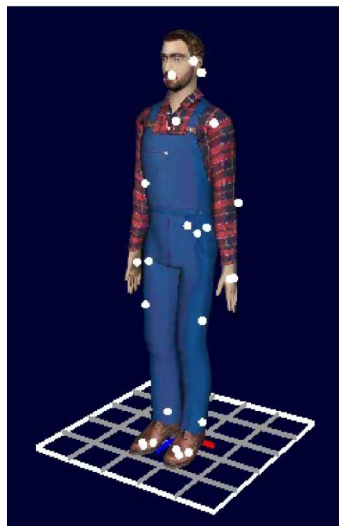


Figure 59: The pose used for subject calibration.

Up to 11 different full-body movements have been recorded using the IMPULSE mocap system (Figures 60-70). These figures show that, in general, the obtained full-body poses are visually satisfactory enough. There are cases in which non-natural poses of the spine are obtained (frame 5 of Figure 64, frames 3 and 4 of Figure 65). These occur mainly because of the rough approximation of the user-humanoid calibration and also because of the occlusions in some of the back's markers, which make the bending plane change abruptly towards non-natural orientations. There are also some odd poses for the forearms in the case that arms are almost stretched (frame 2 of Figure 63). These happen again because of the rough approximation of the user's calibration. Finally, it must be stated that biomechanical limits can lead also to odd results because they might be too restrictive comparing to those of the real user (left foot of frame 5 of Figure 62).

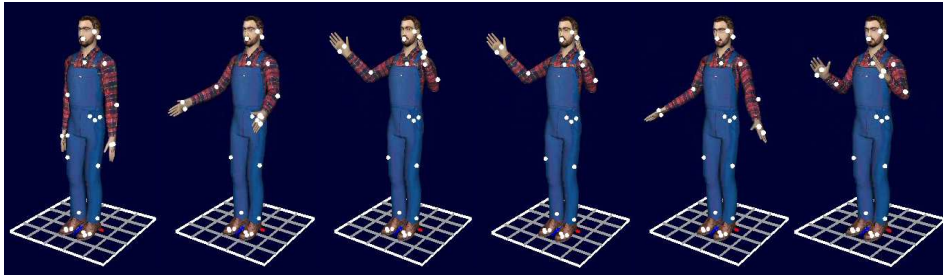


Figure 60: Reconstructed poses for *Rising Arms* animation.

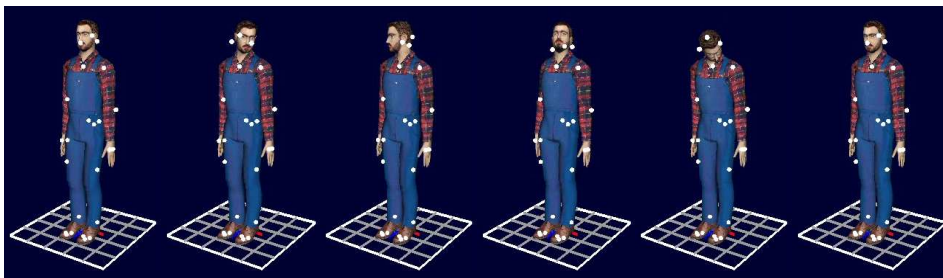


Figure 61: Reconstructed poses for *Head Movements* animation.

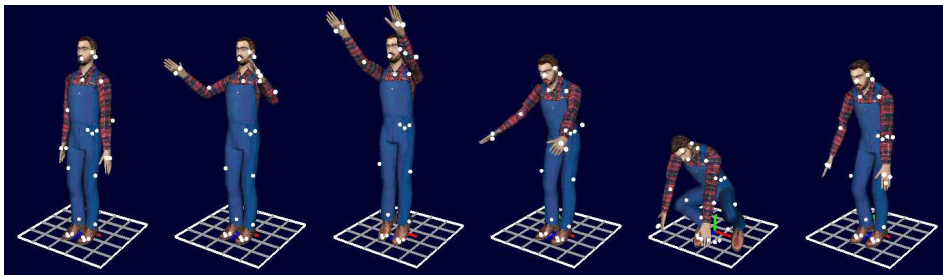


Figure 62: Reconstructed poses for *Stretching and Crouching* animation.

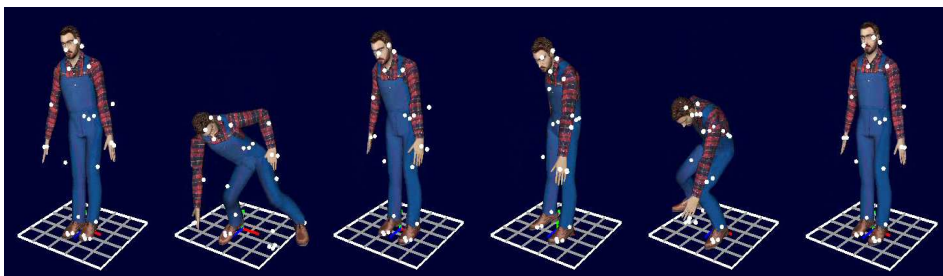


Figure 63: Reconstructed poses for *Picking Objects* animation.

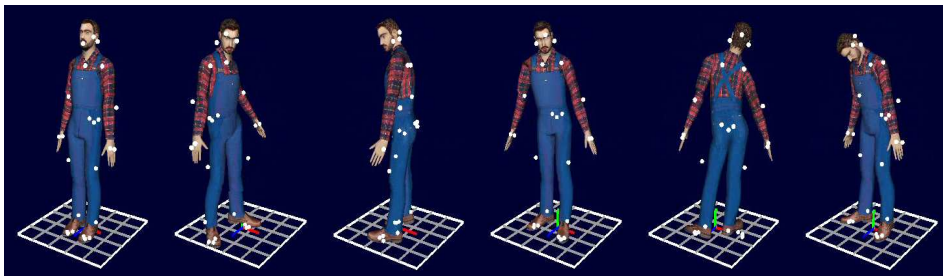


Figure 64: Reconstructed poses for *Spinning Around* animation.

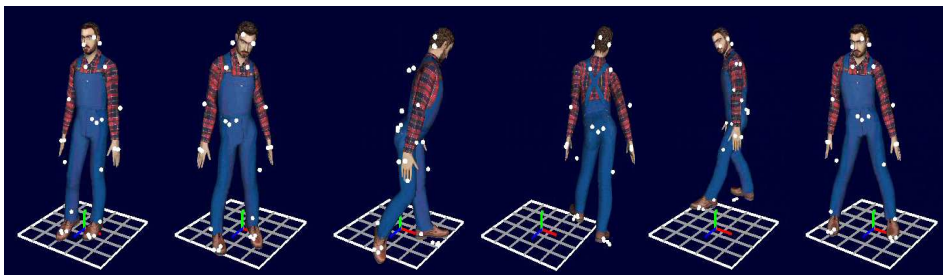


Figure 65: Reconstructed poses for *Walking Around* animation.

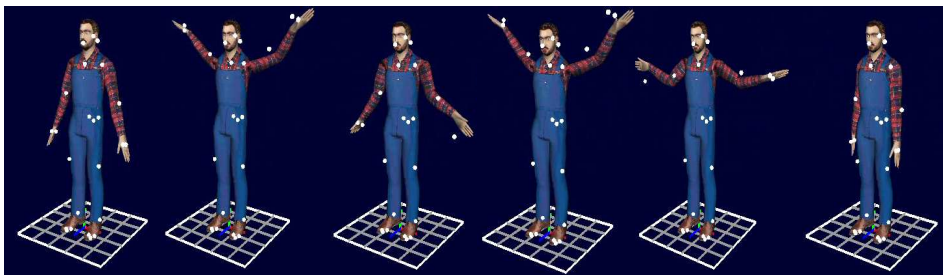


Figure 66: Reconstructed poses for *Flapping* animation.

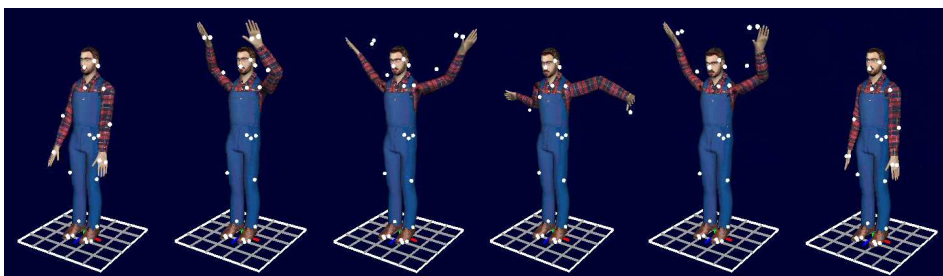


Figure 67: Reconstructed poses for *Rotating Arms Backwards* animation.

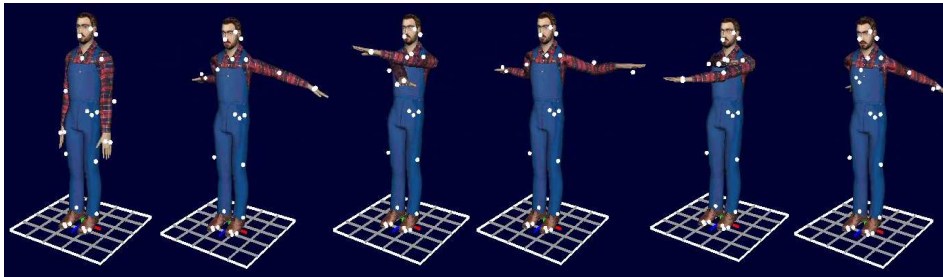


Figure 68: Reconstructed poses for *Crossing Arms* animation.

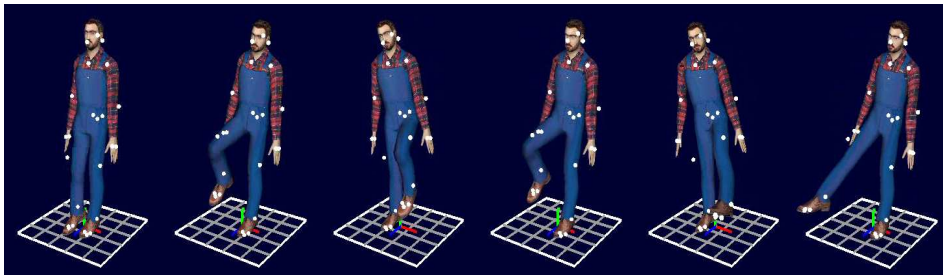


Figure 69: Reconstructed poses for *Bending and Stretching Legs* animation.

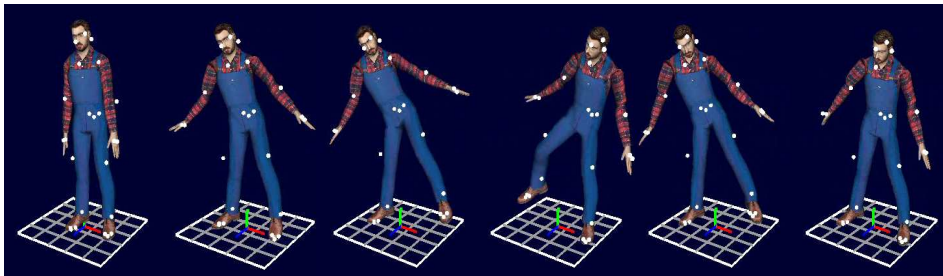


Figure 70: Reconstructed poses for *Rolling* animation.

The obtained full-body poses in this experiment are, in general, visually satisfactory enough, but there are some cases in which odd poses of the spine and the arms are obtained. These can be enhanced by making a more accurate user-humanoid calibration and by designing more sophisticated strategies to handle marker occlusions. On the other hand, it has also been shown that the biomechanical rotation limits can lead to important differences between the real and the reconstructed poses if they do not fit with those of the real person.

CHAPTER 5

HUMAN MOTOR ACTION RECOGNITION

One of the biggest issues that a motor action recognition system has to overcome is to define a representation of motor actions, i.e., to establish the features that allow the classification of movements. As shown in section 2.5, these features are data extracted from the information provided by the mocap system and they are directly dependent on the way it works. If a movement is captured using image analysis, the features can be extracted directly from changes that appear in video frames through time. On the other hand, if body poses are estimated from observations, features can be extracted from the positions, angles, angle velocities, etc., of the reconstructed joints, which allows more complex motor action descriptions.

Liu et al. (2006) demonstrated that data that comes from most mocap systems exhibit considerable redundancy and presented an approach to automatically obtain the essential information of movements in a marker-based optical mocap system. In their work they reconstruct full-body postures using as input data only the positions of some markers, the *principal markers*, situated in the areas around the hands, feet, head and the middle of the torso, by searching full-body poses from a database using the Random Forest classifier (Breiman 2001). This way they can obtain postures of which the principal markers are the nearest possible to the current values, and from these they can obtain a posture adjusted to the features. This approach can be a good basis for defining proper pose models that allow a quick and discriminative search in databases for an adequate motor action semantic recognition.

Another important issue to overcome for motor action recognition in HCI applications is to segment the continuous data flow being captured into meaningful and non-meaningful gestures, which is called *gesture spotting*. At this point, we can distinguish between two types of gestures or motor actions for their semantic description while data are being captured: (1) *static* or *quasi-static* gestures and (2) *dynamic* gestures. The former obtain their meaning only from the spatial configuration of the tracked data, i.e., only the instantaneous values are important for their semantic description. Humans are not perfect static posture performers and that is the reason for employing the word “quasi-static” in order to refer to those gestures of actions where the pose is meaningful. On the other hand, dynamic gestures rely on both the spatial and also the temporal information, i.e., the sequence of poses. The challenging problem is to determine their starting and ending instants while capturing in real time.

Time warping, which means that the performed actions can have accelerations, decelerations and durations with respect to the prototype of the knowledge database, also constitutes another issue to develop a correct action classification. Finally, discerning actions being performed at the same time like, e.g., “saying hello” while “walking” or while “being still”, is another challenging issue, especially when full-body actions are expected to be semantically described. Thus, our purpose in the present chapter is to overcome these problems in order to recognize gestures aimed to HCI applications.

This chapter is organized as follows: firstly, a method for gesture spotting is presented. It relies on the temporal evolution of the tracked measurements of a body part of interest expressed with a single non-negative scalar value called kinetic pseudo-energy. Secondly, how the spotted quasi-static and dynamic gestures are classified, for their labeling with respect to the database of known possible semantic descriptions, is explained. Then, a full-body pose model aimed for an effective search in full-body pose databases, for both depth warping of the tracked end-effectors using a single standard camera, and for combined actions recognition, is presented. The next section explains how the mentioned depth warping can be achieved in order to improve the visual appearance of the pseudo 3D full-body reconstruction with a single standard camera. Then, how combined actions can be detected from full-body movements is explained. Finally, experimental results of the presented methods are shown, combined with the motion capture and pose reconstruction methods presented in chapters 3 and 4, in order to evaluate their overall suitability for HCI applications.

5.1 GESTURE SPOTTING WITH KINETIC PSEUDO-ENERGY HISTORY

The determination of the starting and the ending instants of dynamic gestures is a challenging task due to the fact that non-relevant movements can be performed while the user's movements are being captured. In our approach we make use of the data flow's most recent kinetic status to determine when a potentially meaningful dynamic gesture happens. This strategy, combined with the gesture recognition method presented in the next section (5.2), is able to discern meaningful static or quasi-static and dynamic gestures from those which are not for HCI applications.

In general terms, the kinetic status of a point-like solid of mass m moving with a vectorial velocity $\dot{\mathbf{x}}$ can be depicted with a single value by its kinetic energy (Equation 11).

$$E_k = \frac{1}{2} m |\dot{\mathbf{x}}|^2 \quad (11)$$

This energy provides a non-negative scalar quantity that reflects the motion of the solid, which can correspond in our context, e.g., to the positions of the tracked joints. This energy may be used for determining the starting and ending instants of a potential dynamic gesture where the transitions may be considered as potential quasi-static gestures. Potential, in this context, means that they should be checked by the recognition procedure presented in section 5.2 in order to determine if they really are movements or poses with an expected semantic description. For instance, periods of time where sudden kinetic energy variations occur may correspond to dynamic gestures, while "landing" periods may mean that the user's intention is to perform a meaningful static or quasi-static gesture. Thus, the key factor of the kinetic energy for motor action recognition tasks is the non-negative scalar value of the velocity $\dot{\mathbf{x}}$, as the mass will keep constant during the capture.

However, the data flow may not only contain the trajectories of joints, which can make direct use of Equation 11, or more specifically the non-negative scalar value of the velocity, for gesture spotting tasks, it may also include angular information, accelerations, etc. Nevertheless, the non-negative scalar value of their temporal variation can be used in the same way for gesture spotting tasks, as they also reflect the kinetic status variation. We call this data flow temporal variation expressed as a non-negative scalar value: *kinetic pseudo-energy* psE_k . We apply it to check the kinetic evolution of those body parts that we consider important for the HCI application.

A simple way of detecting when a dynamic gesture may happen can be to threshold psE_k . But it may occur, during the performance of a gesture, that the body part changes its direction from one side to another which could register as $psE_k = 0$. This situation arises, e.g. in “hand waving” or “shaking” gestures. Therefore, this approach is not sufficient to detect gestures involving direction changes. On the other hand, we can consider the n most recent frames of a body part’s kinetic pseudo-energy at frame t , or in other words, the kinetic pseudo-energy *history* (Equation 12).

$$psE_k^{history} = [psE_k^{(t-(n-1))}, \dots, psE_k^{(t)}, \dots, psE_k^{(t)}] \quad (12)$$

We propose to threshold the mean value of $psE_k^{history}$ in order to handle direction changes. This magnitude evolves with a higher inertia than the instantaneous psE_k and therefore it is more robust to sudden changes. On the other hand, it moves with a certain delay with respect to the real movements, which must be determined in order to get the starting and ending instants of gestures more adjusted to the real ones. This delay can be estimated with Algorithm 14, which is applied to correct both the starting and ending instants with the same value.

Algorithm 14 Mean Kinetic Pseudo-Energy History Delay Determination

```

1: procedure DELAYDETERMINATIONALGORITHM( $psE_k^{history}$ ,  $meanPSE_k^{threshold}$ )
2:    $startingPoint \leftarrow t$ 
3:   for  $i = t - (n - 1)$  to  $t$  do
4:     if  $psE_k^{history}(i) < meanPSE_k^{threshold}$  then
5:        $startingPoint \leftarrow i$ 
6:     end if
7:   end for
8:    $delay \leftarrow t - startingPoint$ 
9:   return  $delay$ 
10: end procedure

```

Figure 71 shows how the application of the kinetic pseudo-energy history approach detects satisfactorily the execution of a dynamic gesture ($Threshold = 1.2$). How the delay is corrected, and how the spotted segment (grey area) corresponds to a region of the data flow with sudden kinetic energy variations, i.e. a potential dynamic gesture, can also be appreciated. Additionally, the instantaneous kinetic status in this region falls on two occasions below the threshold but, nevertheless, the segment includes them.

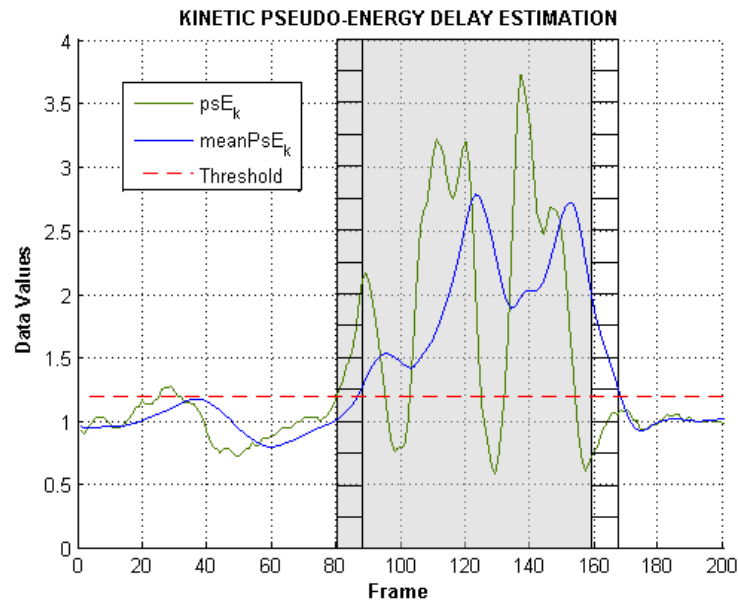


Figure 71: Mean kinetic pseudo-energy history delay correction for gesture spotting.

5.2 QUASI-STATIC AND DYNAMIC GESTURE RECOGNITION

Once the potential gestures are segmented from the continuous data flow with the kinetic pseudo-energy history, it is necessary to determine whether they are gestures of the database or not. The tracked m features $[f_1, f_2, \dots, f_j, \dots, f_m]$ of the body part can represent its position coordinates and/or orientation angles, their accelerations, etc., and evolve in a continuous stream through time. At frame t the feature values represent a pose that can be expressed as a vector of m elements: $pose^{(t)} = [f_1^{(t)}, f_2^{(t)}, \dots, f_j^{(t)}, \dots, f_m^{(t)}]$. Thus, static or quasi-static gestures can be easily classified using the K-NN procedure (Dasarathy 1991) between the current pose, $pose^{(t)}$, and those of the database. Ibarguren et al. (2007) showed the robustness of this procedure to noisy measures with respect to other statistical classifiers. Another important feature of K-NN is that the distance allows both spotting and determining the correctness of a pose. If the current pose is too distant from its nearest neighbors it may be considered as “unknown”, and therefore, not spotted. Meanwhile, a movement, which is a set of poses ordered in time, can also be expressed as the vector of $m \times n$ elements, where m is the number of features and n the number of recent frames, as shown in Equation 13.

$$mov = \left[f_1^{(t-(n-1))}, \dots, f_m^{(t-(n-1))}, \dots, f_1^{(t-(n-2))}, \dots, f_m^{(t-(n-2))}, \dots, \right. \\ \left. f_1^{(t-1)}, \dots, f_m^{(t-1)}, \dots, f_1^{(t)}, \dots, f_m^{(t)} \right] \quad (13)$$

In order to recognize dynamic gestures it would also be interesting to measure the distance between the performed gesture and those of the database, but the frame number may vary both from gesture to gesture and from performance to performance, i.e., time warping occurs. Therefore, to be able to recognize dynamic actions using K-NN we propose to resample the performed actions to the same number of frames with a cubic spline. The first derivatives of both the starting and ending points should be the same as in the original which can easily be calculated with Equation 14 (bearing in mind that t is the current frame).

$$\frac{\partial f_j^{(t-(n-1))}}{\partial t} = f_j^{(t-(n-2))} - f_j^{(t-(n-1))}, \text{ and } \frac{\partial f_j^{(t)}}{\partial t} = f_j^{(t)} - f_j^{(t-1)} \quad (14)$$

Additionally, in order to cope with scaling and offset differences in gestures, which comes from anthropometric variations and the manner in which capture is carried out, these must be translated and rescaled to controlled regions. For instance, if both positive and negative data are captured, they can be translated and rescaled to values between 0 and a user defined $N \in \mathfrak{R}$ positive value. The conjunction of the resampling, translating and rescaling processes is called *normalization* of the gesture. This way we can have a database of different dynamic gesture performances expressed as single vectors, making the recognition procedure more robust to time warping and anthropometric differences. This allows measuring the distance and, therefore, evaluating the correctness of a new performance. This distance determines the disparity in the posture evolution along time. Again, too distant performances can be labeled as “unknown”, which can be used to spot meaningful gestures from those which are not. Moreover, we can analyze which features, and at which time instants, are further from their reference performance. This can be helpful in applications for skills acquisition and transferring.

Figure 72 shows the general procedure for the recognition of dynamic gestures. Each circle represents a body part state, i.e., its instantaneous configuration. These states are ordered in time and therefore transitions between them are well-defined in the dynamic gestures of the database. Quasi-static gestures are treated in a similar way for their recognition, but neither applying the normalization step nor taking into account their temporal transitions as they are considered to be independent from each other.

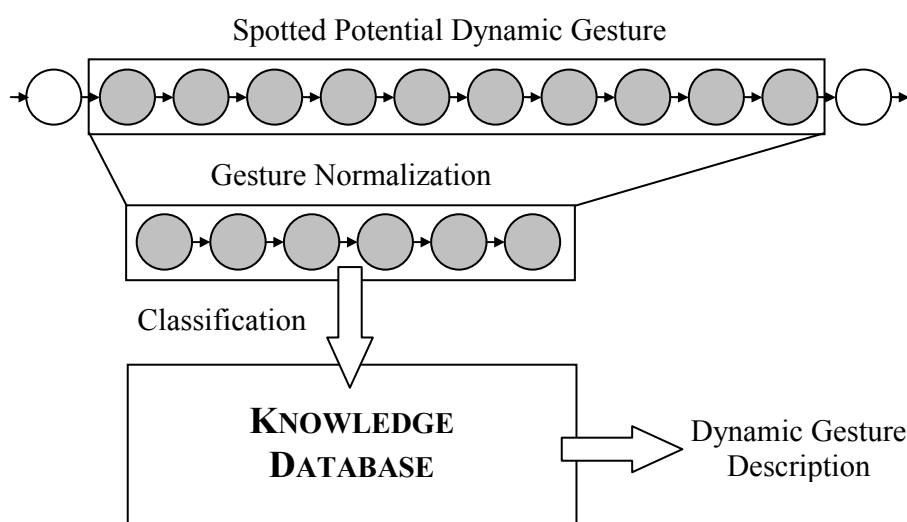


Figure 72: The dynamic gesture recognition procedure.

The K value of the K -NN classifier represents the number of nearest neighbors in order to determine the label of the new incoming data. This gets assigned an odd number, usually from 3 to 7, and the classification is attained by voting, which requires more than one sample for each class. In case there is only one sample available for each class, the distance to it determines the classification. Both voting and the distances can also be combined. It depends on the purpose of the HCI application, e.g., it may be interesting to use $K=1$ for evaluating the imitation of a specific database gesture, or it may be interesting to determine only the class and not the mimesis, etc.

5.3 FULL-BODY POSE MODEL FOR DATABASE SEARCH

Typically, in computer graphics, a human full-body pose is determined with the relative rotation angles of the joints and the position of the root joint of the multi-body mechanism that defines the subject. Thus, a high number of degrees of freedom (DoF) is used, in the order of 40, but it may occur that most of them do not really contribute significantly in the performance of a meaningful movement. This is especially striking in a case where combined actions are being performed, e.g., the rotation angles of legs must not have any influence in the recognition of a motor action that involves only arms, like in a “waving” gesture.

On the other hand, we propose to define full-body poses for motor action recognition with only 10 DoF in the case of a pseudo 3D motion capture (with a single standard camera), and 15 DoF in the case of a full 3D

motion capture (with a depth sensing camera or multiple camera systems): the positions of hands, feet and head relative to a coordinate system attached to the pelvis. This way we extend the philosophy of the work of Liu et al. (2006) for motor action recognition instead of motion reconstruction as they do. The advantages of this representation are multiple:

- The essential information about movement is retained in a significantly reduced form without the need of applying statistical techniques such as PCA (Fukunaga 1990).
- Actions can be represented in the same way independently of the position and orientation of the person with respect to the absolute coordinate system.
- These data are easy to track with most mocap systems, even by computer vision based markerless strategies.
- Unlike rotations, positions are Euclidean, so true distances between positions are much easier to calculate.
- Minor differences in rotations may lead to big differences in posture unlike the cases of minor differences in positions.
- The use of end-effector positions instead of joint angles opens up the possibility of establishing direct relations with surrounding objects to enrich the interaction.

It must be remarked that these positions correspond to the end-effectors of the humanoid and not those of the user directly, i.e., it is necessary to reconstruct the pose. It is shown in the experimental results (section 5.6.2) that reconstruction allows easy rescaling of the same knowledge database to different anthropometries, as many poses maintain their semantic descriptions if the rotation angles are the same in many actions involving only the body. It additionally constrains the spatial search space of meaningful poses and allows users to control humanoids with different anthropometries as if they were puppets.

Figure 73 shows the measurements for the pose database search in a pseudo 3D motion capture using a single standard camera: the 2D vector $\mathbf{v}_{\text{torso}}$, which goes from the projected pelvis to the projected head, and the 2D vectors corresponding to the limbs relative to $\mathbf{v}_{\text{torso}}$ expressed in polar coordinates (d_{Hand} , a_{Hand} , d_{rHand} , a_{rHand} , d_{Foot} , a_{Foot} , d_{rFoot} and a_{rFoot}). These are usable for both depth warping of tracked body parts and combined action recognition as shown in the following sections (5.4 and 5.5). The use of polar coordinates assumes that the database limb pose search is more sensitive to distance

variation than angle variation, since the Euclidean metric is applied for that search and distances can get higher values than angles. As it is shown in section 5.6.2 this is sufficient for our context. Otherwise, Cartesian coordinates with respect to the torso's coordinates system will be used.

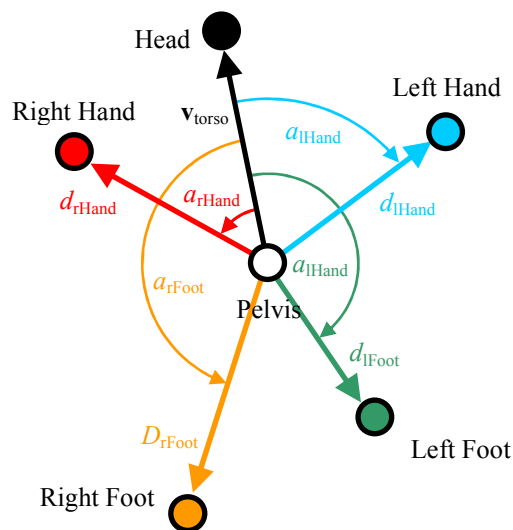


Figure 73: Vision measurements for depth warping and combined action recognition using a single standard camera for motion capture.

5.4 DEPTH WARPING IN A SINGLE-VIEW TRACKING

As explained in chapter 3, once hands, head, feet and pelvis are tracked in 2D, they can be unprojected to the virtual character that represents the user. This can be achieved if: (1) the camera is calibrated, (2) the humanoid overlaps the user in the image and (3) the depth of the corresponding 3D body parts are supposed to be constant. This way, visually apparent 3D reconstructed poses can be obtained with the method proposed in chapter 4 by observing the humanoid from the same point of view as the camera, but not from other points of view.

The 3D pose reconstruction can be improved measuring the depth variation of the tracked body parts. This is achievable with more than one view or using a depth-sensing camera, but it is not possible with only one standard camera. However, as it is intended to track the user in order to make the system interpret the performed actions for HCI, it is possible to make use of previously known poses to allow depth variation.

Consider a database of known body poses. In a calibrated camera it is possible to measure the depths corresponding to hands, head, feet and pelvis in those configurations and relate them to the vision measurements (Figure 73). Note that these measurements are independent from the place of the image in which the humanoid is located, i.e., their value is relative to the coordinate system attached to the pelvis, and therefore will have specific values that will maintain the limb pose even if the trunk has been translated and rotated.

Thus, a set of 2D points corresponding to the projections of hands, head, and feet relative to the torso position and orientation with known and different depths, which transform them into 3D points in virtual 3D space, can be obtained. The projected 2D points can be triangulated by applying Delaunay's (Delaunay 1934) triangulation method (upper row of Figure 74). This way, the screen plane is subdivided into a set of non-overlapped regions (facets) that cover the whole plane. During the triangulation process, a set of virtual points is created out of the screen borders connected by edges with the exterior subdivision points (namely, convex hull points) of the surface. As these virtual points have not been obtained from the database, their depth is not available. A reasonable depth value for them may be the mean depth of the convex hull points to which they are directly connected through edges.

Therefore, as all the facet vertex depths are now known, this 2D triangulation can lead to a warped surface in 3D space by unprojection. Hence, the depth corresponding to a certain tracked body part projection can be estimated in any position of the 2D image by, firstly detecting which 2D triangle contains the projection of the end-effector, and secondly calculating its depth from those of the triangle vertices, considering it lays on the triangle plane in 3D space. In this way, a continuous depth variation of each end-effector can be attained enhancing the visual appearance of the pseudo 3D motion capture. This procedure must be done separately for each of the tracked end-effectors as their separate motion should not affect the depth variation of the rest. The Delaunay triangulation is only applied when the system is initialized and their triangular relations are maintained constant during the capture. Then, the tracked point positions are transformed into the triangulated surface coordinate system according to the torso position and orientation, as they will change because of the user's movements.

The procedure to estimate the depth of a 2D projection on a view having a warped surface generated from a database of poses is shown in Algorithm 15. It must be stated that OpenCV (Intel 2001) has tools to attain the Delaunay 2D triangulation procedure as well as steps 3, 4 and 12 of this algorithm. Here, the triangulated 2D plane is expressed as *triang2DPlane*.

Algorithm 15 Depth Estimation of a Projected End-Effector

```

1:  procedure DEPTHESTIMATION(p2D, triang2DPlane)
2:      Set p2D with respect to the triang2DPlane coordinate system (this
        is done due to the fact that the trunk may be translated and
        rotated)
3:      Locate the e0 edge of triang2DPlane p2D falls onto or to the right
        of
4:      Locate the e edge next around the left facet of e0
5:      while e  $\neq$  e0 do
6:          Get the origin vertex org of e
7:          for every vertex of triang2DPlane do
8:              if current triang2DPlane vertex = org then
9:                  Store org and its corresponding depth
10:             end if
11:          end for
12:          Locate the e edge next around the left facet of e0
13:      end while
14:      if number of stored vertices = 1 then
15:          return vertex depth
16:      else
17:          Calculate the depth by linear interpolation considering that
            p2D lays on a triangular facet formed by the stored vertices
18:      end if
19:  end procedure

```

Figure 74 shows examples of depth warping from the triangulated surface of the left hand. It can be appreciated in the lower row that the hand position in the direction perpendicular to that of the image (upper row) changes depending on its projected 2D position. Using this triangulated warped surface it is possible to vary smoothly the depth of the tracked end-effector, even during the transitions from facet to facet, while the end-effector transits through them. Limb pose databases or viewpoints in which two or more samples with similar projected coordinates have big depth differences should be avoided. Hence, the obtained 3D poses enhance those in which the depth is maintained constant if the humanoid is observed from other points of view different from that of the camera.

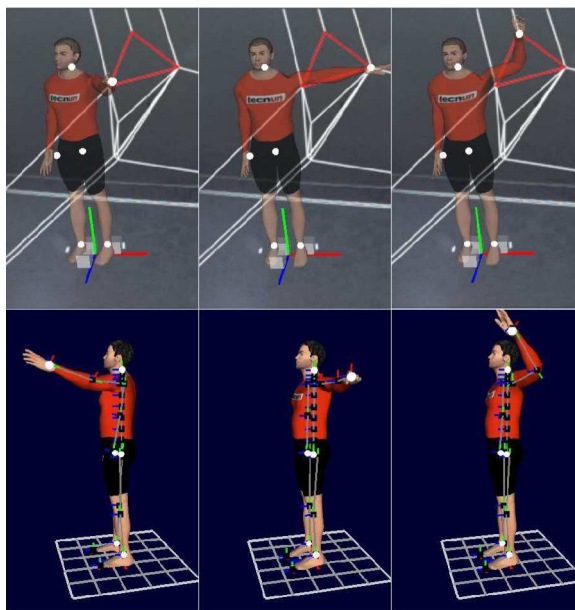


Figure 74: The left hand position depth warping on the camera view. The three projections are inside the red triangle, which is used for the depth calculation.

5.5 HUMAN FULL-BODY COMBINED ACTIONS

As was the case for the depth warping of the tracked end-effectors, by using the full-body pose model for the database search proposed in section 5.3 separately for each limb, it is possible to attain the recognition of combined actions.

Consider the same pose database used for the generation of the triangulated warped surfaces for end-effector depth warping. We can describe each of the limb body configurations in a low level semantic way. Here, “low level” means that we describe limbs (arms, legs and trunk) in a general form considering only their spatial relative poses. For instance, arms and legs can be described as “straight” or “flexed” towards a certain direction with respect to the trunk, while the trunk configuration can be described with respect to the surrounding environment. These basic descriptive “bricks” can be employed to build a higher level description of the full-body pose. This way, a “standing” pose would be built from the “straight-down” limb descriptions of arms and legs and “straight-up” description of the trunk. Thus, due to the independence in which the body configurations are stored in the knowledge database using the full-body pose model of section 5.3, it is possible to combine these limb

descriptions in order to generate higher level descriptions only with the limbs involved.

Hence, it is possible to detect “waving” actions involving only one arm, for example, while ignoring the configuration of the rest of the limbs, improving its recognition rate and performance as the size and the lower complexity of the database samples are more adequate for the label search. Besides, the semantic description of the full-body is enriched since it is possible to discern whether those limb descriptions correspond to the left or right side of the body.

These spatial limb descriptions are applicable for combined quasi-static full-body gesture recognition. In the case of combined dynamic gestures, it is necessary to determine if spotted gestures for each limb occur in a combined way or if they are independent from each other. For this reason, while a dynamic gesture is being spotted in one limb it must be checked whether any of the remaining limbs have also “entered” into a potentially meaningful dynamic gesture “region”. In that case, the system, before sending any response message, must wait until the separate dynamic gesture labels are available, and if their combination has a higher level meaning, release it, otherwise it must send separate messages.

5.6 EXPERIMENTAL RESULTS

5.6.1 RESULTS ON DYNAMIC GESTURE SPOTTING AND RECOGNITION

In order to evaluate the suitability of the approaches for dynamic gesture spotting and recognition, the test explained in this section has been carried out. The implementations of the statistical classifiers used on it have been obtained from the machine learning tools of the OpenCV (Intel 2001) library for C++. The calculations were performed by a 2.66-GHz Pentium 4 with 512 MB RAM with a FireWire camera capturing with 30 FPS images at a resolution of 320×240 .

Two subjects perform a set of dynamic gestures, separately, with their right hand tracked using the CFOF approach presented in chapter 3. These dynamic gestures correspond to the numbers from 0 to 9 traced as shown in Figure 75. Both subjects perform the gestures at a similar position and orientation with respect to the camera (the camera is around 3.5 meters in front of the user) having similar full-body poses (standing up). Apart from the obvious variability in gestures which stems from the non-perfection of human mimesis capabilities, the remarkable differences in their anthropometries (subject 1 is 1.78 m tall, while subject 2 is 1.62 m) must also be taken into

account. Due to these differences, hands are visualized in different positions of the camera and performed gestures tend to have different sizes from subject to subject.

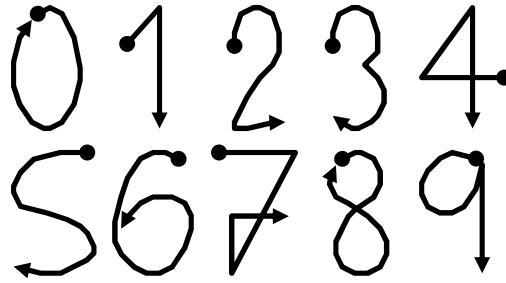


Figure 75: The set of dynamic gestures for the experiment. The dot represents the starting point while the arrow represents the direction and the end point.

Each subject performs five samples of each number recorded continuously. Therefore, 10 continuous data flows containing five repetitions of the gestures for each subject, i.e., 100 samples to be spotted and recognized, are available for evaluation. The objective of the test is to, firstly, spot potentially meaningful gestures from the continuous data flows and then use them for number recognition in a supervised manner. Table 12 shows the gesture spotting and normalization parameter values used for the test.

Kinetic Pseudo-Energy Parameters	Number of History Frames	10
	Threshold	2.5
Gesture Normalization Parameters	Maximum DoF Length	50
	Number of States	30

Table 12: Considered parameters for gesture spotting and normalization.

The spatial forms of the spotted potentially meaningful gestures are shown in Table 13. It can be observed that they indeed correspond to the numbers which are intended to be performed by both subjects. The differences in size and positions can also be appreciated as well as how these are diminished after normalization for subsequent gesture recognition. On the other hand, Figure 76 and Figure 77 show the segmentations of the “number 7” gestures from the continuous data flows of both users, where $p_s E_k$ is the instantaneous kinetic pseudo energy and $mean P_s E_k$ its mean value derived from its history. It can be seen that the spatio-temporal form of gestures follows a similar pattern as well as how the transition movements are ignored with respect to the rest. The obtained segments have different durations due to the difficulty humans have for reproducing gestures in the same way. Their time variability is shown in Table 14.

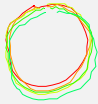






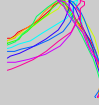











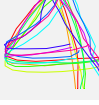








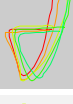


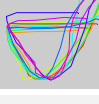








Number	Subject 1 Gestures	Subject 2 Gestures	Overlapped Gestures	Normalized Gestures
0				
1				
2				
3				
4				
5				
6				
7				
8				
9				

Table 13: Spotted gestures spatial form in the performances made by subjects 1 and 2 and their corresponding normalized shapes.

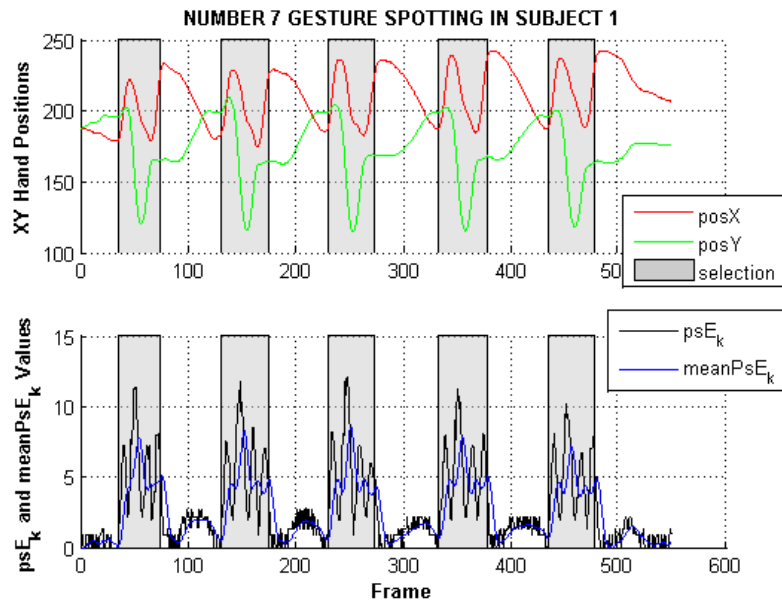


Figure 76: The continuous data flow containing 5 performances of number 7 made by subject 1 and its segmentation results.

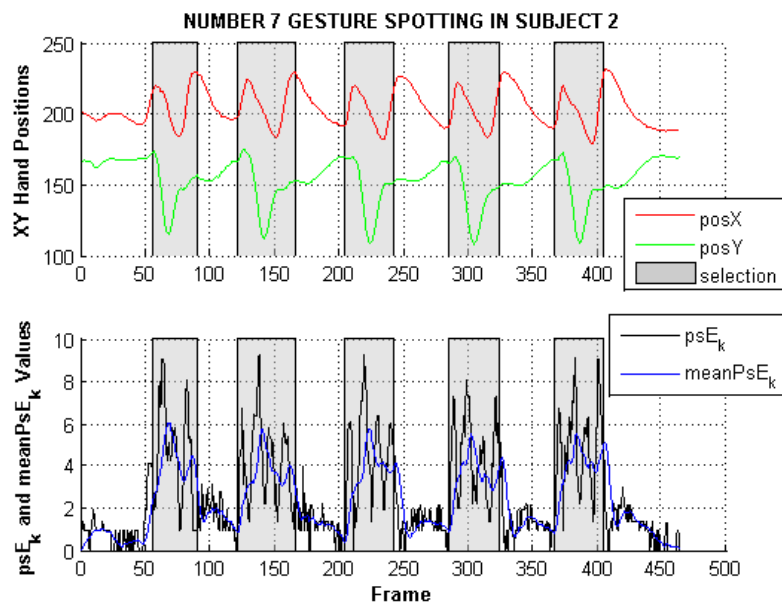


Figure 77: The continuous data flow containing 5 performances of number 7 made by subject 2 and its segmentation results.

Number	0	1	2	3	4	5	6	7	8	9
Time variation in S_1 (%)	10.51	3.67	4.80	6.37	2.86	3.88	4.95	5.33	10.11	1.32
Time variation in S_2 (%)	5.26	7.48	5.56	7.12	4.82	5.33	6.42	10.26	6.77	7.75

Table 14: Time variation in gestures performed by subjects 1 (S_1) and 2 (S_2) expressed as the standard deviation with respect to the mean in percentage.

The Leave One Out (LOO) technique (Stone 1974) has been employed to obtain the recognition ratio of the proposed approach. This technique consists on training the classifier using all samples of the database except one which is used for classification. If the sample is correctly classified a point is summed to the number of recognized samples. This procedure is applied to every sample of the database and the ratio is obtained with the relation between the total number of recognized samples and the total number of samples.

The performance of the K-NN classifier is compared with the other two: SVM (Shawe-Taylor and Cristianini 2000) and Random Forest (Breiman 2001). The parameters used for these classifiers in this test are shown in Table 15.

K-NN Parameters	K	1
	Maximum Distance	-
SVM Parameters	Type of SVM	<i>n</i> -class classification
	Kernel Type	Polynomial
	Kernel Parameters (<i>degree, gamma, coef0</i>)	(3, 0.01, 0)
	Generalized SVM Optimization Problem Parameters (<i>C</i>)	10
Random Forest Parameters	Maximum Tree Depth	20
	Minimum Number of Samples for Node	1
	Maximum Categories for Split	15
	Number of Variables for Best Split	4

Table 15: Parameters for the statistical classification comparison.

Using this database and the LOO technique, both K-NN and SVM classifiers obtain a 100% of recognition ratio. On the other hand, the Random Forest classifier obtains an 86% recognition ratio. Its confusion matrix is

shown in Table 16. It can be seen that the most confusing gestures for this classifier correspond to numbers 7 and 8, which have a high variability in space along their path, and also in time as shown in Table 14.

Assigned Number	Real Number									
	0	1	2	3	4	5	6	7	8	9
0	9	0	0	0	0	0	1	5	2	0
1	0	10	0	0	0	0	0	0	0	0
2	0	0	10	0	0	0	0	0	0	0
3	0	0	0	10	0	0	0	0	0	0
4	0	0	0	0	10	0	0	0	0	0
5	0	0	0	0	0	9	0	0	0	0
6	0	0	0	0	0	1	8	1	1	0
7	1	0	0	0	0	0	0	4	1	0
8	0	0	0	0	0	0	1	0	6	0
9	0	0	0	0	0	0	0	0	0	10

Table 16: Confusion matrix of the classification using Random Forest.

The reason for the high recognition rate, especially in the case of K-NN and SVM, are the well-defined clusters which are obtained thanks to the normalized spatio-temporal form in which gestures are represented, and also because a correct gesture spotting is attained. Figure 78 shows the database gestures represented as three-dimensional points in which dimensions correspond to the three highest principal components in PCA space. It can be seen that the most disperse clusters correspond to those gestures with a higher variability, which need more than three principal components to be distinguished with more consistency from the rest. Note that PCA is not applied for the database storage for further recognition, it is only used for the visualization of gestures in this figure.

Regarding the training stages, SVM needs around 1 second to train the database and Random Forest around 24 seconds. K-NN does not theoretically need training but its implementation needs an initialization process which takes around 25 milliseconds. On the other hand, the time needed for the classification of a new sample, including the delay correction and the normalization steps, is of around 1 millisecond for the three classifiers. The size of this database can be considered typical for HCI applications involving human gestures. Therefore, focusing on the computation times, K-NN is preferable to SVM.

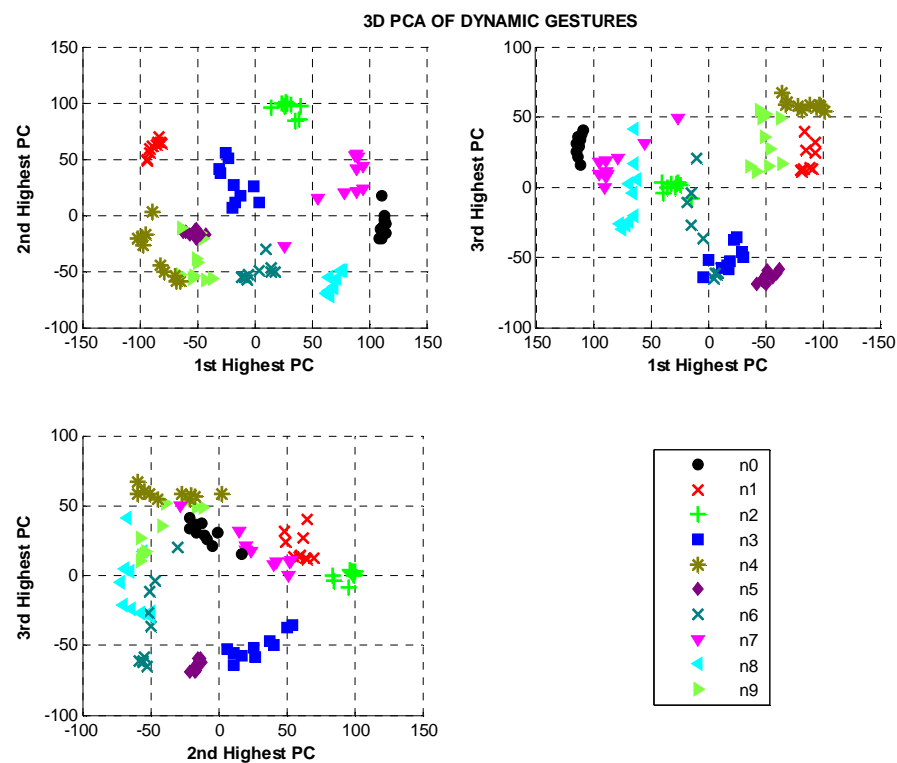


Figure 78: The three highest principal components of the gestures. The tendency to cluster can be observed. More disperse clusters need more principal components to be better defined. Note that PCA is not used for the classification.

The approaches for gesture spotting and quasi-static and dynamic gesture detection that have been presented in this study have also been satisfactorily tested with the triple axis accelerometer iMEMS ADXL330 (Analog-Devices 2007) embedded in the Nintendo Wii (2006) video game console remote, obtaining similar conclusions. These results can be found in the the work of Unzueta et al. (2008) corresponding to one of the publications generated during this thesis project.

5.6.2 RESULTS ON COMBINED MOTOR ACTION RECOGNITION

In this section a test to evaluate the combined actions recognition strategy is described. The same two subjects of the experiment described in section 5.6.1 are intended to communicate with the computer performing full-body movements, which may include quasi-static combined actions. Once again, a single FireWire camera is situated around 3.5 meters in front of the user, capturing images at 30 FPS with a resolution of 320×240 , and it is calibrated

using Zhang's approach (2000). The software is programmed in C++ with OpenCV (Intel 2001), and the computer in this case is a 2.4 GHz Intel Core 2 Duo with 2 GB of RAM. A set of quasi-static limb gestures are meant to be recognized on the test from the full-body poses captured with the real-time markerless approach presented in this thesis project (chapters 3 and 4). Tables 17 and 18 show the considered limb pose semantic descriptions.

Trunk	Right Arm	Left Arm	Right Leg	Left Leg
Straight Up (<i>tsu</i>)	Straight Down (<i>rasd</i>)	Straight Down (<i>lasd</i>)	Straight Down (<i>rlsd</i>)	Straight Down (<i>llsd</i>)
Straight Left (<i>tsl</i>)	Straight Right (<i>rasr</i>)	Straight Left (<i>lasl</i>)	Straight Right (<i>rlsr</i>)	Straight Left (<i>llsl</i>)
Straight Right (<i>tsr</i>)	Straight Forward (<i>rasf</i>)	Straight Forward (<i>lasf</i>)	Flexed Down (<i>rlfd</i>)	Flexed Down (<i>llfd</i>)
Flexed Forward (<i>tff</i>)	Flexed Down (<i>rafid</i>)	Flexed Down (<i>lafid</i>)	-	-
-	Flexed Up (<i>rafu</i>)	Flexed Up (<i>lafu</i>)	-	-

Table 17: Limb pose labels for combined actions building.

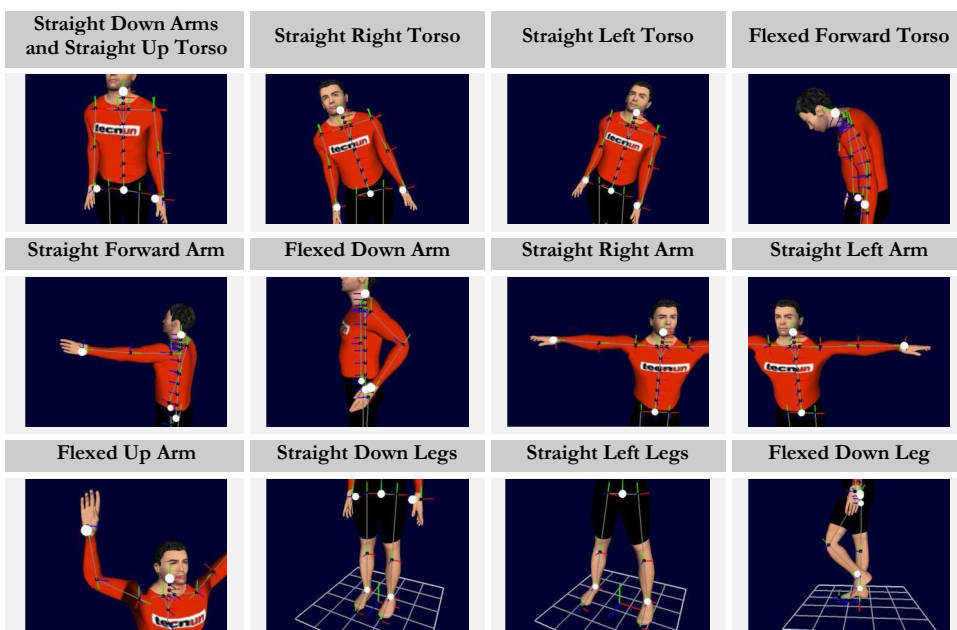


Table 18: Limb postures from the databases.

Thus, the limb pose databases have a total of 40 samples (5 *tsu*, 1 *tsr*, 1 *tsl*, 1 *tff*, 4 *lasd*, 1 *lasf*, 1 *lafd*, 1 *lafu*, 1 *lasl*, 4 *rasd*, 1 *rasf*, 1 *rafu*, 1 *rasr*, 6 *llsd*, 1 *llsl*, 1 *llfd*, 6 *rlsd*, 1 *rlsr* and 1 *rlfd*) obtained from 8 full-body poses, which have been generated by an animator with the full-body pose reconstruction method described in chapter 4. The combined actions to be recognized in the test (Figure 79) are built from the quasi-static gestures shown in Table 19.

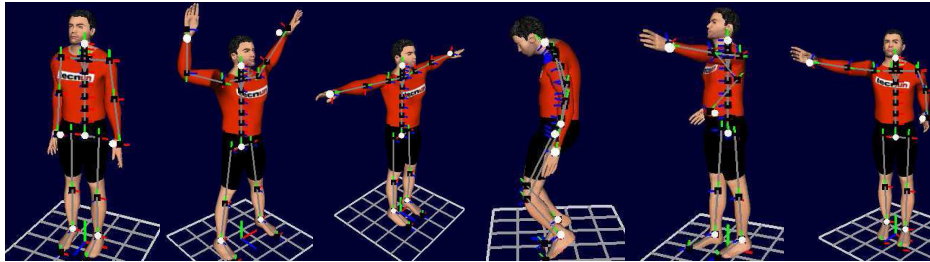


Figure 79: Combined quasi-static gestures to be recognized: *neutral*, *initial*, *cross*, *crouch*, *left punch* and *right punch*.

Combined Action	Limb Pose Combinations
Neutral	<i>tsu & lasd & rasd & llsd & rlsd</i>
Initial	<i>tsu & lafu & rafu & llsl & rlsr</i>
Cross	<i>rasl & rasr</i>
Crouch	<i>tff & llfd & rlfd</i>
Left Punch	<i>lasf & rafu</i>
Right Punch	<i>lafu & rasf</i>

Table 19: Combined motor action semantic descriptions from individual limb pose combinations.

As stated in section 5.6.1 both subjects have remarkable differences in their anthropometries (subject 1 is 1.78 m tall, while subject 2 is 1.62 m), which in the case of using only the image-features for recognition would require the generation of specific databases for each user. Nevertheless, as the full-body pose reconstructions are available, and the limb poses maintain their semantic descriptions by keeping the rotation angles of joints constant even if the body part lengths change from subject to subject, it is possible to use directly the same database for both users. The anthropometry of the virtual humanoid is set to be similar to that of each user during the capture initialization.

The CFOF tracking method (PyrLK + CMDP + KF) for the capture of hands, head and feet uses the parameters shown in Table 7 (chapter 3). On the other hand, the grid used to alleviate the pelvis position tracking calculations is of 4 pixels. Both the Gaussian RGB model and chroma-key background

subtraction methods described in section 3.2 have been used (the former for subject 1 and the latter for subject 2). The 3D humanoid (MIRALab 2008) that represents the user has 24 usable joints for the SIK pose reconstruction method (3 for each leg, 10 for the spine, 1 for each clavicle and 3 for each arm). This way, the humanoid has 47 usable DoF (4 for each leg, 27 for the spine, 2 for each clavicle and 4 for each arm). Its mesh has 5,646 vertices and 9,867 faces rendered in OpenGL (SGI 2008) using the Cal3D 3D character animation library (Heidelberger et al. 2006). Additionally, the deformation of the mesh is considered in the areas around the joints in order to obtain smooth mesh transitions from body part to body part while moving. The biomechanical limits have been modeled from anatomical measurements obtained from the books of Kapandji (1974; 1982; 1988), and the collision avoidances explained in section 4.6 have also been considered. Depth warping of the tracked body parts is also considered as explained in section 5.4.

Figure 80 and Figure 81 show some samples of the real-time markerless full-body motion capture during the interaction between the user and the computer with the considered combined actions of subjects 1 and 2, respectively. It can be observed that the humanoid performs similar poses to those of both users taking into account the depth variations, which are especially noticeable in the punch actions.

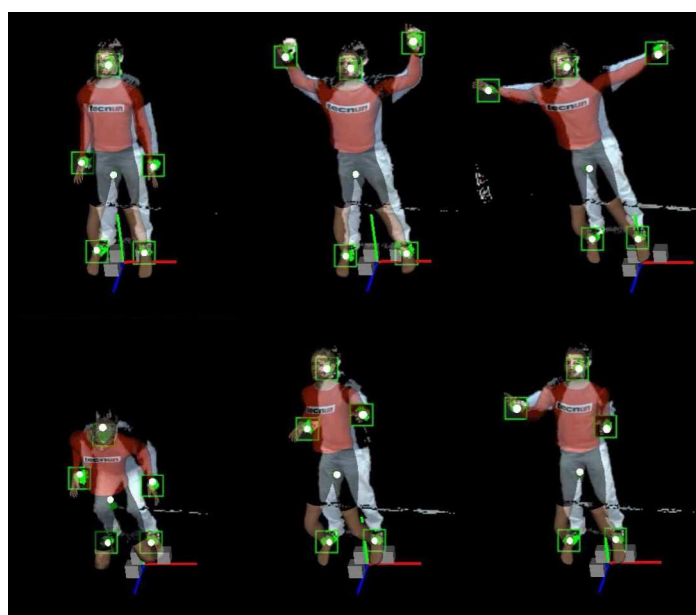


Figure 80: Interaction examples of subject 1 with Gaussian RGB model background: *neutral, initial, cross, crouch, left punch and right punch.*

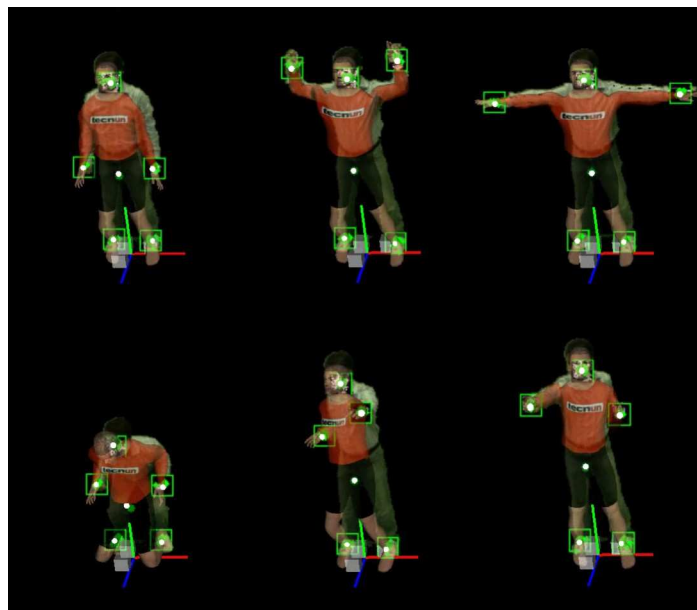


Figure 81: Interaction examples of subject 2 using chroma-key background subtraction: *neutral, initial, cross, crouch, left punch* and *right punch*.

Thus, the system runs at a median frame rate of 42.97 Hz using the Gaussian RGB background subtraction procedure and 44.38 Hz with the chroma-key technique, including capture, pose reconstruction, motor action recognition and rendering. These are satisfactory frame rates to attain a satisfactory interaction between the user and the computer. It can be observed in Figure 80 that there are more false positives in background subtraction than in the case of chroma-key (Figure 81). These mainly come from the shadows cast by the user on bright surfaces. Nevertheless, these do not affect the calculation of the pelvis position.

Regarding the interaction, the system may be visualized as six mouse devices to be controlled at the same time by the user with full-body movements. A learning process is required to get a novice user to operate it. Hence, as the evaluation of the system depends on the user's skill to control it, an alternative method has been adopted to demonstrate its suitability for HCI applications. The response of the system to every projection of the humanoid's hands, head and feet with respect to a still coordinate system attached to the pelvis (i.e., the pose model for database search of section 5.3) in every pixel of the image is evaluated. In this way, the quasi-static gesture recognition areas of the limbs are obtained (Figures 82-86). Note that the humanoid limbs are not

able to reach every pixel of the image, and therefore these areas are constrained by the 3D model.

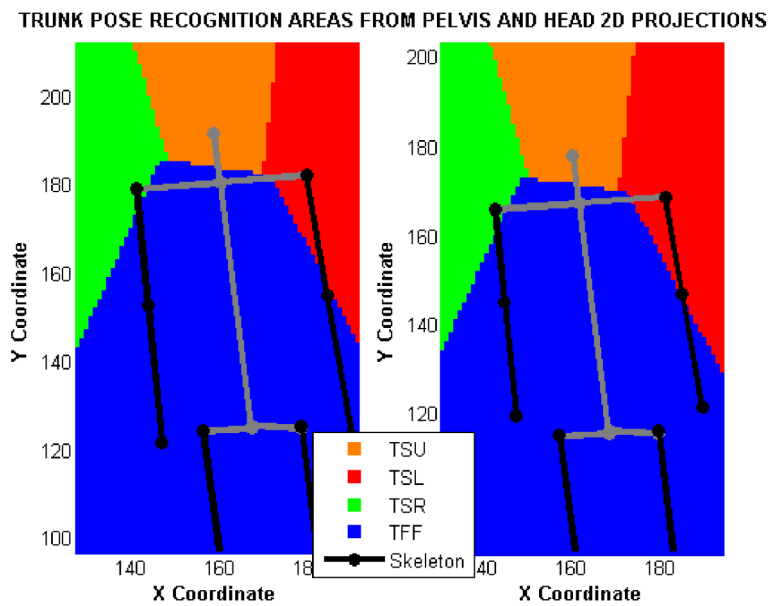


Figure 82: Trunk pose recognition areas in subjects 1 and 2.

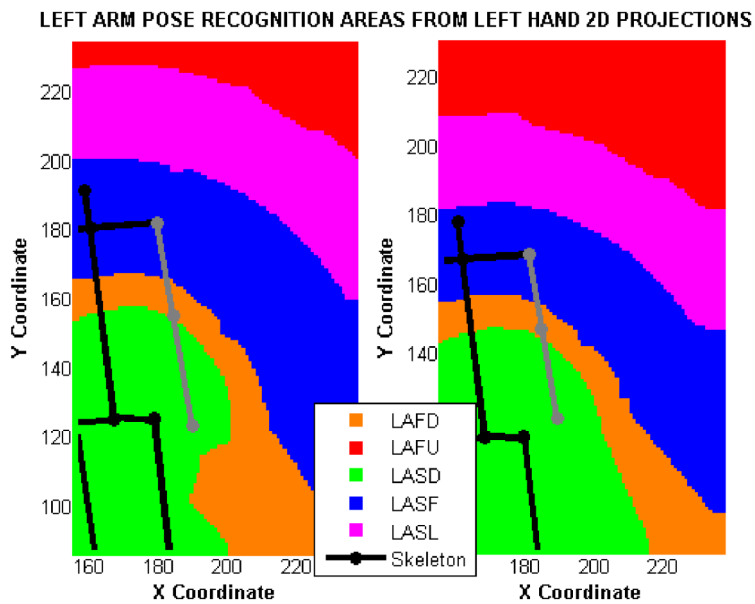


Figure 83: Left arm pose recognition areas in subjects 1 and 2.

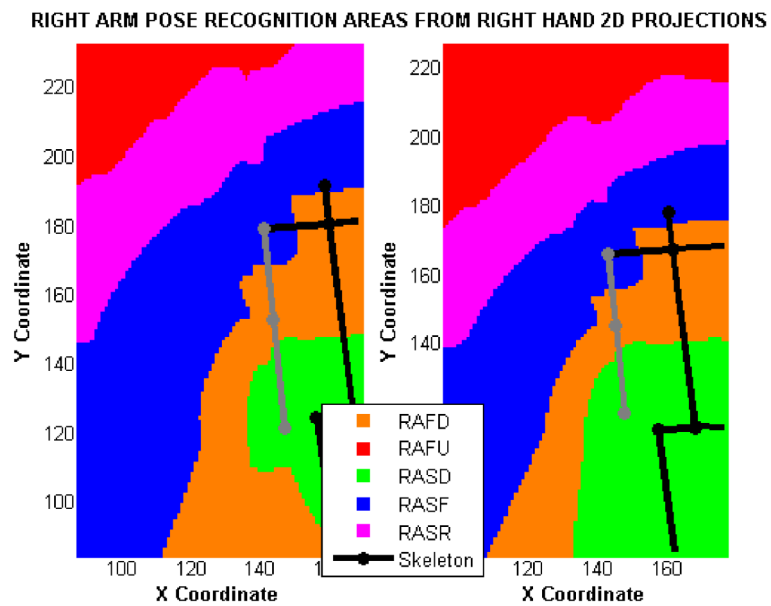


Figure 84: Right arm pose recognition areas in subjects 1 and 2.

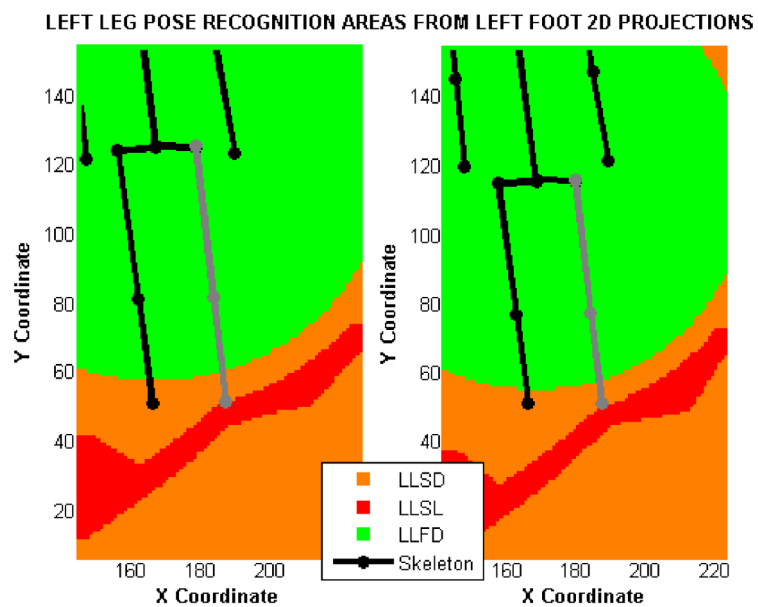


Figure 85: Left leg pose recognition areas in subjects 1 and 2.

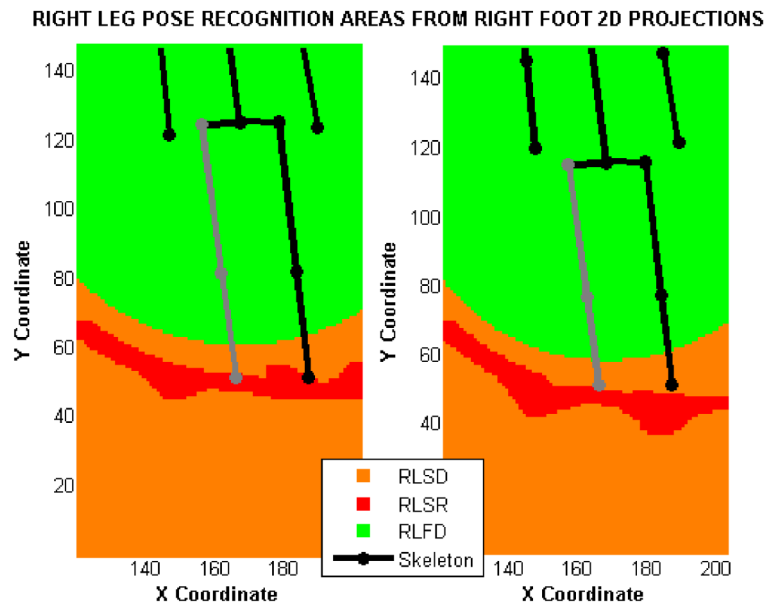


Figure 86: Right leg pose recognition areas in subjects 1 and 2.

These figures show that perfectly recognizable image regions are outlined which represent the areas where the limb poses receive a specific semantic description. It can be observed that they indeed correspond to the projections of the limb poses recorded in the database, especially in their neighborhoods. It can also be stated that the areas corresponding to *llsl* and *rlsr* labels are small compared to the surrounding ones, which makes it more difficult for users to step on them. This occurs because there is only one sample for each of them in the database while there are six for their respective neighbors *llsd* and *rlsd*, and also because there is a small distance between them. This situation can be enhanced with the inclusion of more *llsl* and *rlsr* samples and more distance between *llsd* and *rlsd*. Thus, it can be concluded that skilled users are indeed able to consciously perform combined actions for HCI applications by setting their end-effectors in the proper recognition areas when desired.

CHAPTER 6

CONCLUDING REMARKS

6.1 CONCLUSIONS

This thesis project has presented a strategy composed of a collection of methods for markerless real-time capture and automatic interpretation of full-body human movements for human-computer interaction (HCI). Three stages have been distinguished for the fulfillment of this objective corresponding to three lowest abstraction layers of non-verbal communication: (1) *motion capture*, (2) *pose reconstruction* and (3) *motor action recognition*. The first layer, motion capture, refers to the extraction and tracking of the user's features from images taken by cameras. Pose reconstruction refers to the estimation of the subject's body configuration from the tracked image-features. Finally, the motor action recognition layer refers to the semantic description of the body pose sequences through time. There is a higher abstraction layer in non-verbal communication, *activity interpretation*, which refers to the semantic description of complex psychomotor tasks involving the labeled motor actions and the interaction between the user and the context. This higher layer has not been studied in this thesis project. The conjunction of these stages has to be solved in real time in order to attain a satisfactory HCI. The main contributions and conclusions obtained from this work are the following.

1. Motion Capture Layer:

- A markerless 2D strategy for real-time tracking of the subject's hands, head and feet has been designed. This approach can be catalogued according to the literature taxonomy as a *template based kernel tracking* combined with a *point tracking statistical method*. Experimental results have shown that it can handle different skin-colors and clothing,

obtaining low noise positions and allowing fast movements. It has been compared to other tracking alternatives, including the *flock of features* of Kölsch and Turk (2004), the Pfänder (Wren et al. 1997) and the Condensation (CONDitional DENSity propagATIION) or particle filtering algorithm (Isard and Blake 1998), obtaining better overall results. Additionally, how this method can handle partial occlusions along the path, and how it is possible to detect severe occlusions (due to the overlapping of tracking regions or with non-tracked body parts) and how to make the system react to these, has been demonstrated. This 2D tracking strategy can be directly extended to 3D with the use of depth-sensing cameras.

- A markerless 2D strategy for the real-time tracking of the subject's pelvis has been designed. This approach can be catalogued according to the literature taxonomy as *point tracking statistical method*. It makes use of the user's silhouette which is obtained with background subtraction techniques. Two methods have been presented for the extraction of the silhouette: (1) for static backgrounds and (2) with the use of a backdrop. Both run at a satisfactory frame rate for HCI. How casted shadows can easily be removed in the former when the background is darker than the user has been demonstrated. With respect to the latter, an algorithm has been shown that allows removing shadows even when backdrop folds change, which improves the classical chroma-key technique, which directly removes the G or B channel (depending on the backdrop color) in RGB color space.
- A real-time markerless strategy to track full-body movements for a pseudo 3D motion reconstruction (using a single standard camera), or full 3D (using a single depth-sensing camera) has been designed. This procedure can be catalogued as a *top-down end-effector driven* tracking approach. It has been seen in the literature revision that the problem of estimating the 3D poses of the subject from a single camera is the most ill-posed one due to the perspective ambiguities and self-occlusions. Experimental results show that the obtained postures are of sufficient quality for a subsequent action recognition procedure.

2. Pose Reconstruction Layer:

- A real-time human pose reconstruction procedure for under-guided kinematic problems has been designed, which considers the possible relative rotations of all vertebrae and the shoulder complex, but ignores those of the fingers and toes. The minimum considered input data are the positions of pelvis, head, hands and feet, while the maximum are

both the positions and orientations of those body parts along with the positions of elbows and knees. No pre-recorded full-body pose database is needed for this approach. Along with this, a novel method to estimate the orientation of the root joint from the minimal data applicable to a wide range of movements has been presented. Experimental results with real data (CMU 2008) show that very fast and visually acceptable reconstruction results are obtained, good enough for further motor action recognition. This method has been compared to other well-known robot inverse kinematics methods; the method of Kulpa et al. (2005), the CCD (Wang and Chen 1991), the Jacobian Transpose (Balestrino et al. 1984; Wolovich and Elliot 1984), Pseudoinverse (Whitney 1969), the DLS (Nakamura and Hanafusa 1986; Wampler 1986), the DLS with SVD (Maciejewski 1990; Maciejewski and Klein 1988), the SDLS (Buss and Kim 2005), the PIK (Baerlocher and Boulic 2004; Peinado et al. 2004), and the method of Tolani et al. (2000) for anthropomorphic limbs, obtaining better overall results. How this method can be adapted for the use of a marker-based optical mocap system which also obtains visually satisfactory full-body poses in real time has also been shown.

- A simple and efficient method for modeling complex biomechanical joint limits using only a few biomechanical measurements has been presented. Along with this, a real-time strategy for setting joints within biomechanical limits is presented. The advantage of this strategy is that, even with few true anatomical measurements, the full complex circumduction limit surface of the swing movements of joints, such as the shoulders, can be modeled with a high degree of realism.
- Real-time collision-avoidance strategies to prevent elbow-torso, wrist-torso and foot-floor interpenetrations have been designed. The mesh-surfaces of the graphics are considered for penetration measurement. These approaches allow enhancing the reconstruction quality obtained with the robot inverse kinematics method by setting the arms and legs within realistic configurations, supported also by the biomechanical rotation limit constraints.

3. Motor Action Recognition Layer:

- A procedure for potentially meaningful *gesture spotting* in a real-time continuous motion data flow, which relies on the history of the kinetic status of the tracked body part data, has been presented. Experimental results have shown that it can correctly determine the starting and ending time instants of potentially meaningful gestures performed at

different speeds, with time warping, which needs little computation time and is of sufficient quality for HCI applications.

- A method for the recognition of quasi-static and dynamic gestures, based on the K-Nearest Neighbors (K-NN) approach (Dasarathy 1991), has been presented. Experimental results have shown that it can satisfactorily determine to which class gestures correspond for HCI applications in a wide range of gesture sizes. How the presented method of storing movements in the database allows a well-defined clusterization of different gestures has also been shown. The classification has also been compared to other methods, such as the SVM (Shawe-Taylor and Cristianini 2000) and the Random Forest (Breiman 2001), obtaining better overall results.
- A method for the enhancement of single-camera pseudo 3D motion reconstruction, based on pose recognition, which allows a continuous depth warping of the tracked body parts has been presented. It has been shown that this method obtains more realistic reconstructions when the considered actions are already known.
- A strategy to recognize combined actions, based on proper database storage of human motion patterns and poses, has been presented. Experimental results have shown that taking advantage of the reconstructed poses it can be easily adapted to the different anthropometries of users without changing the known databases. It has also been shown that this strategy allows more complex interactions between the subject and computer in HCI applications as motor actions can easily be recognized in a wider set of poses than those offered by *holistic* procedures.

6.2 FUTURE WORK

Several possible research lines are open in order to continue with the work presented in this thesis project. Once again, these are described according to the non-verbal communication abstraction layers in which they are circumscribed.

1. Motion Capture Layer:

- The simultaneous tracking of the head, pelvis, hands and feet positions of multiple users can be explored for more complex HCI applications. This requires labeling the silhouette pixels corresponding to the

different users, in order to calculate properly the pelvis positions, and introduces more complex occlusion situations.

- The current algorithms run satisfactorily above the real-time frame rate, but by improving memory management the frame rate can be optimized, especially for multiple user motion capture. It may be improved using, e.g., multithreading programming, or implementing the computer vision algorithms in the GPU with libraries such as OpenVIDIA (Fung et al. 2005) or GPUCV (Farrugia et al. 2006).
- Taking advantage of the similar chrominance characteristics of skin-color despite the user's race, how to fully automatize the tracking initialization can be studied. With respect to the feature point tracking, its search should be performed in every frame instead of only the first one. This way, their motion correspondence from frame to frame (currently done by optical flow) can be readapted to this situation. This continuous search may also be helpful for recovering from severe occlusions.
- The shape of the silhouette can be exploited, especially if a depth-sensing camera is used, for a better correspondence between the limbs of the subject and the humanoid, and also for a better anthropometry estimation.
- Background subtraction techniques can also be improved to make the system usable in noisy environments. This is currently one of the toughest research areas in the computer vision field.
- The possibility of tracking with a moving camera can be explored. This is also a challenging task as it involves the most complex background subtraction problem along with the need of calibrating the camera frame by frame.

2. Pose Reconstruction Layer:

- The presented root joint orientation estimation method which makes use of the tracked positions of hands, head, pelvis and feet is valid for a broad range of movements. However, it may be generalized to every kind of movement, paying special attention to the weight values that may be dynamically adapted in order to avoid abrupt changes or odd configurations in the spine's reconstruction.
- Self-collision avoidance can also be extended taking into account all possible combinations, not only the elbows and wrist with the torso,

and the feet with the floor. In addition, the use of deformable models to define the body part shapes may also be explored. Attention may also be paid to collision avoidance of the humanoid with other objects in a given scenario, such as for reaching tasks.

- It may also be interesting to explore the possibility of combining the presented reconstruction method with PIK (Baerlocher and Boulic 2004) in order to obtain fast reconstruction due to the analytic calculations of our approach and to have the option of adding prioritized constraints like in PIK, which would be helpful for improving reconstruction quality.
- The use of databases to improve reconstruction quality may also be explored taking advantage of the low computation time of our approach. For example, an initial posture could be obtained from the database and then the readjustment process would be carried out by the reconstruction method presented here.

3. Motor Action Recognition Layer:

- The gesture spotting procedure can be enhanced by taking into account not only the kinetic status variation of the tracked features, but also checking the temporal advance of the observations with respect to the known patterns. This way, a more robust strategy for time warping that simultaneously traces gestures at lower speeds, as the kinetic status threshold would not be the only factor to segment the potentially meaningful gestures from the data flow. Some efforts in temporal advance measurement have already been made in the work of Mena et al. (2008), corresponding to one of the publications generated during this thesis project.
- The gesture recognition procedure allows the use of a distance to measure the spatio-temporal differences of new gestures with respect to the known patterns, which opens up the possibility of analyzing their level of imitation. This mimesis can be used for skills acquisition and transfer tasks from expert to novice users in motor actions, which could be from human to human, or even from computer to human or human to computer, where the computer may be in the form of a robot. Again, the gesture recognition procedure may be improved by measuring the temporal advance, as a response can be given at every frame during the gesture performance, which allows correcting the body part trajectories.

- The work done at this layer, especially if a real-time response on the motor action recognition during the performance is given, can be taken into account in order to improve pose reconstruction and motion capture tasks during severe occlusion situations.
- The automatic selection of the meaningful features to be extracted from the reconstructed motion for further and more complex motor action recognition may be explored. The work of Liu et al. (2006) for the automatic search of the *principal markers* from a full set of markers attached to the human body in a marker-based optical mocap system may be a good basis for this. However, this automatization would make the database knowledge storage and management more intricate, especially if combined actions are expected to be identified, as observations would change dynamically. Thus, this problem constitutes a challenging research line.

4. Activity Interpretation Layer:

- The inclusion of other explicit subject, object and environment models opens up a wider set of research lines for higher level scene interpretations, but also to enhance some of the lower, especially the occlusion and background subtraction challenging problems.
- The inclusion of other modes of interfacing with the computer, such as voice and touch, increases the level of complexity in the interaction, but how these can be combined in an optimal way for applications such as the acquisition and transferring of psycho-motor skills, intelligent tutoring systems and automatic surveillance systems, is still a work to be done. But that is another story...

GENERATED PUBLICATIONS

The following reference list corresponds to the publications generated during the thesis project work related to it, ordered from newer to older. These include contributions to congresses (in the form of articles, posters and presentations) and journals. The following pages present the first page of the published articles.

Vélaz, Y., **Unzueta, L.**, and Suescun, Á. (2008). "Fast Human 3D Voxelized Shape Reconstruction for Human-Computer Interaction." *NAUN International Journal of Computers*, 2(4), 371-380.

Unzueta, L., Peinado, M., Boulic, R., and Suescun, Á. (2008). "Full-Body Performance Animation with Sequential Inverse Kinematics." *Graphical Models*, 70, 87-104.

Unzueta, L., Mena, O., Sierra, B., and Suescun, Á. (2008). "Kinetic Pseudo-Energy History for Human Dynamic Gestures Recognition." *Proceedings of the Conference on Articulated Motion and Deformable Objects, LNCS 5098*, Pto. Andratx, Mallorca, Spain, 390-399.

Mena, O., **Unzueta, L.**, Sierra, B., and Matey, L. (2008). "Temporal Nearest End-Effectors for Real-Time Full-Body Human Actions Recognition." *Proceedings of the Conference on Articulated Motion and Deformable Objects, LNCS 5098*, Pto. Andratx, Mallorca, Spain, 269-278.

Unzueta, L. (2008). "Human Motion Capture." Oral Presentation in the Tutorial "Skills Capture and Transfer." Organizers: Avizzano, C. A., and Ruffaldi, E. *Robotics: Science and Systems Conference*, Zurich, Switzerland. Slides in: <http://www.skills-ip.eu/events/rss08/>.

Vélaz, Y., **Unzueta, L.**, and Suescun, Á. (2008). "Human 3D Shape Accelerated Reconstruction for Real-Time Markerless Motion Capture." *Proceedings of the ENMA Electric International Conference*, Bilbao, Spain, 187-193.

- Unzueta, L.**, Velaz, Y., and Suescun, Á. (2007). "Markerless Motion Capture & Reconstruction." *Poster of the International Workshop on MultiModal Interfaces for the Transfer of Human Skills*, Pisa, Italy.
- Ruffaldi, E., Avizzano, C. A., Gopher, D., Mottet, D., La Garde, J., **Unzueta, L.**, Mena, O., Suescun, Á., and Matey, L. (2007). "Digital Representation of Skills." *Poster of the International Workshop on MultiModal Interfaces for the Transfer of Human Skills*, Pisa, Italy.
- Boulic, R., Varona, J., **Unzueta, L.**, Peinado, M., Suescun, Á., and Perales, F. (2006). "Evaluation of On-line Analytic and Numeric Inverse Kinematics Approaches Driven by Partial Vision Input." *Virtual Reality*, 10(1), 48-61.
- Boulic, R., Varona, J., **Unzueta, L.**, Peinado, M., Suescun, Á., and Perales, F. (2005). "Real-Time IK Body Movement Recovery from Partial Vision Input." *Proceedings of the International Conference on ENACTIVE Interfaces*, Genova, Italy. **Awarded the "Best Collaborative Paper of the European Union ENACTIVE Network of Excellence"**.
- Unzueta, L.**, Berselli, G., Cazón, A., Lozano, A., and Suescun, Á. (2005). "Genetic Algorithms Application to the Reconstruction of the Human Motion using a Noninvasive Motion Capture." *Proceedings of the ECCOMAS Thematic Conference on Advances in Computational Multibody Dynamics*, Madrid, Spain, 137.
- Boulic, R., Varona, J., Hervelin, B., **Unzueta, L.**, Suescun, Á., Jaume, A., Perales, F., and Thalmann, D. (2005). "Vision-Based Comparative Study of Analytic and Numeric Inverse Kinematic Techniques for Recovering Arm Movements." *ENACTIVE Workshop*, Pisa, Italy.
- An Internet video showing the results of this thesis project can be found here:
- Unzueta, L.** (2008). "Markerless Human Motion Capture and Motor Action Recognition." *Video on Youtube*,
<http://www.youtube.com/watch?v=vqYursD7OCQ>.

Fast Human 3D Voxelized Shape Reconstruction for Human-Computer Interaction

Yaiza Vélaz, Luis Unzueta, and Ángel Suescun

Abstract—At present, there is an increasing interest in multimodal interaction, which supports multiple modes of interaction of the user with the computer. Among these modes of interfacing, those obtained from face expressions, voice, body postures and movements are the most important and researched within a human face-to-face interaction context. Some of the tools used to achieve this communication are: motion capture, speech recognition and force feedback. Thus, in order to carry out the human-computer interaction (HCI), some features and parameters are needed, which can be extracted via cameras or haptics depending on the communication mode.

The 3D shape of the target person constitutes one of the most used features in markerless motion capture for a posterior pose and motion pattern recognition, which can be obtained from a multi-camera system. The process starts by subtracting the background from the images in order to get the projected silhouettes of the subject. Then, having established the relation between the real 3D world and the camera projections, the 3D shape reconstruction can be attained with voxel carving methods. The reconstruction quality depends on the number of cameras, the size of the voxels and the used method. The more precise the 3D shape reconstruction is, the higher the computational cost will be, which may prevent its use for HCI applications.

We present a novel approach that accelerates existing voxel carving methods. It goes from coarse to fine, preserving the matching with the captured silhouettes, and thus achieving a good reconstruction quality. Results show its suitability for real-time markerless motion capture, applicable to HCI applications, such as videogames.

Keywords—Voxel Carving, Human-Computer Interaction, Markerless Motion Capture, Computer Vision.

I. INTRODUCTION

In the last decades many efforts have been invested in representing a real moving human inside the 3D virtual world, i.e., human motion capture, due to the variety of applications this computer vision field presents, including virtual reality, surveillance, or motor skills capture and transfer, among others.

In 1994, Laurentini [1] defined the foundations of the 3D space reconstruction, although before those days some related papers had already been published to reconstruct 3D objects from silhouettes [2-6]. Laurentini defined and demonstrated the *visual hull* of an object S as the closest approximation that gives the same silhouette of S obtained from all views outside the convex hull of the object. However, as stated by Slabaugh et al. [7], volumetric representations use a finite number of viewpoints computing the inferred visual hull. Many of the methods which implement the 3D reconstructions from silhouettes are based on Shape-From-Silhouette (SFS), or *voxel carving*. A voxel representation has the advantage that it is simple to implement, and additionally it is a look-up table, i.e., the voxel structure can be accessed with a simple array indexing operation.

The resulting 3D space representation can be used as input for *voxel coloring* or *space carving* [8-14], which are methods to colorize the 3D reconstruction in order to get a more realistic representation.

The coloring process has the handicap of needing to determine which parts of the reconstruction are visible to the observers, in order to assign the proper color to each voxel. This color information is useful among other applications, for the identification and tracking of the positions and orientations of body parts.

In this paper, an algorithm for human 3D shape reconstruction that accelerates current voxel carving approaches is presented. The algorithm is additionally quickened by implementing part of the system with parallel threads (those processes that can be parallelized). Thus, the procedure achieves real-time performance for small voxel sizes, obtaining good reconstruction quality results compared to the approaches that have been accelerated. Along with this procedure, we show two fast approaches to extract the user's silhouettes from static backgrounds: (1) for general color static backgrounds and (2) for backdrops with a predefined color (*chroma-key*). The latter procedure enhances the classical chroma-key technique, as it can handle more robustly with casted shadows, including those coming from backdrop folds, and also even if those folds change during the subject's performances. Finally, we evaluate our algorithm accelerating an existing reconstruction approach, using as input data four points of view, in order to extract the four silhouettes of a subject in each frame.

Manuscript received May 28, 2008; Revised version received October 14, 2008.

Y. Vélaz, L. Unzueta and A. Suescun are in the Applied Mechanics Department in the CEIT and Tecnun of the University of Navarra, Paseo de Manuel Lardizabal, 15, 20018, Donostia-San Sebastián (Spain).

E-mail: yvelaz@ceit.es, lunzueta@ceit.es, asuescun@ceit.es.

Graphical Models 70 (2008) 87–104



Contents lists available at ScienceDirect

Graphical Models

journal homepage: www.elsevier.com/locate/gmod

Full-body performance animation with Sequential Inverse Kinematics

Luis Unzueta^{a,*}, Manuel Peinado^b, Ronan Boulic^c, Ángel Suescun^a

^aDepartment of Mechanics, CEIT and Tecnun, University of Navarra, Manuel de Lardizabal 15, 20018 San Sebastián, Spain

^bEscuela Politécnica, University of Alcalá, Spain

^cVirtual Reality Laboratory, Ecole Polytechnique Fédérale de Lausanne, Switzerland

ARTICLE INFO

Article history:

Received 25 May 2007

Received in revised form 26 February 2008

Accepted 27 March 2008

Available online 8 April 2008

Keywords:

Inverse Kinematics
Motion reconstruction
Human animation

ABSTRACT

In this paper, we present an analytic-iterative Inverse Kinematics (IK) method, called Sequential IK (SIK), that reconstructs 3D human full-body movements in real time. The input data for the reconstruction is the least possible (i.e., the positions of wrists, ankles, head and pelvis) in order to be usable within a low-cost human motion capture system that would track only these six features. The performance of our approach is compared to other well-known IK methods in reconstruction quality and computation time obtaining satisfactory results for both. The paper first describes how we handle the spine and the clavicles before offering a simple joint limit model for ball-and-socket joints and a method to avoid self-collisions induced by the elbow. The second part focuses on the algorithms comparison study.

© 2008 Elsevier Inc. All rights reserved.

1. Introduction

In performance animation, the captured movements of human performers are mapped in real time to the skeletons of virtual characters that represent their poses. The applications of this mapping are multiple, like TV production, virtual reality, workspace design, entertainment, human-machine interaction, skills acquisition and rapid prototyping of animations. Most systems addressing performance animation require the use of large sensor or marker sets, making them too cumbersome and expensive for low-cost applications such as home entertainment.

Human motion is typically represented as a series of different configurations of a rigid multibody mechanism consisting of a set of segments connected by joints. Segments correspond to body parts such as the thighs, shanks, upper arms, forearms, etc., and joints correspond to articulations such as hips, knees, shoulders, elbows, etc. These joints are hierarchically ordered and have one or more Degrees of Freedom (DoF) which represent the rotations

relative to their parent joints. There is a root joint of which the position and orientation are represented with respect to the absolute coordinate system. A good example of this way of representing humanoid characters is the H-Anim standard [1]. This standard places the root joint at the pelvis, and defines a standard name for each joint, as well as a standard reference, or neutral, posture. In the context of this work we adhere to this standard.

It is often too cumbersome and time-consuming for an animator to manually set all the DoFs of a virtual character. This is solved by Inverse Kinematics (IK) techniques, in which only the positions (or sometimes also the orientations) of certain joints, usually the end-effectors, must be specified by the animator or by the motion capture system. The remaining DoFs are automatically determined according to different criteria that depend on the IK variant one employs. End-effector positions can be modeled as a function of the DoFs, leading to formal definition of the IK problem as $f(\mathbf{q}) = \mathbf{G}$, where \mathbf{q} is the vector of DoFs and \mathbf{G} is a vector that gathers all the desired end-effector positions. This problem is highly under-constrained as \mathbf{q} usually has a much larger dimension than \mathbf{G} . In addition, it is a nonlinear problem as f involves complex combinations of trigonometric functions.

* Corresponding author. Fax: +34 943 213076.
E-mail address: lunzueta@ceit.es (L. Unzueta).

Kinetic Pseudo-energy History for Human Dynamic Gestures Recognition

Luis Unzueta¹, Oscar Mena¹, Basilio Sierra², and Ángel Suescun¹

¹ CEIT and Tecnun, University of Navarra, Manuel de Lardizabal 15, 20018 Donostia-San Sebastián, Spain

{lunzueta, omena, asuescun}@ceit.es

<http://www.ceit.es/mechanics/index.htm>

² Computer Engineering Faculty, University of the Basque Country, Manuel de Lardizabal 1, 20018 Donostia-San Sebastián, Spain

b.sierra@ehu.es

<http://www.sc.ehu.es/ccwrobot/index.html>

Abstract. In this paper we present a new approach, based on the kinetic status history, to automatically determine the starting and ending instants of human dynamic gestures. This method opens up the possibility to distinguish static or quasi-static poses from dynamic actions, during a real-time human motion capture. This way a more complex Human-Computer Interaction (HCI) can be attained. Along with this procedure, we also present a novel method to recognize dynamic gestures independently from the velocity with which they have been performed. The efficiency of this approach is tested with gestures captured with a triple axis accelerometer, and recognized with different statistical classifiers, obtaining satisfactory results for real-time applications.

Keywords: Human Action Recognition, Human Tracking, Kinetic Pseudo-Energy.

1 Introduction

At present, marker-based systems are the most popular human motion capture (mocap) systems. We can distinguish mainly among optical, magnetic, mechanical and inertial mocap systems. Depending on the system, the data provided can be the marker positions, orientations, accelerations, etc. Users need to wear these markers all over the body in specific configurations and the system must be calibrated properly which, along with their high cost, makes this method cost prohibitive for home-users. Nevertheless, more recently, new devices that have embedded accelerometers have appeared in order to obtain the user's movements at an affordable cost, such as the Nintendo Wii [6] video game console remote, and the Apple iPhone [5] and Sony Ericsson W910i [7] mobile phones. On the other hand, in recent years, optical markerless mocap systems have been developed. Cameras are capable of detecting pixel colors and intensities and from

Temporal Nearest End-Effectors for Real-Time Full-Body Human Actions Recognition

Oscar Mena¹, Luis Unzueta¹, Basilio Sierra², and Luis Matey¹

¹ CEIT and Tecnun, University of Navarra. Manuel de Lardizabal 15, 20018 Donostia-San Sebastián, Spain
{omena,lunzueta,lmatey}@ceit.es
<http://www.ceit.es/mechanics/index.htm>

² Computer Engineering Faculty, University of the Basque Country. Manuel de Lardizabal 1, 20018 Donostia-San Sebastián, Spain
b.sierra@ehu.es
<http://www.sc.ehu.es/ccwrobot/index.html>

Abstract. In this paper we present a novel method called Temporal Nearest End-Effectors (TNEE) to automatically classify full-body human actions captured in real-time. This method uses a simple representation for modeling actions based exclusively on the recent positions of the user's end-effectors, i.e. hands, head and feet, relative to the pelvis. With this method, the essential information of full-body movements is retained in a reduced form. The recognition procedure combines the evaluation of the performed poses and the temporal coherence. The performance of TNEE is tested with real motion capture data obtaining satisfactory results for real-time applications.

Keywords: Human Action Recognition, Human Tracking, Articulated Motion, End-Effectors.

1 Introduction

One of the biggest issues that an action recognition system has to overcome is to define a representation of actions, i.e. to establish the features that allow the classification of movements. These features are data extracted from the information provided by the motion capture (mocap) system and they are directly dependent on the way it works. If movement is captured using image analysis, the features are extracted from changes that appear in video frames through time. On the other hand, if movement is captured using physical markers located on the user's body, features are extracted from the positions, angles, angle velocities, etc. of these markers.

As stated by Liu et al. [1] data that comes from most mocap systems exhibit considerable redundancy and demonstrate that there can be a reduced set of information, the principal markers, which retain the essential information of movement. In their work they reconstruct full-body postures using as input data only the positions of some markers situated in the areas around the hands,

HUMAN 3D SHAPE ACCELERATED RECONSTRUCTION FOR REAL-TIME MARKERLESS MOTION CAPTURE

Yaiza Vélaz¹, Luis Unzueta¹, Ángel Suescun²
yvelaz@ceit.es lunzueta@ceit.es asuescun@ceit.es

¹CEIT and Tecnum
University of Navarra
Paseo de Manuel Lardizabal, N° 15, 20018
Donostia-San Sebastián (Spain)

Abstract. In the recent times there has been an increasing interest in multimodal interaction, which combines seamlessly multiple modes of interfacing with the computer. In the case of robots, the multimodal interaction can be used with the aim of learning by demonstration. Among these modes of communication (motion capture, speech recognition and force feedback), those obtained from face expressions, body postures and movements, represent most of the transferred information in a human face-to-face interaction. Needed features for this kind of interaction can be captured using cameras and applying computer vision algorithms.

The 3D shape of the target person constitutes one of the most used features in markerless motion capture for a posterior pose and motion pattern recognition, which can be obtained from a multi-camera system. Hence, in this paper we deepen in the process of obtaining this human 3D shape. Firstly, the background is subtracted in order to get the projected silhouettes. Then, the 3D reconstruction of the shape can be achieved from these projections by a voxelization procedure applied to the calibrated views. The more number of cameras and the finest granularity level of space subdivision, the more precise 3D shape reconstruction, but the higher computational cost.

In order to accelerate this process, we present a novel method that goes from coarse to fine, preserving the matching with the captured silhouettes, and thus maintaining the reconstruction quality of the finest granularity level voxelization. Results show its suitability for real-time markerless motion capture, applicable to Human-Computer Interaction, Robot Vision and rendering for Virtual Reality.

1 INTRODUCTION

In the last decades there has been an increasing interest in representing a static scene or a moving human inside the 3D virtual world. In 1994 Laurentini (6) defined the visual hull of an object S as the closest approximation that gives the same silhouette of S obtained from all views outside the convex hull of the object. However, as stated by Slabaugh et al. (7), volumetric representations use a finite number of viewpoints computing the inferred visual hull. Many of the methods which implement it are based on Shape from Silhouette (SFS), or voxel carving. A voxel representation has the advantage that it is simple to implement, and additionally it is a look-up table, i.e., the voxel structure can be accessed with a simple array indexing operation.

Another spatial division technique are octrees. An octree starts from a bounding cube, which is divided into smaller cells or octants in case it corresponds to the subject being captured instead of the background. It works efficiently when there are large space regions as demonstrated by Dyer (3). Thus, it is usually used for scene reconstruction. On the contrary, when the space has little free spaces the algorithm becomes slower as more operations have to be undertaken.

Related to voxelization algorithms, it is interesting to colorize the voxels and make a more realistic reconstruction of a moving subject or scene. A voxel coloring process has the handicap

Virtual Reality (2006) 10: 48–61
DOI 10.1007/s10055-006-0024-8

ORIGINAL ARTICLE

Ronan Boulic · Javier Varona · Luis Unzueta
Manuel Peinado · Angel Suescun · Francisco Perales

Evaluation of on-line analytic and numeric inverse kinematics approaches driven by partial vision input

Received: 20 December 2005 / Accepted: 31 March 2006 / Published online: 21 April 2006
© Springer-Verlag London Limited 2006

Abstract Despite its central role in the constitution of a truly enactive interface, 3D interaction through human full body movement has been hindered by a number of technological and algorithmic factors. Let us mention the cumbersome magnetic equipments, or the underdetermined data set provided by less invasive video-based approaches. In the present paper, we explore the recovery of the full body posture of a standing subject in front of a stereo camera system. The 3D position of the hands, the head and the center of the trunk segment are extracted in real-time and provided to the body posture recovery algorithmic layer. We focus on the comparison between numeric and analytic inverse kinematics approaches in terms of performances and overall quality of the reconstructed body posture. Algorithmic issues arise from the very partial and noisy input and the singularity of the human standing posture. Despite stability concerns, results confirm the pertinence of this approach in this demanding context.

Keywords Inverse kinematics · Motion capture · On-line image analysis

R. Boulic (✉)
Virtual Reality Laboratory, Ecole Polytechnique Fédérale de
Lausanne, Station 14, 1015 Lausanne, Switzerland
E-mail: Ronan.Boulic@epfl.ch
Tel.: +41-21-6935246
Fax: +41-21-6935328

J. Varona · F. Perales
Dept Mat. i Informatica, Universitat de les Illes Balears (UIB),
Balears, Spain
E-mail: vdmijvg4@uib.es
E-mail: Paco.Perales@uib.es

L. Unzueta · A. Suescun
CEIT and Tecnun (University of Navarra), San Sebastian, Spain
E-mail: lunzueta@ceit.es
E-mail: asuescun@ceit.es

M. Peinado
Escuela Politécnica University of Alcalá, Alcalá, Spain
E-mail: manup@aut.uah.es

Abbreviations IK: Inverse kinematics · PIK: Prioritized
inverse kinematics · dof: Degree of freedom

1 Introduction

The sense of movement has been under-exploited until now in classical interfaces. Integrating the kinaesthetic sense at a larger scale than desktop manipulations is fundamental for building effective enactive interfaces where our dexterity and full body postural knowledge can be exploited. Exploiting the sole 3D location of one or two hands is indeed not sufficient for the evaluation of complex tasks in virtual environments. For example, very often a new product to assess is part of a cluttered environment hence raising accessibility issues for the human operator in charge of using or maintaining it. Therefore, it is crucial to be able to easily specify the full body posture of a virtual mannequin for conducting such evaluations on the virtual prototype as early as possible in the conception process. The present paper targets such objective with the long-term aim of offering a real-time non-invasive technology for the intuitive specification of human full body postures while interacting with complex virtual environments. Until now, the exploitation of real-time motion capture of full body human movements has been limited to niche applications such as the expressive animation of a virtual character in a live show (Sturman 1998). Multiple factors hinder a wider adoption of full body movement as a popular 3D user interface. Among others, we can cite: the invasiveness of the sensor system, the limited acquisition space and sensor precision, the spatial distortions, the high dimension of the posture space, and the modeling approximations in the mechanical model of the human body. These sources of errors accumulate and result in an approximate posture. It can be sufficient for performance animation where expression counts the most. However, if precise spatial control is desired, this channel may not suited for evaluating complex interaction with virtual objects.

Real-Time IK Body Movement Recovery from Partial Vision Input

Ronan Boulic^{*} Javier Varona[†] Luis Unzueta[‡] Manuel Peinado^{**}

Angel Suescun[‡] Francisco Perales[†]

(^{*})EPFL, Switzerland

([†])UIB, Spain

([‡])CEIT, Spain

(^{**})University of Alcalá, Spain

E-mail: Ronan.Boulic@epfl.ch, vdmijvg4@uib.es, lunzueta@ceit.es, manupg@aut.uah.es,
asuescun@ceit.es, Paco.Perales@uib.es

Abstract

Despite its central role in the constitution of a truly enactive interface, 3D interaction through human full body movement has been hindered by a number of technological and algorithmic factors. Let us mention the cumbersome magnetic equipments, or the underdetermined data set provided by less invasive video-based approaches. In the present paper we explore the recovery of the full body posture of a standing subject in front of a stereo camera system. The 3D position of the hands, the head and the center of the trunk segment are extracted in real-time and provided to the body posture recovery algorithmic layer. We extend our comparison of two Inverse Kinematics approaches to this more complex context. Algorithmic issues arise from the very partial and noisy input and the singularity of the human standing posture. Despite stability concerns, results confirm the pertinence of this approach in this demanding context.

1. Introduction

The sense of movement has been under-exploited until now in classical interfaces. Integrating the kinaesthetic sense at a larger scale than desktop manipulations is fundamental for building effective Enactive Interfaces where our dexterity and full body postural knowledge can be exploited. Until now the exploitation of real-time motion capture of full body human movements has been limited to niche applications such as the expressive animation of a virtual character in a live show [1]. Multiple factors hinder a wider adoption of full body movement as a popular 3D user interfaces. Among others we can cite:

the invasiveness of the sensor system, the limited acquisition space and sensor precision, the spatial distortions, the high dimension of the posture space, and the modeling approximations in the mechanical model of the human body. These sources of errors accumulate and result in an approximate posture. It can be sufficient for performance animation where expression counts the most. However, if precise spatial control is desired, this channel may not suited for evaluating complex interaction with virtual objects.

The factor we want to improve in the present study is the comfort of the user through a non-invasive vision-based acquisition technology. A prior work has shown the feasibility of the vision-based recovery of the arm posture [2]. We extend the posture recovery to the full body while interacting in front of a workbench. In addition to the 3D position of the hands and the head, we can also exploit an estimate of the center of the trunk segment. The performances of the analytic and the numeric Inverse Kinematics approaches are compared in this highly under-determined context. Additional issues also arise due to the standing human posture which is close to the fully extended singularity. We described the solutions we have experimented to overcome these challenges.



Fig. 1: The simplified body model includes one virtual leg, a simplified spine and two arms.

MULTIBODY DYNAMICS 2005, ECCOMAS Thematic Conference
J.M. Goicolea, J.Cuadrado, J.C.García Orden (eds.)
Madrid, Spain, 21–24 June 2005

GENETIC ALGORITHMS APPLICATION TO THE RECONSTRUCTION OF THE HUMAN MOTION USING A NON- INVASIVE MOTION CAPTURE

Luis Unzueta*, Giovanni Berselli*, Aitor Cazón*, Alberto Lozano* and Ángel Suescun*

*CEIT and Tecnun (University of Navarra)
Manuel de Lardizábal 15, 20018 San Sebastián, Spain
e-mail: lunzueta@ceit.es

Keywords: Inverse Kinematics, Human Motion, Motion Capture, Genetic Algorithms.

Abstract. *Current motion capture systems require the person, whose motion is to be captured, to wear expensive and complex setups of markers, or capture devices such as exoskeletons, thus limiting the number and range of applications of these systems. In the European HUMODAN project (IST-2001-32202) a new non-invasive capture system is being developed whose only input data are the raw images captured by calibrated video cameras. Only some body-features of the person being recorded can be detected this way. In this capture system the detected body-features are the end-effectors of a human body such as the hands, head and feet, and also the orientation of the trunk. The system requires inverse kinematics algorithms to reconstruct the whole 3D human structure. Inverse kinematics can be solved by analytic methods or by an optimization process. This optimization problem is heavily non-linear due to the complexity of the mechanism that represents the human figure and also due to the lack of input data that comes from the capture system. The most used optimization algorithms for inverse kinematics are the gradient-based methods. There are other optimization algorithms like the Genetic Algorithms (GA) that could be applied for this task.*

GAs are well suited for problems that are non-linear but which have solutions whose quality can be easily evaluated, and have a large number of degrees of freedom. GAs generate a population of solutions that are modified applying operators such as crossover and mutation, akin those occurring in the natural selection within biological systems. The selection of the members of a population to be combined or mutated to generate new solutions depends on their quality or fitness. Thus better solutions have more chances of evolution. The GAs create several generations of solutions and at the end the best member of a population will be considered the optimal solution.

This paper will study the suitability of GAs for the resolution of these problems for both real time (virtual reality) and non-real time applications. Several strategies are studied along with their impact upon an acceptable solution and in an acceptable processing time. The satisfactory solution for each frame would be the one in which the humanoid model is situated matching the detected body-features, within biomechanical limits and based on previous frames so that the motion is sufficiently smooth.

Vision-Based Comparative Study of Analytic and Numeric Inverse Kinematic Techniques for Recovering Arm Movements

Ronan Boulic¹, Javier Varona², Bruno Herbelin¹, Luis Unzueta³, Angel Suescun³
Antoni Jaume², Francisco Perales², Daniel Thalmann¹

¹ VRLAB, EPFL, {ronan.boulic | bruno.herbelin | daniel.thalmann}@epfl.ch

² UIB, {vdmijvg4 | paco.perales}@uib.es

³ CEIT, {lunzueta | asuescun}@ceit.es

Abstract:

The present paper compares the advantages and weaknesses of analytic and numeric Inverse Kinematics (noted IK) for computing both arms movements in a non-invasive vision-driven context. The experimental setting is the following: the performer first stands still in front of two cameras in a calibration posture for both color calibration of the hand, and skeleton estimation. Then the performer is allowed to move freely both arms while his hands' positions are tracked in real-time. Either the hand center or an estimation of the wrist position of the wrist joint center is provided to both IK techniques. Such a context is redundant as the dimension of the provided 3D position is smaller than the four degrees of freedom that can be exploited in the shoulder and the elbow. We examine how each IK technique handles this under-constrained context in terms of believability of the resulting posture and in terms of performances.

1. Introduction

Two IK methods are compared with input provided by a color tracking vision system [VBP05]. The numeric IK method is described in [BB04] while the analytic IK method extends the work from [TGB00] to ensure temporal continuity.

2. Vision-based experimental protocol and results

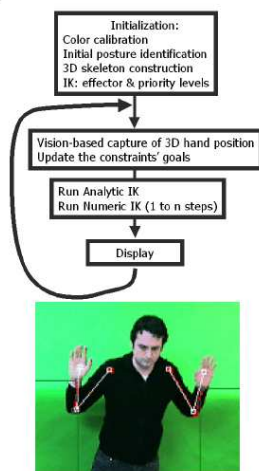


Fig. 1: the vision system provides the 3D hands position (stereo color tracking) to the 2 IK methods which in turn compute the arm posture

	Performance AMD Athlon 2800+	Continuity	Flexibility
Analytic IK	Stable 1μs 23 fps	Difficult to prevent some instability with noisy input	Arm + leg Case by case
Numeric IK 15 dof 3 priority levels 3x5 controlled dimensions	Depends on the nb of conv. steps (+latency): 1 : 0.8μs 24fps 5 : 2.6μs 22fps 20: 8.3μs 19fps	Latency Low pass filter Coherent solution with attraction towards initial positions	Generic Can be extended to spine and full body

3. Conclusion

Our current results show the maturity of numeric IK in real-time context. Its computing cost is equivalent to analytic IK while offering a greater potential in terms of stability and flexibility. Future work will extend the study to movements including the torso.

References

- [BB04] Baerlocher P., Boulic R., "An Inverse Kinematic Architecture Enforcing an Arbitrary Number of Strict Priority Levels", The Visual Computer, Springer Verlag, 20(6), 2004, 402-417.
- [TGB00] Tolani D., Goswami A., Badler N. "Real-time inverse kinematics techniques for anthropomorphic limbs". Graphical Models and Image Processing, Elsevier, 62-5 (Sept. 2000), 353-388.
- [VBP05] Varona J., Buades J., Perales f.J., "Hands and face tracking for VR applications", Computers and Graphics, Elsevier, 29-2 (April 2005), 179-187.

INDEX

A

Activity interpretation, 3
Analysis, 15
Analysis-by-synthesis, 28

B

Background subtraction, 19, 49
Backprojection, 44
Biomechanical limits, 28, 88
Blob, 19
Block matching (BM), 66

C

Camera calibration, 16
 Extrinsic, 16
 Intrinsic, 16
Camera parameters, 16
CamShift, 22, 42
Chroma-key, 49
Chrominance, 43
Collision avoidance, 28, 90
Color space, 5, 43, 60
Colored features optical flow (CFOF), 42,
 127, 135
Colored maximum distance filter (CMDf),
 45
Computer vision, 5
Condensation, 23, 66
Contour-figure, 25
Control, 15
Cyclic coordinate descent (CCD), 30, 75, 94

D

Damped least squares (DLS), 30, 75, 94
Depth of the pixel, 5
Depth warping, 123

DLS with single value decomposition (SVD),
 30, 75, 94
Dynamic Bayesian networks (DBN), 37
Dynamic time warping (DTW), 36

E

Edges, 19
End-effectors, 30, 41

F

Feature points, 19
Flexion-extension, 87
Flock of features, 42

G

Gesture spotting, 33, 38, 117
 Dynamic gestures, 116, 119, 127
 Quasi-static gestures, 116, 119
 Segmentation ambiguity, 38
 Spatio-temporal (ST) variability, 38

H

H-Anim, 73, 97
Hidden Markov models (HMM), 36
HLS, 43, 60
Horn-Schunck (HS), 66
HSV, 43, 60
Human-computer interaction (HCI), 2

I

Image-feature, 17
Inverse kinematics (IK), 7, 74
 Analytical, 74
 Exactly-guided, 7
 Hybrid, 74
 Numerical, 74
 Over-guided, 7

Under-guided, 7

J

Jacobian transpose, 30, 75, 94

K

Kalman filter (KF), 22, 43
 Kanade-Lucas-Tomasi (KLT), 22, 42
 Kinetic pseudo-energy, 117
 K-nearest neighbors (K-NN), 36, 119, 131
 Kulpa-Multon-Arnaldi (KMA), 76, 94

L

Lab, 43
 Leave one out (LOO), 131
 Lucas-Kanade (LK), 66
 Luminance, 43
 Luv, 43, 60

M

Maximum distance filter (MDF), 66
 Mocap system, 3
 Marker-based, 3, 107
 Markerless, 5, 41
 Motion capture, 3
 Motion reconstruction, 17
 Robustness, 15
 Motion-feature, 32
 Motor action recognition, 3, 7, 32, 115
 Combined actions, 38, 126, 133
 Holistic, 32
 Pose model, 121
 State-space, 34
 Template, 33
 Multibody, 6
 Multi-camera system, 16

N

Neural networks, 37
 Normalized-RGB (nRGB), 43, 60

O

Occlusion, 23, 53
 Online motion retargeting (OMR), 75
 Optical flow, 21

P

Particle filtering, 23, 66
 Penetration depth (PD), 90
 Pfinder, 66
 Pinhole camera model, 16
 Pose reconstruction, 3, 6, 24, 73
 Bottom-up, 25, 26
 Top-down, 25, 28
 Principal component analysis (PCA), 33, 133
 Principal markers, 41, 115, 147
 Prioritized inverse kinematics (PIK), 30, 75, 94
 Pronation-supination, 87
 Pseudoinverse, 30, 75, 94
 Pyramidal Lucas-Kanade (PyrLK), 21, 42, 66

R

Radial-ulnar deviations, 87
 Random forest, 115, 131
 Region of interest (ROI), 43
 Relevance vector machines (RVM), 37
 RGB, 5

S

Selectively damped least squares (SDLS), 30, 94
 Sequential inverse kinematics (SIK), 74, 136
 Shadows, 50
 Silhouette, 19
 Mask, 50
 Statistical classification, 8
 Stereo camera, 5
 Stick-figure, 24
 Supervised learning, 35
 Support vector machines (SVM), 37, 131
 Surveillance, 15
 Swing, 88

T

Time warping, 116
 Time-of-Flight (ToF) camera, 5
 Tolani-Goswami-Badler (TGB), 76, 94
 T-pose, 29
 Tracking, 17, 20, 24
 Kernel, 22
 Point, 21
 Silhouette, 23

Triangulation, 124
Twist, 81, 88

V

Visual hull, 20
Volumetric-figure, 25

W

White balance, 17

X

xyY, 43, 60

Y

YCrCb, 43, 60
YUV, 66

REFERENCES

- Agarwal, A., and Triggs, B. (2006). "Recovering 3D Human Pose from Monocular Images." *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44-58.
- Aggarwal, J. K., and Cai, Q. (1999). "Human Motion Analysis: A Review." *Computer Vision and Image Understanding*, 73(3), 428-440.
- Ahmad, M., and Lee, S.-W. (2006). "Human Action Recognition Using Multi-View Image Sequences Features." *Proceedings of the International Conference on Automatic Face and Gesture Recognition*, Southampton, UK, 523-528.
- Allmen, M. C., and Dyer, C. R. (1990). "Cyclic Motion Detection Using Spatiotemporal Surfaces and Curves." *Proceedings of the International Conference on Pattern Recognition*, Atlantic City, NJ, USA, 365-370.
- Analog-Devices. (2007). "iMEMS Accelometers: iMEMS ADXL330." <http://www.analog.com/>.
- Andrews, R. J., and Lovell, B. C. (2003). "Color Optical Flow." *Proceedings of the Workshop on Digital Image Computing*, Brisbane, Australia, 135-139.
- Apple. (2007). "iPhone Mobile Phone." <http://www.apple.com/iphone/>.
- Ardizzone, E., Chella, A., and Pirrone, R. (2000). "Pose Classification Using Support Vector Machines." *Proceedings of the IEEE-INNS-ENNS International Joint Conference on Neural Networks*, Como, Italy, 6, 317-322.
- Argyros, A. A., and Lourakis, M. I. A. (2006). "Vision-Based Interpretation of Hand Gestures for Remote Control of a Computer Mouse." *Proceedings of the HCI Workshop (in Conjunction with ECCV)*, LNCS 3979, Graz, Austria, 40-51.
- Ascension. (2004). "MotionStar Wireless 2 Magnetic Motion Capture System." <http://www.ascension-tech.com/products/motionstarwireless.php>.

- Ausejo, S. (2006). "A New Robust Motion Reconstruction Method Based on Optimisation with Redundant Constraints and Natural Coordinates," PhD Thesis, Tecnun, University of Navarra, Donostia-San Sebastián, Spain.
- Avidan, S. (2001). "Support Vector Tracking." *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Kauai Marriott, Hawaii, USA, 1, 184-191.
- Badler, N. I., Hollick, M. J., and Granieri, J. P. (1993a). "Real-time control of a virtual human using minimal sensors." *Presence: Teleoperators and Virtual Environments*, 2(1), 82-86.
- Badler, N. I., Phillips, C. B., and Webber, B. L. (1993b). *Simulating Humans: Computer Graphics, Animation, and Control*, Oxford University Press.
- Baerlocher, P., and Boulic, R. (2000). "Parametrization and Range of Motion of the Ball-and-Socket Joint." *Deformable Avatars*, 180-190.
- Baerlocher, P., and Boulic, R. (2004). "An Inverse Kinematic Architecture Enforcing an Arbitrary Number of Strict Priority Levels." *The Visual Computer: International Journal of Computer Graphics*, 20(6), 402-417.
- Baker, S., Scharstein, D., Lewis, J. P., Roth, S., Black, M., and Szeliski, R. (2007). "A Database and Evaluation Methodology for Optical Flow." *Proceedings of the IEEE International Conference on Computer Vision*, Rio de Janeiro, Brazil.
- Balestrino, A., De Maria, G., and Sciavicco, L. (1984). "Robust Control of Robotic Manipulators." *Proceedings of the International Federation of Automatic Control World Congress*, Budapest, Hungary, 5, 2435-2440.
- Bar-Shalom, Y., and Foreman, T. (1988). *Tracking and Data Association*, Academic Press Inc.
- Barrón, C., and Kakadiaris, I. A. (2003). "On the Improvement of Anthropometry and Pose Estimation from a Single Uncalibrated Image." *Machine Vision and Applications*, 13, 229-236.

- Basu, M. (2002). "Gaussian-Based Edge-Detection Methods-A Survey." *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews*, 32(3).
- Baudel, T., and Beaudouin-Lafon, M. (1993). "CHARADE: Remote Control of Objects Using Free-Hand Gestures." *Communications of the ACM*, 36(7), 28-35.
- Bellman, R. (1957). *Dynamic Programming*, Princeton University Press.
- Ben-Arie, J., Wang, Z., Pandit, P., and Rajaram, S. (2002). "Human Activity Recognition Using Multidimensional Indexing." *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(8), 1091-1104.
- BenAbdelkader, C., and Davis, L. S. (2006). "Estimation of Anthropomeasures from a Single Calibrated Camera." *Proceedings of the International Conference on Automatic Face and Gesture Recognition*, Southampton, UK, 499-504.
- Bertalmio, M., Sapiro, G., and Randall, G. (2000). "Morphing Active Contours." *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(7), 733-737.
- Birchfield, S., and Tomasi, C. (1999). "Depth Discontinuities by Pixel-to-Pixel Stereo." *International Journal of Computer Vision*, 35(3), 269-293.
- Black, M., and Jepson, A. (1998). "Eigentracking: Robust Matching and Tracking of Articulated Objects Using a View-Based Representation." *International Journal of Computer Vision*, 26(1), 63-84.
- Bobick, A. F., and Davis, J. W. (2001). "The Recognition of Human Movement Using Temporal Templates." *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(3), 257-267.
- Bouguet, J.-Y. (1999). "Pyramidal Implementation of the Lucas Kanade Feature Tracker." Intel Corporation, Microprocessor Research Labs. OpenCV Documents.
- Boulic, R., Ulicny, B., and Thalmann, D. (2004). "Versatile Walk Engine." *Journal of Game Development*, 1(1), 29-52.

- Bradski, G. R. (1998). "Real-Time Face and Object Tracking as a Component of a Perceptual User Interface." *Proceedings of the IEEE Workshop on Applications of Computer Vision*, Washington, DC, USA, 214-219.
- Brand, M. (1999). "Shadow Puppetry." *Proceedings of the International Conference on Computer Vision*, Corfu, Greece, 2, 1237.
- Brand, M., Oliver, N., and Pentland, A. (1997). "Coupled Hidden Markov Models for Complex Action Recognition." *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, 994-999.
- Bregler, C. (1997). "Learning and Recognizing Human Dynamics in Video Sequences." *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, San Juan, Puerto Rico, 568-574.
- Bregler, C., Malik, J., and Pullen, K. (2004). "Twist Based Acquisition and Tracking of Animal and Human Kinematics." *International Journal of Computer Vision*, 56(3), 179-194.
- Breiman, L. (2001). "Random Forests." *Machine Learning*, 45(1), 5-32.
- Broida, T., and Chellappa, R. (1986). "Estimation of Object Motion Parameters from Noisy Images." *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 8(1), 90-99.
- Buss, S. R., and Kim, J.-S. (2005). "Selectively Damped Least Squares for Inverse Kinematics." *Journal of Graphics tools*, 10(3), 37-49.
- Carranza, J., Theobalt, C., Magnor, M. A., and Seidel, H.-P. (2003). "Free-Viewpoint Video of Human Actors." *ACM Transactions on Graphics. Proceedings of the ACM Siggraph*, San Diego, USA, 22(3), 569-577.
- Cedras, C., and Shah, M. (1995). "Motion-Based Recognition: A Survey." *Image and Vision Computing*, 13(2), 129-155.
- CMU. (2008). "Carnegie Mellon University Graphics Lab Motion Capture Database." NSF EIA-0196217, <http://mocap.cs.cmu.edu>.
- Comaniciu, D., Ramesh, V., and Meer, P. (2003). "Kernel-Based Object Tracking." *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25, 564-575.

- Corazza, S., Mündermann, L., Chaudhari, A. M., Demattio, T., Cobelli, C., and Andriacchi, T. P. (2006). "A Markerless Motion Capture System to Study Musculoskeletal Biomechanics: Visual Hull and Simulated Annealing Approach." *Annals of Biomedical Engineering*, 34(6), 1019-1029.
- Cutting, J. E., and Proffitt, D. R. (1982). "The Minimum Principle and the Perception of Absolute, Common, and Relative Motions." *Cognitive Psychology*, 14, 211-246.
- Chai, J., and Hodgins, J. K. (2005). "Performance Animation from Low-Dimensional Control Signals." *ACM Transactions on Graphics. Proceedings of ACM Siggraph*, Los Angeles, CA, USA, 24(3), 686-696.
- Cham, T.-J., and Rehg, J. M. (1999). "A Multiple Hypothesis Approach to Figure Tracking." *Proceedings of the Conference on Computer Vision and Pattern Recognition*, Ft. Collins, CO, USA, 239-245.
- Cheung, G. K. M., Baker, S., and Kanade, T. (2003). "Shape-From-Silhouette of Articulated Objects and Its Use for Human Body Kinematics Estimation and Motion Capture." *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Madison, WI, USA, 1, 1-77-1-84.
- Cheung, K.-M. G., Baker, S., and Kanade, T. (2005). "Shape-From-Silhouette Across Time Part I: Theory and Algorithms." *International Journal of Computer Vision*, 62(3), 221-247.
- Choi, K.-J., and Ko, H.-S. (1999). "On-line Motion Retargetting." *Proceedings of the International Pacific Graphics*, Seoul, Korea, 32-42.
- Choo, K., and Fleet, D. J. (2001). "People Tracking Using Hybrid Monte Carlo Filtering." *Proceedings of the IEEE International Conference on Computer Vision*, Vancouver, BC, Canada, 2, 321-328.
- D'Souza, A., Vijayakumar, S., and Schaal, S. (2001). "Learning Inverse Kinematics." *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, Maui, Hawaii, USA, 1, 298-303.
- Darrell, T. J., and Pentland, A. P. (1993). "Space-Time Gestures." *Proceedings of the Conference on Computer Vision and Pattern Recognition*, New York, NY, USA, 335-340.

- Dasarathy, B. V. (1991). *Nearest Neighbor (NN) Norms: NN Pattern Recognition Classification Techniques*, IEEE Computer Society Press.
- Date, N., Yoshimoto, H., Arita, D., and Taniguchi, R.-i. (2004). "Real-Time Human Motion Sensing Based on Vision-Based Inverse Kinematics for Interactive Applications." *Proceedings of the International Conference on Pattern Recognition*, Cambridge, UK, 3, 318-321.
- Delamarre, Q., and Faugeras, O. (2001). "3D Articulated Models and Multiview Tracking with Physical Forces." *Computer Vision and Image Understanding*, 328-357.
- Delaunay, B. (1934). "Sur la Sphère Vide. A la Mémoire de Georges Voronoi." *Bulletin of Academy of Sciences of the USSR*, 7, 793-800.
- Deutscher, J., and Reid, I. (2005). "Articulated Body Motion Capture by Stochastic Search." *International Journal of Computer Vision*, 61(2), 185-205.
- DoMotion. (2004). "DoMotion Mechanical Motion Capture System."
<http://www.domotion.co.kr/>.
- Efros, A., Berg, A., Mori, G., and Malik, J. (2003). "Recognizing Action at a Distance." *Proceedings of the IEEE International Conference on Computer Vision*, Nice, France, 2, 726-733.
- Elgammal, A., and Lee, C. S. (2004). "Inferring 3D Body Pose from Silhouettes Using Activity Manifold Learning." *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Washington, DC, USA, 2, 681-688.
- Elgammal, A. M., and Davis, L. S. (2001). "Probabilistic Framework for Segmenting People Under Occlusion." *Proceedings of the IEEE International Conference on Computer Vision*, Vancouver, BC, Canada, 2, 145-152.
- Emering, L., Boulic, R., and Thalmann, D. (1998). "Interacting with Virtual Humans Through Body Actions." *IEEE Journal of Computer Graphics and Application*, "Projects in VR", 8-11.

- Emering, L., Boulic, R., and Thalmann, D. (1999). "Conferring Human Action Recognition Skills to Life-Like Agents." *Journal of Applied Artificial Intelligence, Special Issue on Animated Interface Agents*, 13(4-5), 539-565.
- ENACTIVE. (2007). "The Alter Body Experience." *Deliverable D.EES2.2*, European Network of Excellence IST-002114-ENACTIVE Interfaces.
- Engin, A. E., and Chen, S. M. (1986). "Statistical Data Base for the Biomechanical Properties of the Human Shoulder Complex I: Kinematics of the Shoulder Complex." *Journal of Biomechanical Engineering*, 108, 215-221.
- Farrugia, J.-P., Horain, P., Guehenneux, E., and Allusse, Y. (2006). "GPUCV: A Framework for Image Processing Acceleration with Graphics Processors." *Proceedings of the IEEE International Conference on Multimedia and Expo*, Toronto, ON, Canada, 585-588, <http://picoforge.int-evry.fr/cgi-bin/twiki/view/Gpucv/Web/>.
- FBI. (1908). "Federal Bureau of Investigation of the United States Department of Justice." <http://www.fbi.gov/>.
- Fernández-García, N. L., Carmona-Poyato, A., Medina-Carnicer, R., and Madrid-Cuevas, F. J. (2008). "Automatic Generation of Consensus Ground Truth for the Comparison of Edge Detection Techniques." *Image and Vision Computing*, 26(4), 496-511.
- Freeman, W. T., Tanaka, K., Ohta, J., and Kyuma, K. (1996). "Computer Vision for Computer Games." *Proceedings of the International Conference on Automatic Face and Gesture Recognition*, Killington, VT, USA, 100-105.
- Fukunaga, K. (1990). *Introduction to Statistical Pattern Recognition*, Academic Press Professional Inc.
- Fung, J., Mann, S., and Aimone, C. (2005). "OpenVIDIA: Parallel GPU Computer Vision." *Proceedings of the ACM Multimedia*, Hilton, Singapore, 849-852, <http://sourceforge.net/projects/openvidia>.
- Gabriel, P. F., Verly, J. G., Piater, J. H., and Genon, A. (2003). "The State of the Art in Multiple Object Tracking Under Occlusion in Video Sequences." *Advanced Concepts for Intelligent Vision Systems*, 166-173.

- Galata, A., Johnson, N., and Hogg, D. (2001). "Learning Variable-Length Markov Models of Behavior." *Computer Vision and Image Understanding*, 81(3), 398-413.
- Gan, J. Q., Oyama, E., Rosales, E. M., and Hu, H. (2005). "A Complete Analytical Solution to the Inverse Kinematics of the Pioneer 2 Robotic Arm." *Robotica*, 23(1), 123-129.
- Gavrila, D. M. (1999). "The Visual Analysis of Human Movement: A Survey." *Computer Vision and Image Understanding*, 73(1), 82-98.
- Gavrila, D. M., and Davis, L. S. (1996). "Tracking of Humans in Action: A 3D Model-Based Approach." *Proceedings of the Conference on Computer Vision and Pattern Recognition*, San Francisco, CA, USA, 73-80.
- González, J. (2004). "Human Sequence Evaluation: the Key-Frame Approach," PhD Thesis, University of Barcelona, Bellaterra, Spain.
- Grassia, F. S. (1998). "Practical Parameterization of Rotations Using the Exponential Map." *The Journal of Graphics Tools*, 3(3), 29-48.
- Grauman, K., Shakhnarovich, G., and Darrell, T. (2003). "Inferring 3D Structure with a Statistical Image-Based Shape Model." *Proceedings of the International Conference on Computer Vision*, Nice, France, 1, 641-647.
- Grochow, K., Martin, S. L., Hertzmann, A., and Popovic, Z. (2004). "Style-Based Inverse Kinematics." *ACM Transactions on Graphics. Proceedings of Siggraph*, Los Angeles, CA, USA, 522-531.
- Guo, F., and Qian, G. (2006). "Dance Posture Recognition Using Wide-Baseline Orthogonal Stereo Cameras." *Proceedings of the International Conference on Automatic Face and Gesture Recognition*, Tempe, AZ, USA, 481-486.
- Guo, Y., Xu, G., and Tsuji, S. (1994). "Understanding Human Motion Patterns." *Proceedings of the International Conference on Pattern Recognition*, Jerusalem, Israel, 2, 325-329.
- Gyaourova, A., Kamath, C., and Cheung, S.-C. (2003). "Block Matching for Object Tracking." *UCRL-TR-200271*, Lawrence Livermore National Laboratory, Livermore, CA, USA.

- H-Anim. (2008). "H-Anim Specification." Humanoid Animation Working Group, <http://www.hanim.org/>.
- Haritaoglu, I., Harwood, D., and Davis, L. S. (2000). "W4: Real-Time Surveillance of People and Their Activities." *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(8), 809-830.
- Hartley, R., and Zisserman, A. (2000). *Multiple View Geometry in Computer Vision*, Cambridge University Press.
- Heidelberger, B., Desmecht, L., Hare, J., Austin, C., Dachary, L., and Lamoureux, P. (2006). "Cal3D Open Source 3D Character Animation Library." <https://gna.org/projects/cal3d/>.
- Herda, L., Urtasun, R., Fua, P., and Hanson, A. (2003). "Automatic Determination of Shoulder Joint Limits using Quaternion Field Boundaries." *International Journal of Robotics Research*, 22(6), 419-438.
- Horn, B. K. P., and Schunck, B. G. (1981). "Determining Optical Flow." *Artificial Intelligence*, 17, 185-203.
- HUMODAN. (2005). "An Automatic Human Model Animation Environment for Augmented Reality Interaction." *IST-2001-32202 European Project*.
- Huttenlocher, D., Noh, J., and Rucklidge, W. (1993). "Tracking Nonrigid Objects in Complex Scenes." *Proceedings of the IEEE International Conference on Computer Vision*, Berlin, Germany, 93-101.
- Ibarguren, A., Maurtua, I., and Sierra, B. (2007). "Recognition of Sign Language in Real Time Through Data Gloves." *Proceedings of the CAEPLA Conference (in Conjunction with the TFLA Workshop)*, Salamanca, Spain.
- Intel. (2001). "OpenCV: Open Source Computer Vision Library." <http://sourceproject.net/project/opencvlibrary>.
- Ioffe, S., and Forsyth, D. (1999). "Finding People by Sampling." *Proceedings of the International Conference on Computer Vision*, Corfu, Greece, 2, 1092-1097.
- Isard, M., and Blake, A. (1998). "Condensation-Conditional Density Propagation for Visual Tracking." *International Journal of Computer Vision*, 29(1), 5-28.

- Jain, A. K., Duin, R. P. W., and Mao, J. (2000). "Statistical Pattern Recognition: A Review." *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(1), 4-37.
- Johansson, G. (1973). "Visual Perception of Biological Motion and a Model for Its Analysis." *Perception and Psychophysics*, 14(2), 210-211.
- Johansson, G. (1975). "Visual Motion Perception." *Scientific American*, 76-88.
- Joic, N., Brumitt, B., Meyers, B., Harris, S., and Huang, T. (2000). "Detection and Estimation of Pointing Gestures in Dense Disparity Maps." *International Conference on Automatic Face and Gesture Recognition*, Grenoble, France, 468-475.
- Julier, S. J., and Uhlmann, J. K. (1997). "A New Extension of the Kalman Filter to Nonlinear Systems." *Proceedings of AeroSense: Symposium on Aerospace/Defense Sensing, Simulation and Controls, Multi Sensor Fusion, Tracking and Resource Management II*, Orlando, FL, USA, 3068, 182-193.
- Kakadiaris, I., and Metaxas, D. (2000). "Model-Based Estimation of 3D Human Motion." *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(12), 81-87.
- Kakadiaris, I. A., and Metaxas, D. (1995). "3D Human Body Model Acquisition from Multiple Views." *Proceedings of the International Conference on Computer Vision*, Cambridge, MA, USA, 618-623.
- Kalman, R. E. (1960). "A New Approach to Linear Filtering and Prediction Problems." *Transactions of the ASME-Journal of Basic Engineering*, 83, 95-107.
- Kallmann, M. (2005). "Scalable Solutions for Interactive Virtual Humans that can Manipulate Objects." *Proceedings of the Artificial Intelligence and Interactive Digital Entertainment*, Marina del Rey, CA, USA, 69-74.
- Kang, H., Lee, C. W., and Jung, K. (2004a). "Recognition-Based Gesture Spotting in Video Games." *Pattern Recognition Letters*, 25(15), 1701-1714.

- Kang, J., Cohen, I., and Medioni, G. (2004b). "Object Reacquisition Using Geometric Invariant Appearance Model." *Proceedings of the International Conference on Pattern Recognition*, Cambridge, UK, 4, 759-762.
- Kang, T., Tillery, S. H., and He, J. (2003). "Determining Natural Arm Configuration Along Reaching Trajectory." *Proceedings of the Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, Cancun, Mexico, 2, 1444-1447.
- Kapandji, I. A. (1974). *Physiology of the Joints Vol. 3: The Trunk and the Vertebral Column*, Churchill Livingstone.
- Kapandji, I. A. (1982). *Physiology of the Joints Vol. 1: The Upper Limb*, Churchill Livingstone.
- Kapandji, I. A. (1988). *Physiology of the Joints Vol. 2: The Lower Limb*, Churchill Livingstone.
- Kehl, R., Bray, M., and Van Gool, L. (2005). "Full Body Tracking from Multiple Views Using Stochastic Sampling." *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, San Diego, CA, USA, 2, 129-136.
- Khan, S., and Shah, M. (2000). "Tracking People in Presence of Occlusion." *Proceedings of the Asian Conference on Computer Vision*, Taipei, Taiwan, China.
- Koffka, K. (1935). *Principle of Gestalt Psychology*, Harcourt Brace.
- Kölsch, M., and Turk, M. (2004). "Fast 2D Hand Tracking with Flock of Features and Multi-Cue Integration." *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshop*, Washington, DC, USA, 158.
- Korein, J. U. (1985). *A Geometric Investigation of Reach*, The MIT Press.
- Kovar, L., Gleicher, M., and Pighin, F. (2002). "Motion Graphs." *ACM Transactions on Graphics. Proceedings of the ACM Siggraph*, San Antonio, TX, USA, 21(3), 473-482.

- Krahnstoever, N., Yeasin, M., and Sharma, R. (2003). "Automatic Acquisition and Initialization of Articulated Models." *Machine Vision and Applications*, 14, 218-228.
- Kulpa, R., Multon, F., and Arnaldi, B. (2005). "Morphology-Independent Representation of Motions for Interactive Human-Like Animation." *Computer Graphics Forum. Proceedings of the Eurographics*, Dublin, Ireland, 4(3), 343-351.
- Laurentini, A. (1994). "The Visual Hull Concept for Silhouette-Based Image Understanding." *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 16(2), 150-162.
- Lee, H.-K., and Kim, J. H. (1999). "An HMM-Based Threshold Model Approach for Gesture Recognition." *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 21(10), 961-973.
- Lin, C.-T., Nein, H.-W., and Lin, W.-C. (1999). "A Space-Time Delay Neural Network for Motion Recognition and Its Application to Lipreading." *International Journal of Neural Systems*, 9(4), 311-334.
- Liu, G., Zhang, J., Wang, W., and McMillan, L. (2006). "Human Motion Estimation from a Reduced Marker Set." *Proceedings of the Symposium on Interactive 3D Graphics and Games*, 35-42.
- Lucas, B. D., and Kanade, T. (1981). "An Iterative Image Registration Technique with an Application to Stereo Vision." *Proceedings of the DARPA Image Understanding Workshop*, Washington, DC, USA, 121-130.
- Luo, Y., Wu, T.-W., and Hwang, J.-N. (2003). "Object-Based Analysis and Interpretation of Human Motion in Sports Video Sequences by Dynamic Bayesian Networks." *Computer Vision and Image Understanding*, 92, 196-216.
- Maciejewski, A. A. (1990). "Dealing with the ill-Conditioned Equations of Motion for Articulated Figures." *IEEE Computer Graphics and Applications*, 10(3), 63-71.

- Maciejewski, A. A., and Klein, C. A. (1988). "Numerical Filtering for the Operation of Robotic Manipulators Through Kinematically Singular Configurations." *Journal of Robotic Systems*, 5(6), 527-552.
- Marey, E. J. (1873). *Animal Mechanism: A Treatise on Terrestrial and Aerial Locomotion*, Appleton. Republished as Vol. XI of the International Scientific Series.
- Masoud, O., and Papanikolopoulos, N. (2003). "A Method for Human Action Recognition." *Image and Vision Computing*, 21(8), 729-743.
- May, S., Werner, B., Surmann, H., and Pervolz, K. (2006). "3D Time-of-Flight Cameras for Mobile Robotics." *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, Beijing, China, 790-795.
- McKenna, S., Jabri, S., Duric, Z., and Wechsler, H. (2000). "Tracking Groups of People." *Computer Vision and Image Understanding*, 80(1), 42-56.
- Mikic, I., Trivedi, M., Hunter, E., and Cosman, P. (2003). "Human Body Model Acquisition and Tracking Using Voxel Data." *International Journal of Computer Vision*, 53(3), 199-223.
- MIRALab. (2008). "European Network of Excellence IST-002114-ENACTIVE Interfaces, EES2 Scenario Humanoid." University of Geneva, Geneva, Switzerland.
- Mitchel, T. (1997). *Machine Learning*, Mc Graw-Hill.
- Mitra, S., and Acharya, T. (2007). "Gesture Recognition: A Survey." *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews*, 37(3), 311-324.
- Moeslund, T. B., and Granum, E. (2001). "A Survey of Computer Vision-Based Human Motion Capture." *Computer Vision and Image Understanding*, 81(3), 231-268.
- Moeslund, T. B., Hilton, A., and Krüger, V. (2006). "A Survey of Advances in Vision-Based Human Motion Capture and Analysis." *Computer Vision and Image Understanding*, 104(2), 90-126.

- Moeslund, T. B., Vittrup, M., Pedersen, K. S., Laursen, M. K., Sørensen, M. K. D., Uhrenfeldt, H., and Granum, E. (2002). "Estimating the 3D Shoulder Position Using Monocular Vision." *International Conference on Imaging Science, Systems, and Technology*, Las Vegas, Nevada, 24-27.
- Molet, T., Boulic, R., and Thalmann, D. (1999). "Human Motion Capture Driven by Orientation Measurements." *Presence: Teleoperators & Virtual Environments*, 8(2), 187-203.
- Monheit, G., and Badler, N. I. (1991). "A Kinematic Model of the Human Spine and Torso." *IEEE Computer Graphics and Applications*, 11(2), 29-38.
- Moreels, P., and Perona, P. (2007). "Evaluation of Features Detectors and Descriptors Based on 3D Objects." *International Journal of Computer Vision*, 73(3), 263-284.
- Mündermann, L., Corazza, S., and Andriacchi, T. P. (2006). "The Evolution of Methods for the Capture of Human Movement Leading to Markerless Motion Capture for Biomechanical Applications." *Journal of Neuroengineering and Rehabilitation*, 3(6).
- Mulligan, J. (2005). "Upper Body Pose Estimation from Stereo and Hand-Face Tracking." *Canadian Conference on Computer and Robot Vision*, Victoria, BC, Canada, 413-420.
- Nakamura, Y., and Hanafusa, H. (1986). "Inverse Kinematics Solutions with Singularity Robustness for Robot Manipulator Control." *Journal of Dynamic Systems, Measurement, and Control*, 108, 163-171.
- Navaratnam, R., Thayananthan, A., Torr, P. H. S., and Cipolla, R. (2005). "Hierarchical Part-Based Human Body Pose Estimation." *Proceedings of the British Machine Vision Conference*, London, UK, 1, 479-488.
- Nintendo. (2006). "Wii Console." <http://www.wii.com/>.
- O'Rourke, J., and Badler, N. I. (1980). "Model-Based Image Analysis of Human Motion using Constraint Propagation." *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2(6), 522-536.

- Okada, R., Stenger, B., Ike, T., and Kondoh, N. (2006). "Virtual Fashion Show Using Real-time Markerless Motion Capture." *Proceedings of the Asian Conference on Computer Vision, LNCS 3852*, Hyderabad, India, 801-810.
- Organic-Motion. (2008). "Stage, Biostage and Openstage Markerless Motion Capture Systems." <http://www.organicmotion.com/>.
- Otani, E. (1989). "Software Tools for Dynamic and Kinematic Modeling of Human Motion." *Technical Report MS-CIS-89-43, MSE Thesis*, University of Pennsylvania, Philadelphia, PA, USA.
- Oyama, E., Chong, N. Y., Agah, A., Maeda, T., and Tachi, S. (2001). "Inverse Kinematics Learning by Modular Architecture Neural Networks with Performance Prediction Networks." *Proceedings of the IEEE International Conference on Robotics & Automation*, Seoul, Korea, 1, 1006-1012.
- Pantrigo, J. J., Montemayor, A. S., and Cabido, R. (2005). "Scatter Search Particle Filter for 2D Real-Time Hands and Face Tracking." *Proceedings of the International Conference on Image Analysis and Processing, LNCS 3617*, Cagliari, Italy, 953-960.
- Parameswaran, V., and Chellappa, R. (2004). "View Independent Human Body Pose Estimation from a Single Perspective Image." *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, Washington, DC, USA, 2, 16-22.
- Park, S., and Aggarwal, J. K. (2004). "Semantic-Level Understanding of Human Actions and Interactions Using Event Hierarchy." *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshop*, Washington, DC, USA, 1, 12.
- Pascale, D. (2003). "A Review of RGB Color Spaces." The BabelColor Company.
- Peinado, M., Herbelin, B., Wanderley, M., Le Callennec, B., Boulic, R., Thalmann, D., and Méziat, D. (2004). "Towards Configurable Motion Capture with Prioritized Inverse Kinematics." *Proceedings of the International Workshop on Virtual Rehabilitation*, Lausanne, Switzerland, 85-96.

- Peinado, M., Meziat, D., Boulic, R., and Raunhardt, D. (2006). "Environment-Aware Postural Control of Virtual Humans for Real-time Applications." *Proceedings of the SAE Conference on Digital Human Modeling for Design and Engineering*, Lyon, France, 2006-01-2341.
- Peinado, M., Meziat, D., Maupu, D., Raunhardt, D., Thalmann, D., and Boulic, R. (2007). "Accurate On-Line Avatar Control with Collision Anticipation." *Proceedings of the ACM Symposium on Virtual Reality Software and Technology*, Newport Beach, CA, USA, 89-97.
- Phasespace. (2005). "IMPULSE Optical Motion Capture System."
<http://www.phasespace.com>.
- Pheasant, S. (1986). *Bodyspace: Anthropometry, Ergonomics, and Design*, Taylor & Francis.
- Piccardi, M. (2004). "Background Subtraction Techniques: A Review." *Proceedings of the IEEE International Conference on Systems, Man and Cybernetics*, The Hague, Netherlands, 4, 3099-3104.
- Plänkers, R., and Fua, P. (2001). "Articulated Soft Objects for Video-Based Body Modeling." *Proceedings of the IEEE International Conference on Computer Vision*, Vancouver, BC, Canada, 394-401.
- PointGrey. (2007). "Bumblebee 2 Stereo Vision Camera."
<http://www.ptgrey.com/products/stereo.asp>.
- Polana, R., and Nelson, R. (1994). "Low Level Recognition of Human Motion (or How to Get Your Man without Finding His Body Parts)." *Proceedings of IEEE Workshop on Motion of Non-Rigid and Articulated Objects*, Austin, TX, USA, 77-82.
- Poppe, R. (2007). "Vision-Based Human Motion Analysis: An Overview." *Computer Vision and Image Understanding*, 108, 4-18.
- Prati, A., Mikic, I., Trivedi, M. M., and Cucchiara, R. (2003). "Detecting Moving Shadows: Algorithms and Evaluation." *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(7), 918-923.
- Raducanu, B., and Vitrià, Y. J. (2006). "A Robust Particle Filter-Based Face Tracker Using Combination of Color and Geometric Information."

- Proceedings of the International Conference on Image Analysis and Recognition, LNCS 4141, Póvoa de Varzim, Portugal, 922-933.*
- Rahman, M. M., and Robles-Kelly, A. (2006). "A Tuned Eigenspace Technique for Articulated Motion Recognition." *Proceedings of the European Conference on Computer Vision, LNCS 3954, Graz, Austria, 174-185.*
- Ramanan, D., and Forsyth, D. A. (2003). "Automatic Annotation of Everyday Movements." *Proceedings of the Neural Information Processing Systems Conference, Vancouver, BC, Canada.*
- Ramanan, D., and Sminchisescu, C. (2006). "Training Deformable Models for Localization." *IEEE Conference on Computer Vision and Pattern Recognition, New York, NY, USA, 1, 206-213.*
- Rangarajan, K., Allen, W., and Shah, M. A. (1993). "Matching Motion Trajectories Using Scale Space." *Pattern Recognition, 26(4), 595-609.*
- Raunhardt, D., and Boulic, R. (2007). "Exploiting Coupled Joints: Anatomic Control of the Spine with IK Through Linearly Coupled Joints." *Proceedings of the International Conference on Computer Graphics Theory and Applications, Barcelona, Spain, 13-20.*
- Ren, H., and Xu, G. (2002). "Human Action Recognition with Primitive-Based Coupled-HMM." *Proceedings of the International Conference on Pattern Recognition, Quebec, QC, Canada, 2, 494-498.*
- Ren, H., Xu, G., and Kee, S. (2004). "Subject-Independent Natural Action Recognition." *Proceedings of the International Conference on Automatic Face and Gesture Recognition, Seoul, Korea, 523-528.*
- Ren, X., Berg, A. C., and Malik, J. (2005). "Recovering Human Body Configurations Using Pairwise Constraints Between Parts." *Proceedings of the International Conference on Computer Vision, Beijing, China, 1, 824-831.*
- Ricquebourg, Y., and Bouthemy, P. (2000). "Real-Time Tracking of Moving Persons by Exploiting Spatio-Temporal Image Slices." *IEEE Transactions On Pattern Analysis and Machine Intelligence, 22(8), 797-808.*

- Rittscher, J., Blake, A., and Roberts, S. J. (2002). "Towards the Automatic Analysis of Complex Human Body Motions." *Image and Vision Computing*, 20, 905-916.
- Roh, M.-C., Christmas, B., Kittler, J., and Lee, S.-W. (2006). "Robust Player Gesture Spotting and Recognition in Low-Resolution Sports Video." *Proceedings of the European Conference on Computer Vision, LNCS 3954*, Graz, Austria, 347-358.
- Rohr, K. (1994). "Towards Model-Based Recognition of Human Movements in Image Sequences." *Computer Vision, Graphics, and Image Processing: Image Understanding*, 59(1), 94-115.
- Ronfard, R. (1994). "Region Based Strategies for Active Contour Models." *International Journal of Computer Vision*, 13(2), 229-251.
- Ronfard, R., Schmid, C., and Triggs, B. (2002). "Learning to Parse Pictures of People." *Proceedings of the European Conference on Computer Vision*, Copenhagen, Denmark, 700-714.
- Rose, R. C. (1992). "Discriminant Wordspotting Techniques for Rejection Non-Vocabulary Utterances in Unconstrained Speech." *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, San Francisco, CA, USA, 105-108.
- Rosenblum, M., Yacoob, Y., and Davis, L. (1994). "Human Emotion Recognition from Motion Using a Radial Basis Function Network Architecture." *Proceedings of the IEEE Workshop on Motion of Non-Rigid and Articulated Objects*, Austin, TX, USA, 43-49.
- Salari, V., and Sethi, I. K. (1990). "Feature Point Correspondence in the Presence of Occlusion." *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12(1), 87-91.
- Sánchez, J., and Borro, D. (2007). "Non Invasive 3D Tracking for Augmented Video Applications." *Proceedings of the IEEE Virtual Reality Conference*, Charlotte, NC, USA, 22-27.
- Santurde, J., Borro, D., and Matey, L. (2006). "Skin Detection for Hand Gesture Interfaces Using Assimilative Background." *Proceedings of the*

- Ibero American Symposium in Computer Graphics*, Santiago de Compostela, Spain, 195-202.
- Sato, K., and Aggarwal, J. (2004). "Temporal Spatio-Velocity Transform and Its Application to Tracking and Interaction." *Computer Vision and Image Understanding*, 96(2), 100-128.
- Schmid, C., Mohr, R., and Bauckhage, C. (2000). "Evaluation of Interest Point Detectors." *International Journal of Computer Vision*, 37(2), 151-172.
- Sedláček, M. (2004). "Evaluation of RGB and HSV Models in Human Faces Detection." *Proceedings of the Central European Seminar on Computer Graphics*, Budmerice, Slovakia, 125-131.
- Semwal, S. K., Hightower, R., and Stansfield, S. (1998). "Mapping Algorithms for Real-Time Control of an Avatar Using Eight Sensors." *Presence: Teleoperators & Virtual Environments*, 7(1), 1-21.
- SGI. (2008). "OpenGL: Open Graphics Library." <http://www.opengl.org/>.
- Shan, C., Wei, Y., Tan, T., and Ojardias, F. (2004). "Real Time Hand Tracking by Combining Particle Filtering and Mean Shift." *Proceedings of the IEEE International Conference on Automatic Face and Gesture Recognition*, Seoul, Korea, 669-674.
- Shannon, C. E., and Weaver, W. (1949). *The Mathematical Theory of Communication*, The University of Illinois Press.
- Shawe-Taylor, J., and Cristianini, N. (2000). *Support Vector Machines and Other Kernel-Based Learning Methods*, Cambridge University Press.
- Shi, J., and Tomasi, C. (1994). "Good Features to Track." *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Seattle, USA, 593-600.
- Shin, H. J., Lee, J., Shin, S. Y., and Gleicher, M. (2001). "Computer Puppetry: An Importance-Based Approach." *ACM Transactions on Graphics*, 20(2), 67-94.
- Sigal, L., and Black, M. J. (2006). "HumanEva: Synchronized Video and Motion Capture Dataset for Evaluation of Articulated Human

- Motion." *Technical Report CS-06-08*, Brown University, Providence, RI, USA.
- Slabaugh, G., Culbertson, B., and Malzbender, T. (2001). "A Survey of Methods for Volumetric Scene Reconstruction from Photographs." *Proceedings of the International Workshop on Volume Graphics*, Stony Brook, NY, USA, 81-100.
- Sminchisescu, C., Kanaujia, A., Li, Z., and Metaxas, D. (2005). "Discriminative Density Propagation for 3D Human Motion Estimation." *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, San Diego, CA, USA, 1, 390-397.
- Softkinetic. (2008). "Softkinetic. Building Natural Interfaces."
<http://www.softkinetic.net/>.
- Song, Y., Goncalves, L., and Perona, P. (2003). "Unsupervised Learning of Human Motion." *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(7), 814-827.
- Sony-Ericsson. (2007). "W910i Mobile Phone."
<http://www.sonyericsson.com/cws/products/mobilephones/overview/w910i/>.
- Sorenson, H. W. (1985). *Kalman Filtering: Theory and Application*, IEEE Press.
- Starck, J., and Hilton, A. (2003). "Model-Based Multiple View Reconstruction of People." *Proceedings of the International Conference on Computer Vision*, Nice, France, 2, 915-922.
- Stone, M. (1974). "Cross-Validation Choice and Assessment of Statistical Procedures." *Journal of Royal Statistical Society*, 36, 111-147.
- Streit, R. L., and Luginbuhl, T. E. (1994). "Maximum Likelihood Method for Probabilistic Multi-Hypothesis Tracking." *Proceedings of the International Society for Optical Engineering*, Bergen, Norway, 2235, 394-405.
- Sundaresan, A., and Chellappa, R. (2005). "Markerless Motion Capture Using Multiple Cameras." *Computer Vision for Interactive and Intelligent Environment*, 15-26.

- Svoboda, T., Martinec, D., and Pajdla, T. (2005). "A Convenient Multi-Camera Self-Calibration for Virtual Environments." *PRESENCE: Teleoperators and Virtual Environments*, 14(4), 407-422.
- Takahashi, K., Seki, S., Kojima, H., and Oka, R. (1994). "Recognition of Dexterous Manipulations from Time-Varying Images." *Proceedings of the IEEE Workshop on Motion of Non-Rigid and Articulated Objects*, Austin, TX, USA, 23-28.
- Takahashi, K., Seki, S., and Oka, R. (1992). "Spotting Recognition of Human Gestures from Motion Images." *Technical Report IE92-134*, The Institute of Electronics, Information and Communication Engineers, Japan, 9-16 (in Japanese).
- Tao, H., Sawhney, H., and Kumar, R. (2002). "Object Tracking with Bayesian Estimation of Dynamic Layer Representations." *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(1), 75-89.
- Tolani, D., Goswami, A., and Badler, N. I. (2000). "Real-Time Inverse Kinematics Techniques for Anthropomorphic Limbs." *Graphical Models*, 62, 353-388.
- Tsai, R. Y. (1986). "An Efficient and Accurate Camera Calibration Technique for 3D Machine Vision." *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, Miami Beach, FL, USA, 364-374.
- Uhlmann, J. K. (1992). "Algorithms for Multiple Target Tracking." *American Scientist*, 80(2), 128-141.
- Varona, J., Buades, J. M., and Perales, F. J. (2005). "Hands and Face Tracking for VR Applications." *International Journal of Systems & Applications in Computer Graphics*, 29, 179-187.
- Veenman, C. J., Reinders, M. J. T., and Backer, E. (2001). "Resolving Motion Correspondence for Densely Moving Points." *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(1), 54-72.
- Vrlab. (2008). "European Network of Excellence IST-002114-ENACTIVE Interfaces, EES2 Scenario Humanoid." EPFL, Lausanne, Switzerland.

- Wachter, S., and Nagel, H.-H. (1999). "Tracking Persons in Monocular Image Sequences." *Computer Vision and Image Understanding*, 74(3), 174-192.
- Wampler, C. W. (1986). "Manipulator Inverse Kinematic Solutions Based on Vector Formulations and Damped Least Squares Methods." *IEEE Transactions on Systems, Man, and Cybernetics*, 16, 93-101.
- Wang, L.-C. T., and Chen, C. C. (1991). "A Combined Optimization Method for Solving the Inverse Kinematics Problem of Mechanical Manipulators." *IEEE Transactions on Robotics and Automation*, 7(4), 489-499.
- Wang, L., Hu, W., and Tan, T. (2003). "Recent Developments in Human Motion Analysis." *Pattern Recognition*, 36(3), 585-601.
- Wang, X., Maurin, M., Mazet, F., De Castro Maia, N., Voinot, K., Verriest, J. P., and Fayet, M. (1998). "Three-Dimensional Modelling of the Motion Range of Axial Rotation of the Upper Arm." *Journal of Biomechanics*, 31, 899-908.
- Weinland, D., Ronfard, R., and Boyer, E. (2005). "Motion History Volumes for Free Viewpoint Action Recognition." *Proceedings of the Workshop on Modeling People and Human Interaction*, Beijing, China, 104, 249-257.
- Whitney, D. E. (1969). "Resolved Motion Rate Control of Manipulators and Human Prostheses." *IEEE Transactions on Man-Machine Systems*, 10, 47-53.
- Wolovich, W. A., and Elliot, H. (1984). "A Computational Technique for Inverse Kinematics." *Proceedings of the IEEE Conference on Decision and Control*, Las Vegas, NV, USA, 23, 1359-1363.
- Wren, C., Azarbayejani, A., Darrell, T., and Pentland, A. (1997). "Pfinder: Real-time Tracking of the Human Body." *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(7), 780-785.
- Wren, C. R. (2000). "Understanding Expressive Action," PhD Thesis, Massachusetts Institute of Technology, Cambridge, MA, USA.

- Wren, C. R., Clarkson, B. P., and Pentland, A. (2000). "Understanding Purposeful Human Motion." *Proceedings of the International Conference on Automatic Face and Gesture Recognition*, Grenoble, France, 378-383.
- Wu, X., Ma, L., Chen, Z., and Gao, Y. (2004). "A 12-DOF Analytic Inverse Kinematics Solver for Human Motion Control." *Journal of Information & Computational Science*, 1, 137-141.
- XSens. (2007). "Moven Inertial Motion Capture System."
<http://www.moven.com/>.
- Yamato, J., Ohya, J., and Ishii, K. (1992). "Recognizing Human Action in Time-Sequential Images Using Hidden Markov Models." *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Champaign, IL, USA, 379-385.
- Yang, H.-D., Park, A.-Y., and Lee, S.-W. (2006). "Human-Robot Interaction by Whole Body Gesture Spotting and Recognition." *Proceedings of the International Conference on Pattern Recognition*, Hong Kong, China, 4, 774-777.
- Yilmaz, A., Javed, O., and Shah, M. (2006). "Object Tracking: A survey." *ACM Computing Surveys*, 38(4), 1-45.
- Yilmaz, A., and Shah, M. (2005). "Actions Sketch: A Novel Action Representation." *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, San Diego, CA, USA, 1, 984-989.
- Yu, H., Sun, G.-m., Song, W.-x., and Li, X. (2005). "Human Motion Recognition Based on Neural Network." *Proceedings of the International Conference on Communications, Circuits and Systems*, Hong Kong, China, 2, 982.
- Zhang, L., Kim, Y. J., Varadhan, G., and Manocha, D. (2006). "Generalized Penetration Depth Computation." *Proceedings of the ACM Solid and Physical Modeling Conference*, Cardiff, UK, 173-184.
- Zhang, Z. (2000). "A Flexible New Technique for Camera Calibration." *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(11), 1330-1334.

- Zhao, J., and Badler, N. I. (1994). "Inverse Kinematics Positioning Using Nonlinear Programming for Highly Articulated Figures." *ACM Transactions on Graphics*, 13(4), 313-336.
- Ziou, D., and Tabbone, S. (1998). "Edge Detection Techniques-An Overview." *International Journal of Pattern Recognition and Image Analysis*, 8, 537-559.
- Zoppi, M. (2002). "Effective Backward Kinematics for an Industrial 6R Robot." *Proceedings of the Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, Montreal, QC, Canada, DETC2002/MECH-34256.
- Zordan, V. B., and Hodgins, J. K. (1999). "Tracking and Modifying Upper-body Human Motion Data with Dynamic Simulation." *Proceedings of the Eurographics Workshop on Computer Animation and Simulation*, Milan, Italy, 13-22.
- Zuliani, M., Kenney, C., and Manjunath, B. S. (2004). "A Mathematical Comparison of Point Detectors." *Proceedings of the Conference on Computer Vision and Pattern Recognition Workshop*, Washington, DC, USA, 172.