CEIG'08, Barcelona, Sept. 3-5 (2008)
L. Matey and J. C. Torres (Editors)

Auto-labeling as a minimization problem with virtual occlusions

A. Cuevas[1], J. Rodriguez-Navarro[1 2] and A. Susín[2]

[1]Easy Biomechanics, SL.; [2]LABSID- Dept. Matematica Aplicada 1 (UPC)

Abstract

In this paper we propose a fast algorithm for the automatic labeling of a set of predefined markers in an optical motion capture system. This algorithm is facing the problem as a minimization problem, using a virtual representation of the real model to predict possible occlusions that happen in the captured images. Moreover, we take advantage of the knowledge of the cameras parameters to solve potential labeling conflicts between markers.

Categories and Subject Descriptors (according to ACM CCS): I.3.7 [Computer Graphics]: Three-Dimensional Graphics and Realism - animation.

1. Introduction

Human motion capture is used to acquire motion data from a video of a real moving person. Applications include virtual reality for human control-interface and video games for realistic simulation of human motion. Motion capture systems can be divided into magnetic, mechanic, and optical. Magnetic systems use electromagnetic sensors connected to a computer which can produce 3D data in real-time with low processing costs. However, a magnetic system restricts movement due to cabling. Mechanical systems use special suits with integrated mechanical sensors that register the motion of articulation in real-time and with no processing. Optical systems are based on photogrammetric methods. They provide high accuracy, complete freedom of movement, and the possibility of interaction between different actors with a higher computational cost.

Optical motion capture is also effective in a clinical context for medical investigation such as the assessment of orthopedic pathologies. Other research areas are kinesiology and biomechanics for movement analysis, performance and injuries research in sports; and robotics for robots control.

An optical motion capture system records translational data for individual points corresponding to retro-reflective markers. Usually infrared (IR) cameras and spotlights are used to achieve better tracking results. The 3D position of the markers is computed from their projections to the camera. Theoretically, two cameras should be adequate for this kind of registration. In fact, more cameras are used due to the frequent marker occlusions found while recording.

Usually a big number of markers are needed to obtain a feasible data set for posterior motion analysis. One of the most tedious tasks of a recording session is the labeling of those markers. Indeed, previous to start a real time capture the system needs to identify all the markers used in the capture in order to track their position along the time. Usually this task is done manually, taking considerable time at the start of a recording session. Many other previous approaches are focused more in the tracking process starting from a previous known position and so, they initialize manually the system for the first time. We propose a solution for this initial labeling problem by modeling the correspondence between initial captured points and the markers of the chosen model as a minimization problem. Thus, the system is able to deal with the small deviations between the theoretical marker positions (relative to the model joints) and the real ones placed by the user.

2. Related work

Nowadays, marker-based motion capture systems have developed into a standard tool within the technical repertoire of professionals in computer animation and biomechanical analysis. Unfortunately, generating a moving kinematic skeleton model from raw marker trajectories with commercial tools is often still a semiautomatic procedure [Vic08, Mot08, Sim08]. Other techniques known as markless, based on the tracking of silhouettes [Org08],

initially avoid the labeling process but they still need an automatic procedure that takes around one minute for the subject calibration in order to track the actor.

One of the first decisions is to fix the marker setup that should be able to capture internal rotations of the main joints. Sources for anthropomorphic feature points of the human body are found in the h-anim project for humanoid animation [Hum08] and in the CAESAR project [Cae08]. Commonly the models that are used for the lower body have between 15 and 23 markers [VDO92] and its location on the subject's body also depends on the complexity of the model. In fact, a complete humanoid model can contain around 30 markers.

After calibration of the system [Tsa86, SMP05] marker positions can be found using image based techniques [Fau93] that are software implemented or by hardware on the cameras [Nat08]. Mainly marker data analysis is solved as a post-process and essentially is focused in different tracking strategies. Some work is focused on estimating the positions of joints and skeleton (topology and parameterization) for an articulated body [SPB*98, KJF05, ATS06]. There are also other interesting papers dealing with multiple interrelated bodies [RL02, QQZ07] but they are based on image postprocess more than real time capture like our approach.

In general the first labeling association is solved manually and it is considered as an initialization for the tracking algorithm. This can be assumed if no real time tracking is aimed which is our case. Before starting the tracker one needs to identify each of the visible markers in each camera and a manual labeling would be too tedious. We present an automatic method for solving the starting labeling step based on a known human skeleton and marker distribution. Essentially we combine 2D values obtained from the images with the 3D coordinates coming from the skeleton model. One important distinction is when you have a big number of cameras (more than 8 or 10) or you use a lower configuration with 4 or less cameras [BBH05] then, the occlusions problem becomes more important and our approach is still more appropriate.

The algorithm is divided in three steps. The initialization step: a first approximation of the scale and orientation of the human model according to the real data. The fitting step: applying a minimization process to obtain a more accurate fitting based on the occlusion information obtained from the model. Finally, the labeling step: all the visible markers are identified and correctly labeled.

3. Initialization

We start from a normalized human model with a fixed hierarchic bone structure. We also assume that a set of marker positions are known for this normalized model.

Previous to the minimization process, a good enough initialization for the human model is required. This means that we must approximate scale, orientation and translations from the normalized model to the real one. We obtain an uniform scale factor from the height of the person; orientation approximation is found from the bounding box assuming the up direction is known. Translation is calculated automatically using the procedure explained next.

Let N be the number of cameras, for each camera, we denote by c_i the 2D centroids calculated by averaging the corresponding detected blobs. After scaling and orienting, the initial normalized model becomes our reference model (RM) and we denote by C_v its 3D centroid. A first approximation of the remaining translation vector T can be obtained imposing the following relation for all cameras

$$c_i = P_i(C_v + T), \quad i = 1, ..., N, \tag{1}$$

where P_i is the known projection matrix corresponding to each camera. Eq. (1) is in fact an over determined system of equations that can be solved using the following steps:

First for each equation we multiply

$$c_i^* c_i = c_i^* P_i (C_v + T) = 0, \tag{2}$$

where, c_i^* is the skew matrix defined by the coordinates $c_i = (c_{ix}, c_{iy}, c_{iz})$

$$c_i^* = \begin{pmatrix} 0 & -c_{iz} & c_{iy} \\ c_{iz} & 0 & -c_{ix} \\ -c_{iy} & c_{ix} & 0 \end{pmatrix} \tag{3}$$

Then, collecting T from the previous equations we obtain

$$T = \left(E^T E\right)^{-1} E^T B, \tag{4}$$

where

$$E = \begin{pmatrix} c_0^* P_0 \\ \dots \\ c_N^* P_N \end{pmatrix}, \quad B = \begin{pmatrix} -c_0^* P_0 C_v \\ \dots \\ -c_N^* P_N C_v \end{pmatrix}. \tag{5}$$

4. Minimization

Once we have a rough initial value for the different parameters involved in the labeling problem, we will build a minimization procedure improving the correspondence between the RM and the real one. Those parameters are scale, rotation and translation. In our approach we consider three dimensional vectors for both scale $S = (s_x, s_y, s_z)$ (allowing different deformation in each

direction) and translation $T = (t_x, t_y, t_z)$. Rotation is represented by normalized quaternions R_q. Thus, final function mapping the 3D points with the 2D points in the images is a 9 degrees of freedom function.

$$x_{ij} = P_i \cdot S \cdot X_j + T, \quad i = 1,...,N, \; j = 1,...,M \quad \textbf{(6)}$$

Here X_j denotes the 3D coordinates of each marker and x_{ij} its 2D projection in the i^{th} camera.

When analyzing each camera image, we will detect the most relevant points (blobs) denoted by b_{ik}. Ideally each of these blobs will match with one of the projections of the markers. In fact, only those markers that can be visible in the images are considered. This means, for instance, to discard the markers in the back when the image is frontal because they are occluded by the body. This test of occlusion in our case is performed by an approximation of the body model build by capsules (see fig.3). A raytrace algorithm from the marker to the camera location is used to decide Jif Jis Jvisible. JOnly Jthe Jprojections Jof Jthe Jvisible markers x_{ij}^v are considered as feasible points in the minimization procedure. We can define a cost function of the 9 parameters that computes the distance between the obtained blobs and the expected projection coordinates of the visible markers. This function can be stated implicitly as:

$$F(S,T,R_q) = \sum_i^N \left(\sum_j^M \left(\min_k | x_{ij}^v - b_{ik} | \right) \right) \quad \textbf{(7)}$$

Here, eq. (6) is used in the computation of x_{ij}^v. Due to the non-linearity of the problem, the minimum of the function is computed using the Levenberg-Marquardt algorithm [HZ03].

We remark that the approximation of the body used here sometimes can produce false visible markers (which is better than discard strictly each marker). Moreover, in order to avoid possible errors, only small values on the evaluation of function (7) are considered.

5. Labeling

After the minimization step, the RM has a better adjust to the real position and the automatic labeling procedure can be started. The labeling process consist in identify each marker in the different cameras. It is very important to avoid a mistake at the beginning of the labeling process that can confuse the system. Thus, a conservative approach is chosen consisting in the division of the assignation blob-marker in three substeps.

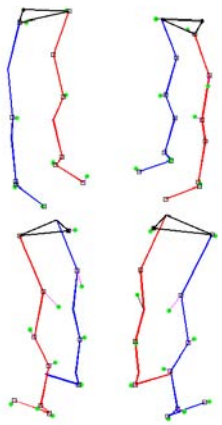

Fig. 1. *Mapping between blobs and marker projection*.

5.1 Visible markers identification

The first step consists in identifying the blobs found in the image using the visible marker projections computed in the minimization procedure. We map each blob with the closer projection. We classify three different kind of mapping: the ones with an *exclusive* marker projection (a one-to-one correspondence). The *conflictive* ones, arising when two or more blobs are associated to the same marker projection, and finally, the ones *not associated* because no blob is found in the marker projection neighborhood.

5.2 Conflict solution

In the second step conflictive blobs are faced. Here we need to use the information coming from different images. For solving the different conflicts we must consider three cases that are related to the number of blobs labeled with the same marker in the other images.

When a blob has been associated *exclusively* in 2 or more cameras, by triangularization one can compute the 3D coordinates of this point. Then, projecting the point into the conflictive image, we take the closer real blob to this projection as the correct one.

On the other hand, if the conflictive marker has been associated *exclusively* only in one camera, no 3D reconstruction is possible and we are forced to use the RM for labeling. To assign the proper blob in the conflictive image, we make a 3D reconstruction using the exclusive blob assigned in the other camera and the conflictive candidates. We choose as the right one, the candidate that recovers the 3D point closer to the marker in the RM.

The last case is when a visible marker has not been identified in any camera. Now we have no sufficient information to make a correct decision and we choose to label the nearest blob to the marker projection.

When we finish this first assignation, the remaining markers are considered free for the next procedure. It is

also important to use the information of the identified blobs for the rest of the process.

5.3 Remaining Blobs

At the end, we can have some of the blobs without labeling. Now we will use the projection of the markers but without the occlusion test like in the previous sections. This is just to consider the case in which the RM, as it is not anatomically precise, can give false negative (return occlusions when they are visible). Thus, we will traverse the unlabeled blobs and compute the distance to the not assigned marker projections. The blob is temporally assigned to the nearest one. At the end if several blobs are pre-assigned to the same marker, the one with minimum distance will be chosen. The rest of possible blobs identified in the images are then considered as false markers possibly due to illumination changes in the capture.

The labeling process is summarized in algorithm 1.

Fig. 2. *Virtual model and marker configuration in a gait analysis model.*

```
//1st Step: Visible markers identification
for cam=1:NCAM
  for each blob in Blobs[cam]
      notAssigned[cam].add(blob)
  end
  //Visible markers projection
  xvcam = Projection[cam]*Xv
  xvcam.setAssociated(false)
  for each blob in Blobs[cam]
      xvassign  = xvcam.findClosestMarker(blob)
      //Exclusive or Conflictive marker
      Classificate(xvassigned,blob)
  end
end
//2nd Step: Conflict solutions
for cam=1:N CAM
  for each xc in Conflictive[cam]
      helpers = GetMarkerInOtherCameras(xc)
      if (helpers.length >1)
          findClosest2DRepjFrom3DReconst(helpers, xc)
      else if (helpers.length == 1)
          findClosest3DMarkerInRM(helpers, xc)
      else
          findClosest2DBlob(xc)
      end
  end
end
//3rd Step: Unassigned blobs
for cam=1:N CAM
  //Not assigned and invisibles markers projection
  xna = Projection[cam]*Xna
  findClosest2DBlob(xna)
end
```

Alg. 1. *Outline of the labeling algorithm.*

6. Results

The algorithm has been tested in the easy Biomechanics facilities. The motion capture optical system is composed by four cameras Basler© 601f IR monochrome at 60 frames per second in a 2x1.8x2m volume of capture. We have used two different normalized models, a whole body one with 26 markers and a gait analysis model (figure 2) with 15 markers. In figure 6, one can see the real images corresponding to this model.
As it has been pointed out, the body volume has been done with a set of capsules approximating each of the body segments (figure 3).

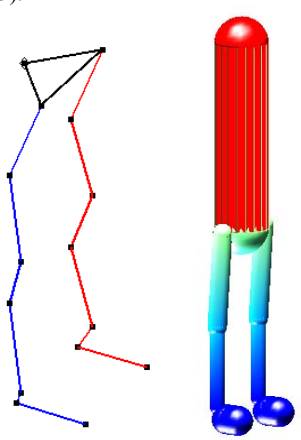

Fig. 3. *On the left, the skeleton based on a gait analysis model is shown. On the right, in a different scale, the normalized model with capsules.*

The quality of the approximation of the virtual model is one of the main features in order to obtain good results. Of course, any other well known problem related to a motion capture system may have influence, like the quality of the recorded data and the camera calibration parameters.

Figure 4 (top) shows the initial captured blobs representation on a usual test. After the first step, the initialization reached by fitting the projection of the centroid of the scaled model with the 2D centroid of the image blobs (equation 1) is shown in figure 4 (bottom). After the minimization step the skeleton reach a better adjustment to the real markers. The identification results of the markers after the labeling is represented in figure 1.

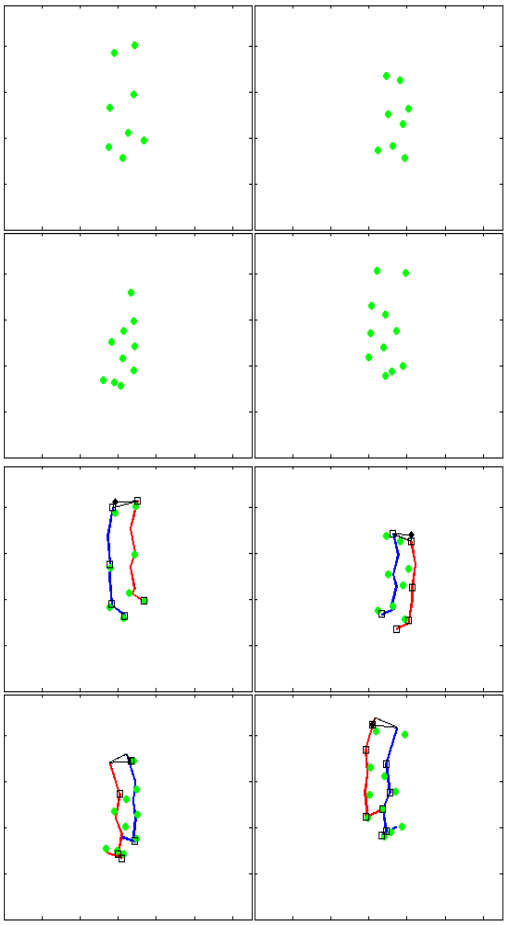

Fig. 4. *Initial blob representation from the recorded data (top). Representation of the virtual skeleton projection on the cameras after the initialization (bottom).*

In figure 5, a closer view of another example (from another capture) where the algorithm solves a blobs conflict is shown. The image shows the left foot markers and their final association after the labeling step. It can be appreciate how two markers choose the same identifying marker causing a conflict. The cyan line represents a conflict. The associations after the second stage are colored in magenta. The black line is the association related to the third labeling step. One can see how the blob 'a' and 'b' have suffered a conflict on the first step. After the second step the blob 'a' is the one that has been associated. The blob 'b' will be labeled on the last step, after the other two had been assigned.

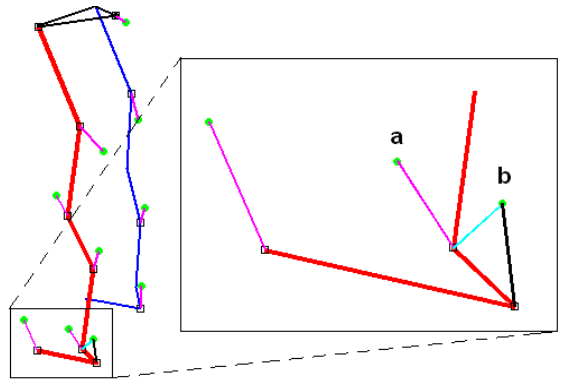

Fig. 5. *Left foot conflict case.*

After different tests, it has been checked that the minimization step allows up to ±20º error on the initial rotation, ±0.25m on the scale and ±0.2m on the translation.

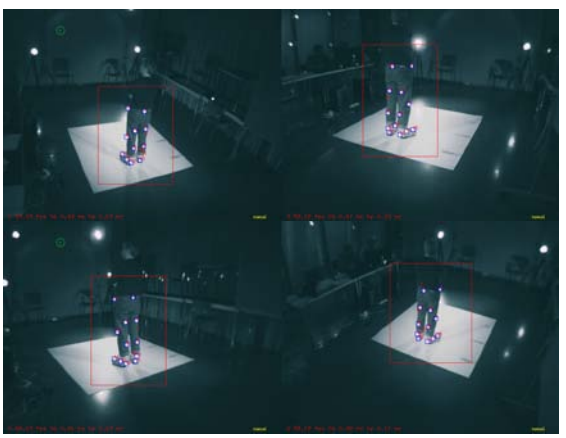

Fig. 6. *Automatic labeling on the real images.*

In that stage, the first iterations are the ones that reach a bigger adjustment of the skeleton. In our case, 10 iterations are needed to reach a good result that allows the last stage to identify the markers correctly.

Model	Step I	Itera-tions	Step II	Step III	Total
Lower 15 markers	0.013s	5	0.8s	0.08s	0.91s
		10	1.3s		1.4s
Whole 26 markers	0.013s	5	1.89s	0.15s	2.05s
		10	3.6s		3.76s

Table 1. *Test results. Step I, II and III correspond to Initialization, Minimization and Labeling steps.*

In the lower body model case, there are some especially conflicting areas such as the feet. This area is more sensitive to the virtual model approximation as well as the location of the cameras due to the markers nearness. Neverthe-

less the algorithm solves their identification in almost all cases.

We have tested our algorithm with a whole body normalized model (figures 7 and 8). In this case the initial model has 26 markers distributed around the body. In table 1, we show the time results for the two models. As it can be observed the iteration number in the minimization step is the more time consuming part. The tests have been implemented with Matlab and executed in an Intel Core 2 Duo 7500, 3 GB of RAM memory and operative system Microsoft Windows XP SP2.

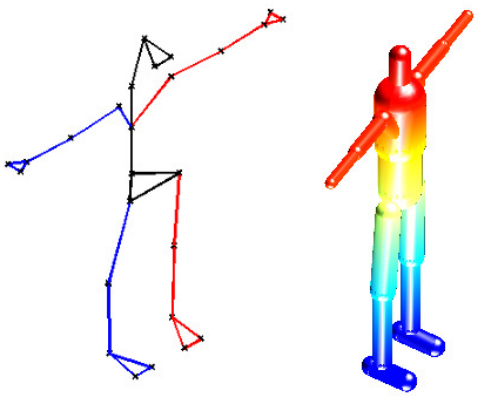

Fig. 7. *On the left, the skeleton based on a whole body model is shown. On the right, the body volume model with capsules.*

The labeling results for these tests are excellent for the gait analysis model, with all markers correctly labeled. The whole body model sometimes produces labeling errors for the hands. They are produced because the markers are very close between them and it is difficult to obtain a precise skeleton definition for the hands. After all, the results are satisfactory because the algorithm solve between 24 and 26 markers in all the tested cases.

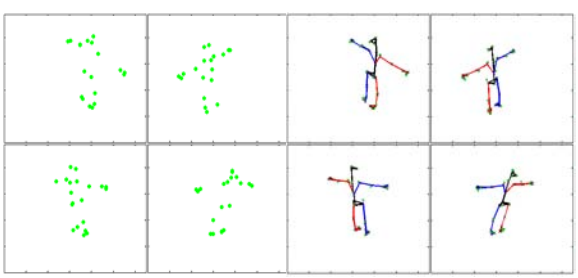

Fig. 8. *Initial blob representation from the recorded data for a whole body model (left). Representation of the virtual skeleton projection on the cameras after the initialization (right).*

7. Future work

Although the algorithm reaches his goal in a very efficient way, we are studying the completely automatization of the initialization process (section 2). One possible way for solving this problem is doing the principal component analysis of the blobs on the images. As we already have all the camera parameters, there is a direct relationship between those principal components and the rotation and scale of the model.

Using the current technology, we are working to move the occlusion calculations of the virtual markers to the graphic card. Some of these graphic cards support occlusion queries techniques which allow working on a higher efficient level. This technique makes possible the use of more complex and precise virtual models. Of course, a bigger number of cameras minimize the occlusion problem and, as a consequence, eventually the occlusion test can be ignored.

After the tests, we are convinced that the algorithm should work on real time using C++ code together with the graphic card. This fact makes us wonder to the possibility of changing it so we can use it as a tracker on the recording stage. As the sampling rate of the capture is 60 fps no initialization will be needed because the position of the previous virtual model can be used as the initial one and hopefully the algorithm can reach better results in less iterations.

Acknowledgments

Last author is partially supported by TIN2007-67982-C02-01 grant. We thanks the easy Biomechanics research team for their help.

References

[ATS06] AGUIAR E. D., THEOBALT C., SEIDEL H.-P.: Automatic learning of articulated skeletons from 3d marker trajectories. In *Proc. of Int'l Symposium on Visual Computing (ISVC06)* , pp. I: 485--494 (2006).

[BBH05] BUDIMAN R., BENNAMOUN M. and HUYNH D.Q. Low Cost Motion Capture. Proc. Image and Vision Computing New Zealand, Dunedin, New Zealand, 28-29 November 2005.

[BRF*90] BORGHESE, N., Di RIENZO, M., FERRINGO, G., and PEDOTTI, A. Elite: a goal-oriented vision system for moving objects detection. *Robotica* 9: 275--282 (1990).

[Fau93] Faugeras O. *Three-dimensional Computer Vision*. MIT Press (1993).

[HFP*01] HERDA, L., FUA, P., PLÄNKERS, R., BOULIC, R., THALMAN, D.: Using Skeleton-Based Tracking to increase the Reliability of Optical Motion Capture. Human Movement Science, vol.20, pp.313--341 (2001).

[KJF05] KIRK A., J.F. O. B., FORSYTH D.: Skeletal parameter estimation from optical motion capture data. In *IEEE CVPR 2005* (2005).

[Kir06] KIRTLEY, C.: Clinical Gait Analysis: Theory and Practice. Churchill Livingstone, Elsevier (2006).

[Moe98] MOESLUND, T.B.: The Analysis-by-Synthesis Approach in Human Motion Capture: A Review. In: 8th Danish conference in pattern recognition and image analysis, pp. 54--66. Copenhagen (1999).

[MG01] MOESLUND, T.B., GRANUM, E.: A Survey of Computer Vision-Based Human Motion Capture. In: Computer Vision and Image Understanding, vol. 81, no. 3, pp. 231--268 (2001).

[RL02] RINGER M., LASENBY J.: Multiple hypothesis tracking for automatic optical motion capture. In *ECCV02* pp. 524--536 (2002).

[QQZ07] QIAN Y., QING L., ZHIGANG D.: Online Motion Capture Marker Labeling for Multiple Interacting Articulated Targets. Computer Graphics Forum (Proc. of Eurographics 2007), pp. 477--483.

[SMP05] SVOBODA, T., MARTINEC, D., PAJDLA, T.: A convenient Multi-Camera Self-Calibration for Virtual Environments. Teleoperators and Virtual Environ-ments, pp 407--422. MIT Press (2005).

[SPB*98] SILAGHI, M., PLÄNKERS, R., BOULIC, R., FUA, P., THALMAN, D.: Local and Global Skeleton Fitting Techniques for Optical Motion Capture. In: International Workshop on Modeling and Motion Capture Techniques for Virtual Environments. LNCS vol. 1537, pp. 26--40. Springer, Heidelberg (1998).

[Tsa86] TSAI, R.Y.: An Efficient and Accurate Camera Calibration Technique for 3D Machine Vision. In: IEEE Conference on Computer Vision and Pattern Recognition, pp. 364--374 Miami Beach (1986).

[VDO92] VAUGHAN, C.L., DAVIS, B.L., O'CONNOR, J.C.: Dynamics of Human Gait (2nd edition) Kiboho Publishers (1992)

[ZS98] ZHENG, J.Y., SUEZAKI, S.: A Model Based Approach in Extracting and Generating Human Motion. In: 14th International Conference on Pattern Recognition, vol. 2, pp. 1201--1205 (1998).

[Vic08] http://www.vicon.com/products/bodybuilder.html

[Mot08] http://www.motionanalysis.com/

[Sim08] http://www.simi.com/

[Nat08] http://www.naturalpoint.com

[Org08] http://www.organicmotion.com/

[Hum08] http://www.h-anim.org/

[Cae08]http://sae.org/technicalcommittees/caesarhome.htm