Markerless Motion Capture with Unsynchronized Moving Cameras

Nils Hasler[1], Bodo Rosenhahn[1,2], Thorsten Thormählen[1], Michael Wand[1], Juergen Gall[1,3] , Hans-Peter Seidel[1]
[1]MPI Informatik [2]TNT, Hannover University [3]BIWI, ETH Zurich

Abstract

In this work we present an approach for markerless motion capture (MoCap) of articulated objects, which are recorded with multiple unsynchronized moving cameras. Instead of using fixed (and expensive) hardware synchronized cameras, this approach allows us to track people with off-the-shelf handheld video cameras. To prepare a sequence for motion capture, we first reconstruct the static background and the position of each camera using Structure-from-Motion (SfM). Then the cameras are registered to each other using the reconstructed static background geometry. Camera synchronization is achieved via the audio streams recorded by the cameras in parallel. Finally, a markerless MoCap approach is applied to recover positions and joint configurations of subjects. Feature tracks and dense background geometry are further used to stabilize the MoCap. The experiments show examples with highly challenging indoor and outdoor scenes.

1. Introduction

Markerless Motion Capture (MoCap) is an active field of research in computer vision and graphics with applications in animation (games, avatars), medicine, and sports science. In contrast to commonly used marker based approaches, markerless methods are based on sensor data (usually images) without special preparation of the subject. The goal is to determine position and orientation as well as the joint angles of a human body from image data, as depicted in Figure 1.

In most approaches the cameras are assumed to be synchronized, calibrated in advance, and static. However, these requirements demand specialized and often expensive hardware since consumer cameras usually do not provide these properties. In this work we explain how to use multiple standard handheld video cameras, observing a moving subject for markerless motion capture. To achieve this goal, we perform three basic steps. Firstly, we compute the camera paths and reconstruct sparse background geometry via Structure-from-Motion (SfM). Then, the reconstructed background models are registered to each other to calibrate each camera with respect to a global coordinate system. The applied techniques are similar to [17, 6, 27].

Figure 1. Three example sequences. **Left**: Indoor-climbing. **Middle**: Dancing in a halfpipe. **Right**: Running and jumping in an outdoor scene.

Secondly, to obtain a synchronous multi-view video stream the cameras have to be synchronized. Unlike [25], we do not synchronize audio and video streams of a single view but the audio streams of several cameras. In [24] a system for detecting speaker location using a multi-modal approach correlating lip-movement with audio signals in a training step is described. Synchronization with a RANSAC based technique, which exploits properties of moving silhouettes, has recently been presented [23]. However, the approach is restricted to static cameras. To avoid issues with wide baselines, only partially overlapping views, etc., we propose to use the simultaneously captured audio streams to synchronize the video signals.

Finally, the resulting synchronized multi-view video stream with separate projection matrices for each camera frame can be used to perform classical markerless pose tracking. Because the cameras are moving, classical background subtraction methods cannot be applied for finding the silhouettes, which are used as input for the pose estimation. Additionally, the cluttered background forbids the use of color keying. Here, we rely on a silhouette based approach, performing joint pose estimation and segmentation [20]. However, similar techniques such as Posecut [2] or the approach by Dambreville et al. [5] could be employed.

An early work on motion capture with moving cameras is given in [9]. The method uses an affine camera

978-1-4244-3991-1/09/$25.00 ©2009 IEEE

model, relies on manually tracked features, and explicitly exploits constraints associated with a dynamic articulated structure. Other works with (sometimes) moving monocular cameras involve learning approaches, such as presented in [22, 26, 28]. For motion capture, recent overview articles, e.g., [12, 18] exist, but they do not explicitly address moving or unsynchronized cameras. A recent research strand aims to integrate further sources of information in motion capture, e.g., by capturing light sources [1] or by using physical models and forces arising from a ground plane [3, 29]. Automatically reconstructed background geometry can be integrated into the tracking process to regularize the pose equations. However, in contrast to earlier works, we do not rely on a simple ground plane but on a triangulated surface, which covers more complex situations occurring, e.g., in outdoor scenarios.

In this work the following contributions to the state-of-the art in motion capture are described:

1. Using SfM, automatic camera registration and synchronization, and static background reconstruction we are able to provide a fully automatic pipeline to integrate moving cameras into markerless motion capture approaches.

2. The audio channels of the cameras are used to synchronize video streams to get a set-up for multi-view human motion capture that does not rely on static or hardware synchronized cameras, but just on audio streams captured in parallel to video by unsynchronized handheld cameras.

3. Automatically reconstructed background geometry can be used to penalize intersections between body parts with the background geometry during tracking. The background model is a triangular surface mesh which is much more complex than simple ground plane constraints used before.

The paper is organized as follows: Section 2 explains the Structure-from-Motion approach, which we use to calibrate the cameras. The method detects outliers caused by the moving subject and estimates the camera parameters of all cameras with respect to the static background region. Section 3 describes our method for video synchronization. Here, we use the audio channel to compute temporal offsets between the employed cameras, so that the video streams are synchronized. In combination, the parts yield synchronized multi-view frames, which can then be used for standard markerless motion capture. Since background subtraction is not possible (the cameras are moving and the lighting conditions are changing), we rely on a silhouette driven approach summarized in Section 4. Experiments are presented in Section 5, and the paper concludes with a summary in Section 6.

2. Camera Calibration

Automatic camera calibration and 3D reconstruction of rigid objects from video (Structure-from-Motion) is a well established technique in computer vision. However, the established algorithms have been developed for a single moving camera and not for multiple moving cameras observing the same scene.

Nevertheless, the algorithms developed for a single moving camera can be applied on the input video sequence of each camera independently, which is described in the following subsection. In subsection 2.2, we present an algorithm to register these independent reconstructions into a common global coordinate system. Subsection 2.3 discusses how a 3D surface model of the static background can be reconstructed.

2.1. Single Camera Structure-from-Motion

To estimate the parameters of a single moving camera we apply a feature-based approach, where corresponding feature points are determined in consecutive frames with the KLT-Tracker [21] or SIFT matching [10].

First, we need to filter out those corresponding feature points that do not belong to the static background, because we want to estimate the camera motion with respect to the static part of the scene. To detect those trajectories we apply RANSAC with multi-view constraints (see [17, 7] for details). All the trajectories that do not fulfil these multi-view constraints, e.g., because they belong to moving objects, are ignored in the following Structure-from-Motion estimation.

Let's assume we are given a video sequence with K images I_k, with $k = 1, \ldots, K$, and we have established J trajectories of 2D feature points $\mathbf{p}_{j,\,k}$, with $j = 1, \ldots, J$, belonging to the static background.

Once corresponding feature points between consecutive frames are established, the parameters of the 3×4 camera matrix $\mathtt{A}_k$ can be estimated for every frame k, and for each trajectory of a 2D feature point a corresponding 3D object point $\mathbf{P}_j$ is determined.

To estimate initial parameters for all the camera matrices $\mathtt{A}_k$ and 3D object points $\mathbf{P}_j$, we apply an incremental Structure-from-Motion approach similar to [17].

If the errors in the positions of the 2D feature points obey a Gaussian distribution, the Maximum Likelihood estimator for camera parameters and 3D object points is called bundle adjustment. Bundle adjustment minimizes the reprojection error of the 3D object points into the camera images:

$$\underset{\mathtt{A}_k, \mathbf{P}_j}{\arg\min} \quad \sum_{j=1}^{J} \sum_{k=1}^{K} \mathrm{d}(\mathbf{p}_{j,\,k}\,,\, \mathtt{A}_k\, \mathbf{P}_j)^2, \qquad (1)$$

where $\mathrm{d}(\ldots)$ denotes the Euclidean distance and $\mathbf{p}_{j,\,k} = (x, y, 1)$ and $\mathbf{P}_j = (X, Y, Z, 1)^\top$ are written in homogeneous coordinates. Optimizing this bundle adjustment

equation distributes the error equally over the whole sequence and is the last step in our single camera Structure-from-Motion algorithm.

2.2. Multi-camera Structure-from-Motion

If we have captured the same scene simultaneously with N cameras, we first apply the described single camera Structure-from-Motion algorithm independently for each camera. The resulting N reconstructions of camera matrices $\mathtt{A}_{k,\,n}$ and 3D object points $\mathbf{P}_{j,\,n}$ for each camera n, with $n = 1, \ldots, N$, are determined only up to a similarity transformation with 7 degrees of freedom (3 for rotation, 3 for translation, and 1 for scale).

In order to register these N reconstructions into a global coordinate system, we need to estimate the transformations $\mathtt{H}$ between the independent reconstructions. This can be achieved by finding and merging common 3D object points that were tracked in at least two cameras. We find these common 3D object points by pairwise matching, where we follow the approach in [27]. The first constraint that two 3D object points need to fulfil is the similarity constraint, which is met if the color intensity in a window around their tracked position in the camera images is similar. The second constraint is a uniqueness constraint, which enforces that the matching score of the best merging candidates is sufficiently higher than the second best match for the involved 3D object points. This constraint is especially important for scenes that contain repetitive structures because otherwise groups of 3D object points may get merged with the wrong repeated structure. All candidates that fulfil these constraints are used to estimate the transformations $\mathtt{H}$ with a robust estimator. The estimated transformations are applied to the independent reconstructions and common 3D object points are merged. Then a bundle adjustment is performed conjointly over all N reconstructions, minimizing

$$\underset{\mathtt{A},\mathbf{P}}{\arg\min} \quad \sum_{n=1}^{N} \sum_{j=1}^{J} \sum_{k=1}^{K} \mathrm{d}(\mathbf{p}_{j,\,k,\,n}\,,\ \mathtt{A}_{k,\,n}\,\mathbf{P}_{j,\,n})^2. \quad (2)$$

2.3. 3D Surface Reconstruction

Next, we estimate the geometry of the static background of the scene. Our task is now the reconstruction of a surface from the sparse point cloud $\mathbf{P}_{j,n}$. The main problem is the very high level of noise and outliers in this set. We address this problem in a two stage process, following the pipeline outlined in [30]: First, we remove outliers that do not form surfaces and, second, we smooth out the remaining noise. For outlier removal, we use a tensor voting filter that removes points that in their local neighborhood do not form a smooth 2D manifold. As outliers in our case appear as either isolated points or volumetric "clouds" of points, this technique is highly effective. The remaining points form a noisy 2-manifold with significant noise level. We apply bilateral moving least squares filtering to smooth out the remaining noise. We set the parameters for these two steps manually, using a rather large neighborhood size for voting and smoothing and low influence of the bilateral (normal deviation) component. We employ the same parameter set for all three scenes in this paper. Finally, we reconstruct a triangle mesh by consistently orienting the normals of the reconstruction result and running a marching cubes algorithm on the implicit surface defined by the point and normal pairs [8].

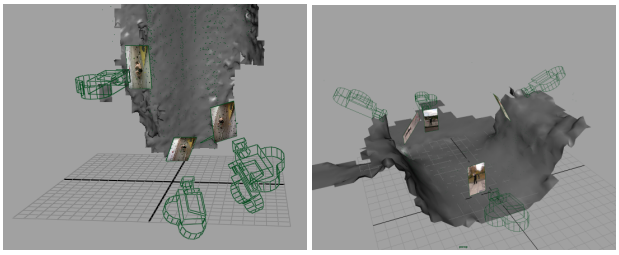

Figure 2. Reconstructed climbing wall (left) and halfpipe (right), estimated camera positions with corresponding camera images (cf. Fig. 1).

3. Camera Synchronization

Consumer level cameras are nowadays typically equipped with a built-in low quality microphone. Hence, cameras are able to capture audio and video streams in parallel. Synchronicity of audio and video channels is guaranteed by the capturing hardware. This property can be exploited to synchronize video streams of cameras capturing the same scene with extremely wide baselines or even non-overlapping views by analyzing the corresponding audio streams.

3.1. Synchronizing Audio Signals

In signal processing a widely employed technique for detecting similar segments within another signal is cross correlation [16]. Assume that a_i represents the audio signal captured by the ith camera in the time domain. The cross correlation between the audio signal of cameras i and j can then be computed by

$$a_i \star a_j \equiv \overline{a_i}(-t) * a_j(t), \quad (3)$$

where $\star$ denotes cross correlation, $*$ convolution and $\overline{a_i}$ complex conjugation [15]. Cross correlation of discrete signals can thus be computed efficiently using Fast Fourier Transform (FFT). Figure 3 shows the audio streams of two cameras and their cross correlation. The audio delay between the signals can be found by locating the peak of the cross correlation signal (cf. Fig. 3).

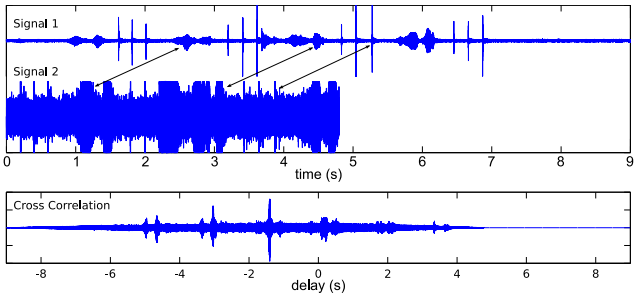

Figure 3. **Top**: The waveforms of two of the captured audio streams. Correct correspondences are indicated by the black arrows. **Bottom**: The cross correlation between the signals. A distinct maximum can easily be computed, although one of the signals is severely corrupted by background noise and the sound level of both audio streams are very different.

This approach works exceptionally well for estimating the temporal offset Δ_{ij} between two given cameras i and j if the observed scene is small, e.g., the indoor-climbing sequence of Section 5.1. However, the relatively low speed of sound in air of about $c \approx 340\,\mathrm{m/s}$ introduces errors. If the difference in distance to the sound source from two cameras is greater than approximately $6.8\,\mathrm{m}$ the delay can be large enough to offset the resulting synchronization by one frame at a frame rate of $f = 25\,\mathrm{Hz}$.

3.2. Correction for Large Camera Displacements

Fortunately, the error can be compensated for, if the positions of cameras and sound source are known. The camera positions are known from the SfM based camera calibration. If the position of the sound source is also known, e.g., because it can be located in the scene, it is possible to account for the time-of-flight, since

$$d_{ij} = \Delta_{ij} + \frac{1}{c}(\mathrm{d}(\mathbf{c}_j - \mathbf{s}) - \mathrm{d}(\mathbf{c}_i - \mathbf{s})). \qquad (4)$$

Here, d_{ij} is the delay between the audio signals of camera i and j, $\mathbf{c}_i$ and $\mathbf{s}$ are the positions of cameras and sound source respectively, and $\mathrm{d}(\dots)$ denotes the unsigned distance function. Since the temporal shift of every camera Δ_i can be expressed relative to an arbitrary point in time and $\Delta_{ij} = \Delta_j - \Delta_i$, w.l.o.g. $\Delta_1 = 0$. For N cameras we are thus left with $N - 1$ unknown Δ_i. Although we can set up $N(N-1)/2$ equations (all pairwise combinations of N cameras) using Eq. (4), only $N - 1$ are linearly independent. A unique solution for all Δ_i can be found by solving the resulting linear equation system with standard numerical techniques.

By manually identifying the primary sound source in one of the original video sequences, we are able to apply this technique to synchronize the running sequence shown in Section 5.3. For the other sequences this manual step was not necessary.

3.3. Unknown Sound Source Location

The more general problem, namely solving for sound source location and temporal shifts simultaneously cannot be solved as easily because the problem is underdetermined. In addition to $N - 1$ shifts Δ_i we also need to solve for the coordinates of $\mathbf{s}$. In the 2D case we have $N - 1 + 2$ unknowns but only $N - 1$ linearly independent equations. The problem can be alleviated by using k sound sources instead of one, since we are able to generate $N - 1$ additional equations per sound source and only add 2 unknowns. Because the Δ_i remain unchanged no matter which sound source is received, we get $N - 1 + 2k$ unknowns and $k(N - 1)$ linearly independent equations. Thus, the problem is solvable only for $k(N - 1) > N - 1 + 2k$. Every equation is quadratic and the number of equations is $O(N)$. The Levenberg-Marquardt algorithm [11], is able to solve the system quickly, unless the system is badly conditioned, e.g., camera or sound source positions are collinear.

A concern with this setup is that we need to be able to distinguish sounds received from the different sources. Although there is work on distinguishing different speakers automatically [13], it is much simpler and more robust to place sound sources in the scene that playback distinguishable recordings. If a single sound source is sampled at different points in time, it is not necessary that the emitted signal is known a priori. However, care must be taken that the trajectory of the sound source deviates significantly from a straight line.

3.4. Evaluation

The audio based synchronization is evaluated using synthetic experiments on the one hand and demonstrated as part of a real world system on the other. The synthetic experiments serve to demonstrate the feasibility of the approach and to find a bound on the expected accuracy in a real world scenario. All experiments use four cameras and three or four sound sources.

Cameras and sound sources are distributed randomly on a plane within a $100\,\mathrm{m} \times 100\,\mathrm{m}$ square and random Δ_i are chosen in the range -5 s to 5 s. Figure 4 shows several experiments displaying the robustness to noise by adding a random offset to the d_{ij}.

3.5. Limitations

The naïve audio based synchronization is able to achieve matching errors below 20 ms, which is sufficient to align 25 Hz video streams. However, the actual delay between cameras may still be up to $\frac{1}{2}$ frame. This may cause inaccuracies during tracking, especially when fast motion is encountered. Unfortunately, even if the audio based synchronization is perfect, it is impossible to get more precise results unless hardware synchronization is used.

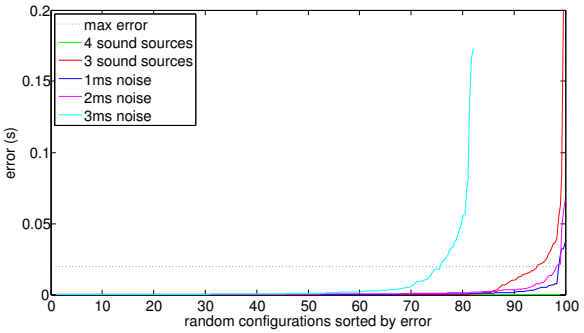

Figure 4. Errors for 100 randomly generated configurations. **Green**: The four sound sources case can be solved optimally for all configurations. **Red**: For three sound sources ambiguities exist and lead to increased errors when the wrong minimum is chosen. **Other colors**: When noise is added to the four sound source setup, some examples cannot be solved accurately any longer. Still, for most configurations, the algorithm converges to the correct minimum. The dashed line shows the error that must not be exceeded for the correct frame offsets to be computable.

4. Motion Capture

Once the cameras are calibrated and synchronized, any markerless motion capture system suited for synchronized and calibrated cameras can be used as long as it does not rely on background subtraction. In this work we use a method similar to [19]. The system requires the subject to be scanned with a 3D scanner to obtain suitable priors for the body shape.

4.1. Kinematic Chains

Articulated objects are modeled as kinematic chains, well known from robotics [14]. Twists in matrix notation ξ_i and angles θ_i are used to model n joint locations given a priori. The consecutive evaluation of exponential functions allows modeling the respective movements of a point X_i on a given end-effector

$$X_i' = \exp(\theta\xi)(\exp(\theta_1\xi_1)\dots\exp(\theta_n\xi_n))X_i, \qquad (5)$$

where $\exp(\dots)$ denotes the exponential map. Similar to other works, we denote a pose configuration by the $(6+n)$-D vector $\chi = (\hat{\xi}, \theta_1, \dots, \theta_n) = (\hat{\xi}, \Theta)$ consisting of the 6 degrees of freedom for the rigid body motion in vector notation $\hat{\xi}$ and the joint angles Θ. Since χ is unknown, the task is to compute the vector χ from (calibrated and synchronized) image data.

4.2. Silhouette Extraction

To fit a given surface model to image data, we first perform a segmentation of the image, based on level set functions $\Phi \in \Omega \mapsto \mathbb{R}$ which are regularized with a 3D shape prior. A level set function splits the image domain Ω into two regions Ω_1 and Ω_2 with $\Phi(x) > 0$ if $x \in \Omega_1$ and $\Phi(x) < 0$ if $x \in \Omega_2$. The null-space or kernel of Φ marks the boundary between both regions.

To achieve partitioning, we minimize the energy of the Chan-Vese model [4]. The fit of each intensity value to a corresponding region is measured in terms of probability densities p_1 and p_2. The densities are modeled by local Gaussian distributions and generated from the color histograms of moving fore- and background.

Since, in addition to the image features, a 3D surface model of the subject is given a priori, we use the projection of the 3D shape as an additional shape prior for segmentation, yielding

$$E(\Phi, p_1, p_2, \chi) = \underbrace{\lambda \int_\Omega (\Phi - \Phi_0(\chi))^2 dx}_{\text{shape error}} - \underbrace{\int_\Omega \mathrm{H}(\Phi)\log p_1 + (1-\mathrm{H}(\Phi))\log p_2 + \nu|\nabla \mathrm{H}(\Phi)|\, dx}_{\text{segmentation}}, \qquad (6)$$

where $\mathrm{H}(s)$ is a regularized Heaviside (step) function. Since this version of segmentation relies on the pose of the 3D shape and vice versa, the segmentation is used to compute a 3D pose (Section 4.3), and segmentation and pose estimation are iterated until convergence is reached. The outputs are the parameters of the kinematic chain and the segmentation of the images.

4.3. Pose Estimation

Once the image contours are extracted (for a given initial pose), we can register the (projected) surface mesh to the image contour by computing the closest point correspondences. After establishing the correspondences, we use the image points on the contour line to reconstruct 3D projection rays. We model each projection ray as a 3D Plücker line $L_i = (n_i, m_i)$, consisting of a 3D (unit) direction n_i and 3D moment m_i [14]. An optimization is then performed to minimize the spatial distance between both contours: The error function for each point-line pair can be expressed as

$$(\exp(\theta\xi)\exp(\theta_1\xi_1)\dots\exp(\theta_j\xi_j)X_i)_{3\times 1} \times n_i - m_i = 0, \qquad (7)$$

Since $\exp(\theta\xi)X_i$ is a 4D vector we skip the homogeneous component (which is 1) to evaluate the cross product with n_i. Then the equation is linearized and iterated to optimize for all correspondences simultaneously.

5. Experiments

In this section we present motion capture results obtained with unsynchronized and moving handheld video cameras. We present experiments with one subject climbing indoors in a climbing gym and another subject dancing and carrying a stereo. In a third experiment a subject is running

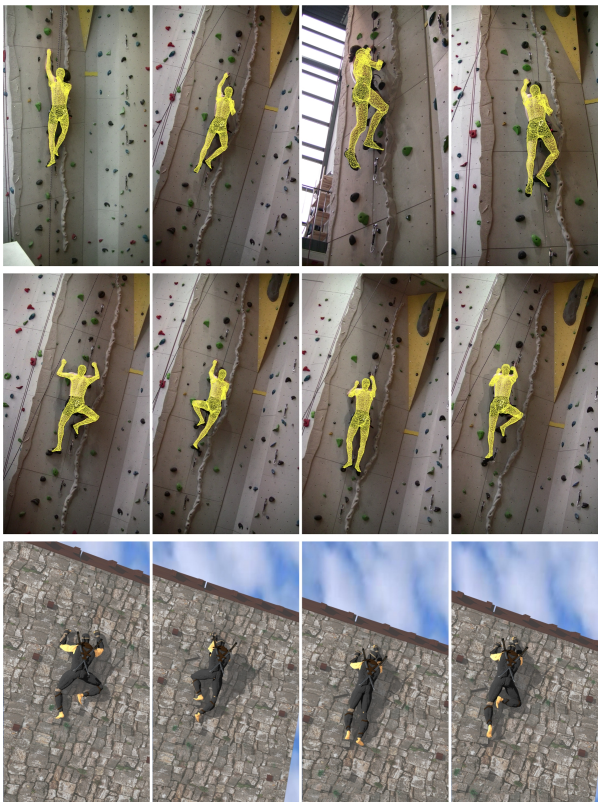

Figure 5. **Top**: Pose result at one time instance overlaid into the four camera views. **Middle**: Pose results at different frames shown by the same camera. **Bottom**: Tracked motion applied to an artist's rendition of the scene.

and jumping in an outdoor scene. The three examples are also shown in the video provided with this paper.

5.1. Climbing

The first experiment is captured in a climbing gym. Four handheld cameras are used to record an athlete climbing a simple route. Afterwards, the subject is scanned and rigged using a commercial full body laser scanner. Then a Structure-from-Motion algorithm is run on the camera streams as explained in Section 2. After registering, the cameras' background geometry is recovered. Then, the video streams are synchronized as explained in Section 3. As handheld cameras are used, the video streams exhibit strong jitter. Additionally, shadows, illumination changes, and the perspectives of the cameras, which are restricted to one hemisphere, since views from the back side of the climbing wall are not possible, add up to a challenging sequence. The latter also results in partial occlusions (e.g., of the hands) leading to further issues. Here, the reconstructed geometry of the climbing wall helps to resolve these ambiguities.

Figure 5 shows the surface mesh of the climber projected back on the four automatically synchronized cameras. The back-projected surface mesh fits well to all camera images. This indicates that registration, synchronization and motion capture work well. The figure also shows pose results from one camera at different frames visualizing the camera's movement while the motion of the subject is still well recovered. As can be seen, sometimes the hands are simultaneously occluded by the body in all views. In these cases the hand is usually placed statically on a hold and can therefore be positioned on the reconstructed background geometry using soft-constraints during pose estimation. This prior resolves ambiguities by avoiding singular systems of equations.

5.2. Halfpipe

In the second experiment a subject carrying a stereo is dancing in a halfpipe. In several camera views the other camera men are visible. The music emitted by the stereo helps the camera synchronization and it is possible to accurately reconstruct the halfpipe shown in Figure 2. Although the jacket of the subject is swinging open during his performance, our algorithm tracks the actual motion of the subject, ignoring the flapping garment. Figure 6 shows several frames from the motion capture results.

Figure 6. **Top**: Pose result at one time instance overlaid into the four camera views. **Bottom**: Pose results at different frames shown by the same camera.

5.3. Running

In a third experiment we present a challenging outdoor scenario: An athlete is running on a path and jumping over a foot path barrier. Again we use four handheld cameras which are synchronized using the audio channels. The images reveal bright sunlight, cluttered (moving) background, shadows, trees, and in some frames the other camera men

sequence	close to subject	background
climbing	0.93	2.55
halfpipe	2.34	1.28
running	4.20	15.28

Table 1. RMSE for a virtual object placed in the scene in [pixel]. For all scenes the calibration error close to the subject is minimal.

are moving in the scene.

Figure 7 shows pose results at different frames from the view of a single camera, whereas Figure 8 shows the pose results for one frame in all cameras. As can be seen, the kinematics are reasonably well computed. Note, that since the actor is running very fast, a prediction of the subject's motion between successive frames is performed (similar to [19]) to compensate for the fast movements. Additionally, the outlier feature tracks from SfM that belong to the subject are used to further constrain the motion estimation. Without the prediction, the silhouette based pose estimation procedure fails.

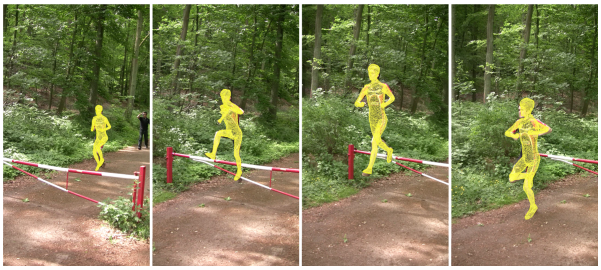

Figure 7. Pose results at different frames shown by the same camera.

In addition to the overlay images quantitative error measures for the camera calibration are summarized in Table 1. Considering that the resolution of the cameras is fairly high (1440×1080) the reprojection errors near the subject are almost as accurate as in static scenes analyzed by markerless motion capture systems before. Quantitative analysis of the pose estimation method, albeit in a static context, is presented in [19]. However, we believe that the presented results ($< 3^\circ$ angular error at the knee in a similar running sequence) can also be applied to the current setup, since accuracy of calibration and synchronization are comparable to the static scene.

6. Summary

In this paper we presented an approach for markerless motion capture. Unlike most systems our approach is suitable for cluttered indoor and outdoor environments and requires neither static nor synchronized cameras. The system is fully automatic with respect to camera setup. Two commonly required steps, namely, manual registration of camera coordinate systems and trigger based hardware synchronization of the cameras can be dropped. We have shown that the presented motion capture method works in the presence of inexact synchronization and calibration of the cameras but will obviously fail if either error becomes exceedingly large.

By using the audio channels captured along with the multi-view video, it is possible to synchronize cameras automatically even in severely noise corrupted environments. In small scenes it is sufficient to use prevailing background sounds. In large scenes, however, further knowledge about sound sources can be integrated to compensate for the slow speed of sound in air.

During the SfM based camera calibration, sparse static background geometry is generated as a by-product. It is possible to reconstruct a dense surface mesh from the point cloud, which can in turn be used to constrain the motion capture algorithm.

The input sequences (videos, projection matrices, models of subjects, and 3D background geometry) are available for scientific purposes[1].

Acknowledgments

The research was funded by the German Research Foundation (DFG), the Max Planck Center VCC (BMBF-FKZ01IMC01) and the Cluster of Excellence on Multimodal Computing and Interaction.

References

[1] A. Balan, L. Sigal, M. Black, and H. Haussecker. Shining a light on human pose: On shadows, shading and the estimation of pose and shape. In *ICCV*, 2007.

[2] M. Bray, P. Kohli, and P. Torr. Posecut: Simultaneous segmentation and 3d pose estimation of humans using dynamic graph-cuts. In *ECCV*, Lecture Notes in Computer Science, pages 642–655, Graz, May 2006. Springer.

[3] M. Brubaker, D. J. Fleet, and A. Hertzmann. Physics-based person tracking using simplified lower-body dynamics. In *CVPR*, Minnesota, 2007. IEEE Computer Society Press.

[4] T. Chan and L. Vese. Active contours without edges. *Trans. Image Processing*, 10(2):266–277, Feb. 2001.

[5] S. Dambreville, A. Yezzi, R. Sandhu, and A. Tannenbaum. Robust 3d pose estimation and efficient 2d region-based segmentation from a 3d shape prior. In *Proc. 10th European Conference on Computer Vision*, Lecture Notes in Computer Science. Springer, 2008.

[6] S. Gibson, J. Cook, T. Howard, R. Hubbold, and D. Oram. Accurate camera calibration for off-line, video-based augmented reality. In *ISMAR*, Darmstadt, Germany, 2002.

[7] R. I. Hartley and A. Zisserman. *Multiple View Geometry*. Cambridge University Press, 2000.

[8] H. Hoppe, T. DeRose, T. Duchamp, J. McDonald, and W. Stuetzle. Surface reconstruction from unorganized points. In *SIGGRAPH*, 1992.

[1] http://www.tnt.uni-hannover.de/staff/rosenhahn/

Figure 8. Pose result at one time instance overlaid into the four camera views.

[9] D. Liebowitz and S. Carlsson. Uncalibrated motion capture exploiting articulated structure constraints. In *ICCV*. IEEE Computer Society Press, 2001.

[10] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *Int. J. Comput. Vision*, 60(2):91–110, 2004.

[11] D. W. Marquardt. An algorithm for least-squares estimation of nonlinear parameters. *Journal of the Society for Industrial and Applied Mathematics*, 11(2):431–441, June 1963.

[12] T. B. Moeslund, A. Hilton, and V. Krüger. A survey of advances in vision-based human motion capture and analysis. *Computer Vision and Image Understanding*, 104(2):90–126, 2006.

[13] D. Morgan, E. George, L. Lee, and S. Kay. Cochannel speaker separation by harmonic enhancement and suppression. *Speech and Audio Processing, IEEE Transactions on*, 5(5):407–424, Sept. 1997.

[14] R. Murray, Z. Li, and S. Sastry. *Mathematical Intro. to Robotic Manipulation.* CRC Press, Baton Rouge, 1994.

[15] J.-R. Ohm and H. D. Lüke. *Signalübertragung*. Springer, 2006.

[16] A. Papoulis. *The Fourier integral and its applications*. McGraw-Hill Electronic Sciences series. McGraw-Hill, 1962.

[17] M. Pollefeys, L. V. Gool, M. Vergauwen, F. Verbiest, K. Cornelis, J. Tops, and R. Koch. Visual modeling with a hand-held camera. *IJCV*, 59(3):207–232, 2004.

[18] R. Poppe. Vision-based human motion analysis: An overview. *Computer Vision and Image Understanding (CVIU)*, 108(1-2):4–18, October 2007.

[19] B. Rosenhahn, T. Brox, and H.-P. Seidel. Scaled motion dynamics for markerless motion capture. In *Conference of Computer Vision and Pattern Recognition (CVPR)*, Minnesota, 2007. IEEE Computer Society Press.

[20] B. Rosenhahn, T. Brox, and J. Weickert. Three-dimensional shape knowledge for joint image segmentation and pose tracking. *IJCV*, 73(3):243–262, 2007.

[21] J. Shi and C. Tomasi. Good features to track. In *CVPR*, pages 593–600, 1994.

[22] H. Sidenbladh, M. J. Black, and L. Sigal. Implicit probabilistic models of human motion for synthesis and tracking. In *ECCV*, pages 784–800. Springer, 2002.

[23] S. Sinha, M. Pollefeys, and L. McMillan. Camera network calibration from dynamic silhouettes. *Proc. International Conference on Computer Vision and Pattern Recognition*, 1:195—202, June 2004.

[24] M. Siracusa, L.-P. Morency, K. Wilson, J. Fisher, and T. Darrell. A multi-modal approach for determining speaker location and focus. In *ICMI*, pages 77–80, New York, NY, USA, 2003. ACM.

[25] M. Slaney and M. Covell. Facesync: A linear operator for measuring synchronization of video facial images and audio tracks. In *NIPS 2000*, volume 13, Denver, CO, USA, Nov. 2000. MIT Press.

[26] C. Sminchisescu and A. Jepson. Generative modeling for continuous non-linearly embedded visual inference. In *Proc. Int. Conf. on Machine Learning*, 2004.

[27] T. Thormählen, N. Hasler, M. Wand, and H.-P. Seidel. Merging of unconnected feature tracks for robust camera motion estimation from video. In *CVMP*, Nov. 2008.

[28] R. Urtasun, D. J. Fleet, and P. Fua. 3D people tracking with Gaussian process dynamical models. In *CVPR*, pages 238–245. IEEE Computer Society Press, 2006.

[29] M. Vondrak, L. Sigal, and O. C. Jenkins. Physical simulation for probabilistic motion tracking. In *Conference of Computer Vision and Pattern Recognition (CVPR)*, Anchorage, 2008. IEEE Computer Society Press.

[30] M. Wand, A. Berner, M. Bokeloh, P. Jenke, A. Fleck, M. Hoffmann, B. Maier, D. Staneker, A.Schilling, and H.-P. Seidel. Processing and interactive editing of huge point clouds from 3d scanners. *Computers and Graphics*, 32(2):204–220, 2008.