Space-time Body Pose Estimation in Uncontrolled Environments

Marcel Germann
ETH Zurich
Switzerland
germann@inf.ethz.ch

Tiberiu Popa
ETH Zurich
Switzerland
tpopa@inf.ethz.ch

Remo Ziegler
LiberoVision AG
Switzerland
ziegler@liberovision.com

Richard Keiser
LiberoVision AG
Switzerland
keiser@liberovision.com

Markus Gross
ETH Zurich
Switzerland
grossm@inf.ethz.ch

***Abstract*—We propose a data-driven, multi-view body pose estimation algorithm for video. It can operate in uncontrolled environments with loosely calibrated and low resolution cameras and without restricting assumptions on the family of possible poses or motions.**

Our algorithm first estimates a rough pose estimation using a spatial and temporal silhouette based search in a database of known poses. The estimated pose is improved in a novel pose consistency step acting locally on single frames and globally over the entire sequence. Finally, the resulting pose estimation is refined in a spatial and temporal pose optimization consisting of novel constraints to obtain an accurate pose. Our method proved to perform well on low resolution video footage from real broadcast of soccer games.

***Keywords*-body pose estimation; uncontrolled environments**

I. Introduction

Pose estimation or motion capture is a fundamental problem in computer vision and graphics [13], [14] with many applications such as character animation in games and movies, controller free interfaces for games [12] and surveillance. Due to the complexity of the problem, there still does not exist a universal solution to all the applications. The solutions strongly depend on the conditions and on the constraints imposed on the setup. In general, the more constraints are opposed to the setup, the more accurately the pose estimation can be computed. In real world scenarios it is often very difficult to impose constraints on the setup. However, many practical applications are based on these scenarios. For instance, Germann et al [11] showed how accurate pose estimation can be used for high quality rendering of players from an arbitrary view-point during a sports game using only video footage already available in TV broadcasts. In addition to applications in rendering, accurate pose estimation of players during a game can also be used for bio-mechanical analysis and synthesis as well as for game statistics or even the porting of a real game play into a computer game.

In this paper, we focus on pose estimation based on unconstrained football broadcast footage. This implies several challenges to camera positions, object size and temporal coherence. Although, the pose estimation can be computed based on a multi-camera setup, there are only few cameras available, which additionally feature wide baselines. Moreover, the cameras are typically placed only on one side of the field providing limited coverage of the scene. The cameras provide high resolution images, but are usually set to be wide-angle for editorial reasons. Therefore, players typically cover only a height between 50 and 200 pixels. Furthermore, the motion of the players can be very complex and, especially in contact sports like football, there is a lot of occlusion.

We propose a data-driven pose estimation algorithm that can operate in an uncontrolled environment with loosely calibrated cameras, low resolution players and in presence of occlusions. Our algorithm can use as little as only two cameras to estimate the pose. No restricting assumption is made on the family of possible poses or motions. By using temporal coherence for the initial pose estimation as well as pose refinement the user interaction is limited to few clicks inverting arms and legs in failure cases.

Many of the state of the art algorithms in pose estimation rely on tracking or segmenting the image in 2D and using calibration information to extrapolate the skeleton to 3D [4], [20]. These approaches work well for high resolution footage, but due to lack of information, they often fail on low resolution images and are sensitive to external lighting conditions. Our algorithm works in completely uncontrolled outdoor setups with low resolutions, since it only relies on coarse silhouettes and coarse calibrations.

Similar to Germann et al. we use a database of poses and silhouette comparison to extract pose candidates in 2D and use camera calibration information to compute the corresponding 3D skeleton. In contrast to their method, we first perform a novel time consistent silhouette based search in the database to extract the closest database candidate with temporal coherence. An additionally applied novel time consistency step is leading to the initial pose estimaton. Because the exact real pose is generally not in the database, this will only result in a closest match, but not in an accurate pose. Therefore, we developed a novel space-time optimization technique that leverages the temporal information to automatically compute the accurate 3D pose.

The main contributions of our paper are:

- A time consistent silhouette based database pose look-up providing an initial pose estimation
- Local and global consistency check to improve initial pose estimation

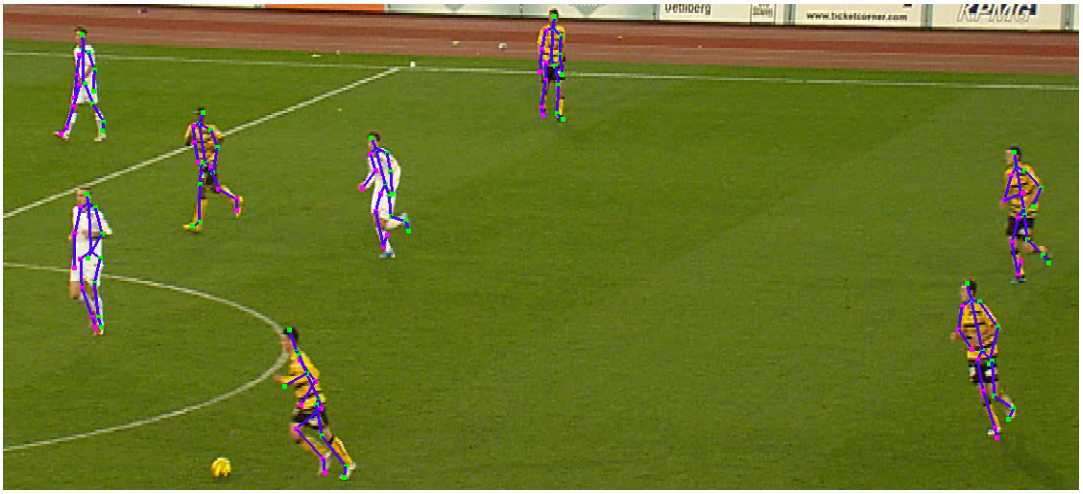


Figure 1. Estimated poses in a soccer game. The image shows a projection of the 3D skeletons into a source camera.

- A space-time pose optimization based on novel constraints

Instead of learning a statistical model for the skeleton, our algorithm directly uses a database of poses. This has two advantages. Firstly, such a data-driven method allows to easily add new pose sequences to adapt to new setups or previously unknown poses. Secondly, there is less statistical bias to more common poses, since the method simply searches for the closest pose in the database. Using a database with anthropometrically correct data will always result in a plausible pose for the initial estimation.

II. Related Work

Many current commercially available motion capture systems [22] typically use optical markers placed all over the body to track the motion over time. These systems are very accurate and can capture all kinds of body poses as well as facial expressions. However, they are invasive and work under controlled environment. Therefore, they are only suitable for a specific range of applications.

Markerless motion capture methods have received a lot of attention in the last decade [13], [14]. Based on the type of footage used, the markerless pose reconstruction (or motion capture) problem can be roughly categorized into two groups [24]: using video sequences from one camera or using footage from multiple calibrated cameras. Pose estimation from monocular video sequences [2], [3], [24], [17], [1], [18] can be more convenient for some applications as it imposes less restrictions to the user, but it has an inherent depth ambiguity. This ambiguity can be solved using structure from motion approaches, a very difficult problem in vision [13], [14]. Structure from motion algorithms typically rely on high-resolution scenes containing a lot of detail which we typically do not have in our scenario. Efros et al. [9] also process soccer footage. Even though their work focuses more on action detection, they showed that even on low resolution data a rough 2D pose can be estimated.

Another major challenge in pose estimation are occlusions. If the footage comes from a single camera it is very difficult to resolve them. Using multiple cameras increases the probability to have an unoccluded view of the same subject. The higher the spatial coverage by cameras is, the fewer ambiguities remain. Moreover, sport broadcasts already use multiple cameras on the field. Therefore, we can leverage this information to compute a more accurate 3D pose estimation.

Most methods for multiple views 3D pose estimation use tracking algorithms to reconstruct the pose at time t from the pose at time $t-1$ [4]. The tracking can be done either using optical flow [4] or stereo matching [6]. These methods can provide very accurate pose estimation, but they generally work in a controlled environment, require larger number of high-resolution cameras (usually at least four) and good spatial coverage of the scene (usually circular coverage) to resolve ambiguities due to occlusions.

Other methods [21], [8], [23] construct a proxy geometry either using multi-view silhouettes or multi-view stereo. The skeleton is then fitted into this geometry. These methods provide very good results, but impose restrictions on the setup. They require a carefully built studio setup, many high resolution cameras and very good spatial coverage.

Another class of algorithms is based on image analysis and segmentation [15], [10]. These algorithms use machine learning methods to discriminate between body parts. This analysis generally requires high resolution footage, which is not available in our setup.

Our setup is more flexible, but entails a more restrictive quality: we are constraint to only two to three cameras that are generally placed only on one side of the field, have very large baselines and weak calibrations. Also, since the image

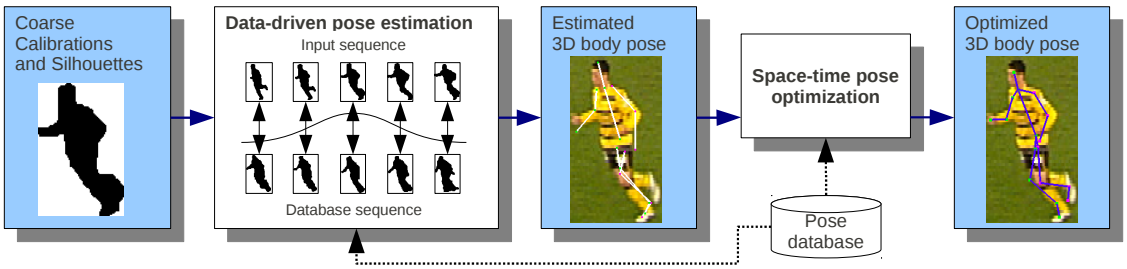


Figure 2. Algorithm overview.

generally covers a large part of the field, each individual player has a very low resolution. The closest to our approach is the method by Germann et al [11]. This method also uses a silhouette based database search to retrieve the 3D pose. However, it is based on a single frame only, without any consistency check. Furthermore, the result only consists of poses already available in the database, which is often not the case. Therefore, the poses will be incorrect and manual correction is required for all poses in all views and all frames making this very tedious and unfeasible for sequences. Our algorithm matches sequences instead of a single frame, insures consistency for the initial guess and employs a novel energy term to compute the final 3D pose.

III. Overview

Our algorithm consists of two steps as illustrated in figure 2. In the first step, the algorithm extracts 2D poses for each individual camera view using a spatial-temporal silhouette matching technique, yielding a triangulated 3D pose guess. This pose detection is inherently prone to ambiguities, namely left right flips of symmetrical parts. Although the skeleton matches the silhouettes quite well, the arms or legs of the player can still be flipped. Due to occlusions and low resolution, these ambiguities are sometimes very difficult to spot even for the human eye. Therefore, we employ an optical flow based technique to detect the cases where flips occur, and correct them to obtain a consistent sequence. It is important to note that optical flow is in such setups not reliable enough for tracking the entire motion of a players body parts over an entire sequence, but it can be used for local comparisons as shown by Efros et al. [9].

However, in general, no pose from the database will match the actual pose exactly. As a consequence, in the second part of the algorithm, this initial 3D pose is refined by an optimization procedure, which is based on spatio-temporal constraints. The resulting optimized 3D skeleton matches the silhouettes from all views and features temporal consistency over consecutive frames.

IV. Initial Pose Estimation

The initial pose estimation is computed by first retrieving the 2D pose from each player and each camera view using a novel space-time data-driven silhouette based search. Once we find the 2D poses for every player in every camera, we can use the calibration information from the cameras to lift the 2D joints to 3D. We compute the 3D location of each joint by intersecting the rays corresponding to each 2D joint in each camera view. The rays will not intersect exactly, therefore we choose the closest point to these rays in least-squares sense. From this we get a triangulation error E_t and an initial camera shift as described by Germann et al. [11].

We represent the 3D skeleton of a pose S in angle space. Every bone i is represented relative to its parent bone using two angles α_i and β_i as well as the length l_i of the bone. The root bone is defined by its orientation given by three angles α_0, β_0, γ_0 and by a global position p_0. The joint positions $\mathbf{j}_i$ in the 3D Euclidian space can easily be computed from this representation and vice-versa.

A. Pose Database Construction

A large database that samples the entire range of human motion is important for our algorithm and is very difficult to create manually. Therefore, we use the CMU motion capture database [7]. A template mesh rigged with the same skeleton is deformed using linear blend skinning to match the pose of the database pose. From this virtual snapshots are taken and the silhouette is extracted. This way we created a database of around 20000 silhouettes.

Unfortunately, the CMU database has only a limited number of types of poses, mostly from running and walking sequences. Therefore, we manually added a set of 900 silhouettes from several soccer scenes. This is significantly fewer than the ones generated automatically, but enough to enlarge the span of example poses to obtain good results. It is important to note, that the added example poses were not taken from the same sequences as we used to fit the poses. The database could continuously be enlarged by new generated poses, resulting in a better initial pose estimation.

B. 2D Pose Estimation

Similar to Germann et al. [11] we assume as an input a coarse binary silhouette mask for each player as well as coarse camera calibrations. We compare these silhouettes against the silhouettes from the database using the technique

presented by Germann et al., that computes the quality of a match between the input silhouette and a database silhouette on a fix raster size (grid with height=40 and width=32) that is fitted to the segmentation.

The silhouette extraction extends the algorithm presented by Germann et al. by leveraging temporal information. Instead of relying on a single frame matching, our approach considers a weighted sum of differences over a sequence. The resulting pixel error $E_q(s)$ of the binary input silhouette image I with index t based on the silhouette image I'_s with index s from the database is evaluated as follows:

$$E_q(s) = \sum_{i \in \{-\frac{n}{2}, .., \frac{n}{2}\}} \theta_s(i) \frac{1}{|P|} \sum_{p \in P} |I_{t+i}(p) - I'_{s+i}(p)|. \quad (1)$$

n is the filter window size, and P is the set of all raster positions where the corresponding pixel is in both images not possibly occluded, i.e., part of another players silhouette. The weights $\theta_s(i)$ describe a normalized Gaussian function with the center around s. For $I'_{s+i}(p)$ not included in the database, $\theta_s(i)$ is set to 0 before the normalization. Comparing sequences instead of single images, does not only add temporal coherence resulting in smooth motions, but also improves pose estimation. Even image parts occluded over few frames can be fitted more robustly. In general, this approach helps to prevent matching a silhouette that is similar but originated from a completely different pose. This is depicted in figure 3, which also shows a direct comparison of our initial pose estimation to the pose estimation in the work of Germann et al.

Using this energy function, we search for each camera view for the best two pose hypotheses and select the best combination of those by choosing the lowest resulting triangulation error E_t.

C. Pose Consistency

The 2D pose detection step relies on silhouette matching, and therefore, is prone to ambiguities. Given a silhouette and an initial 2D pose for it taken from the database, we can not decide if the labellings of left/right in arms and legs are correct. Figure 4(a) shows an example silhouette with two possible labellings for the legs. The possible position of the right knee is marked by a blue diamond. This view is from the left camera in the schema in figure 4(b). The same subject in the same frame but in the other camera shows the silhouette in 4(c), again with the two possible labellings of the legs. Therefore we have four possible positions in 3D for the right knee after lifting into 3D. They are shown in figure 4(b). If the right knee falls on one of the positions marked with a star, the left knee will fall on the other star. If the right knee falls on one of the positions marked with a circle, then the left knee will fall onto the other circle. Let the circles be the correct positions of the knees, then we can have two different failures: either the knees are just wrongly

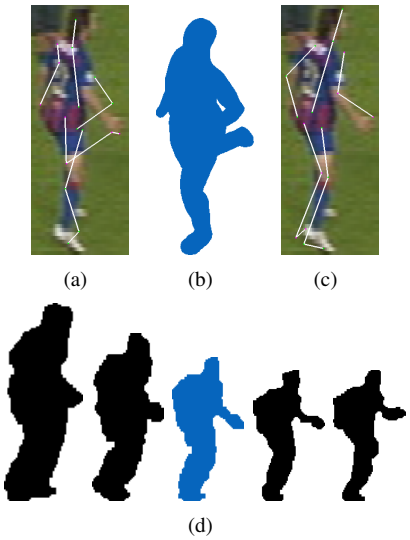


Figure 3. (a) Estimated 2D pose by comparing just the current frame to the database as in Germann et al. [11]. (b) The found database item for the single frame comparison. (c) Estimated 2D pose by comparing sequences of silhouettes. (d) The found database sequence with the corresponding pose in the middle.

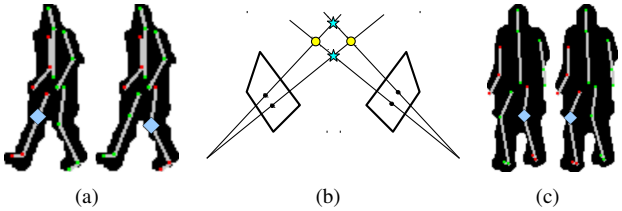


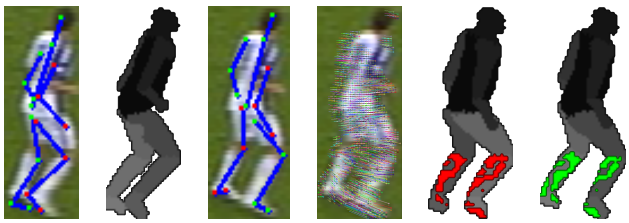
Figure 4. Example for pose ambiguities. (a) Possible labellings in first camera. (b) Schematic view from the top to illustrate the two possible positions of the knees. (c) Possible labellings in the second camera.

labeled in 3D but at the correct positions or the knees are at wrong positions (the stars).

Without additional information we cannot decide in such a situation which positions are correct, i.e., select the only one correct out of the four possibilities – especially when only two cameras are available. A possible approach to disambiguate the flipped cases would consist of checking all possible combinations and keep the anatomically possible ones. However, it is possible that several configurations of flips yield anatomically correct poses.

To correctly solve these ambiguities, we propose a two step approach: first, the local consistency between each pair of consecutive 2D frames is resolved, resulting in an entire sequence of temporally consistent 2D poses. Second, the entire sequence is resolved globally.

1) Local Consistency: The goal of this step is to make sure that the 2D pose recovered from a camera at frames k (figure 5(a) and 5(b)) and $k+1$ (figure 5(c)) are consistent, i.e., there are no flips of arms or legs between consecutive frames. In other words, if a pixel at frame k belongs to the right leg, it should belong to the same leg in the frame $k+1$



(a) (b) (c) (d) (e) (f)

Figure 5. Local consistency: (a) Previous frame and (b) fitted mesh. (c) Wrongly assigned legs in current frame. (d) Optical flow. (e) Fitted mesh in current frame with correct and wrong matches labelled as green and red, respectively. (f) Error of the flipped (correct) legs in the current frame.

as well. To assure this assumption, we assign to each pixel in both color images I_{C_k} and $I_{C_{k+1}}$ a corresponding bone, and we compute the optical flow [5] (figure 5(d)) between the frames. The underlying idea is that a pixel in frame k and its corresponding pixel in frame $k+1$, computed using optical flow, should be assigned to the same bone. Otherwise there could be a flip as shown in figure 5(c). Therefore, we compute this association for all combinations of possible labellings, compute the consistency of the pixels, and select the most consistent label configuration for the second frame. To make the approach more robust in respect to optical flow errors, we only consider pixels with good optical flow and where the corresponding pixel labels in both frames are of the same bone type, i.e., either both arms or both legs. For instance, if a pixel p belongs to the left arm in frame k and to the torso in frame $k+1$, it is most likely due to an inaccurate optical flow based on occlusion and we can thus omit this pixel. If the pixel belongs to a different bone of the same type it is a strong indication of a flip. We employ a voting strategy to select the optimal flip configuration.

To do this, each pixel has to be assigned to its corresponding bone. A naive assignment based on the distance to the bone is incorrect, since it does not take into account occlusions. Therefore, we construct the 3D pose using the information from all the cameras as described in section IV. Again, we use a template mesh deformed and rendered for all possible flips in all cameras using color coding for all the bones. Thus, the pixel assignment is a simple lookup providing an accurate assignment despite of self occlusion. Figure 5(e) shows an initial wrong labeling with the found correct (green) and wrong (red) pixel assignments. Figure 5(f) shows the correct labeling.

This resolves most of the flips of arms or legs. For failure cases, the user can change the flipping for all subsequent frames of a sequence with one mouse-click. This is the only user interaction in our system and for a view of a player takes only about 1 click per 10 frames.

2) Global Consistency: After the local consistency step, all consecutive frames should not have flips between them which means that the entire sequence is consistent. There is still the possibility that the entire sequence is flipped the wrong way. However, this is a simple problem as we only have a binary predicate to apply for the entire sequence. Therefore, the global consistency is checked for the possible global labelings of the arms and the possible global labelings of the legs. The final labeling is selected by choosing the labeling combination that minimizes the following error term:

$$E_g = \lambda_{DB} E_{DB} + \lambda_t E_t \tag{2}$$

This is a weighted sum with constant parameters λ_{DB} and λ_t. E_{DB} ensures that the selected labeling/flipping results in plausible poses along the sequence. It penalties for the distance to the closest pose P in the database:

$$E_{DB} = \min_P \frac{1}{2|J|} \sum_{i=0}^{|J|} (\alpha'_i - \alpha_i)^2 + (\beta'_i - \beta_i)^2 \tag{3}$$

where α and β are the joint angles of the triangulated joint positions J. α' and β' are the ones of the database pose P. Since the database contains only anthropometrically correct poses, this penalties for non-plausible poses.

Figure 6. Comparisons of the pose before (left) and after (right) the pose optimization.

V. Pose Optimization

The best 3D poses computed by the pose estimation are still limited to fit in each view to a pose of the database. The database is only a subset of all possible poses, and therefore, often does not contain the accurate solution. We developed a new optimization method to retrieve a more accurate pose as shown in figure 6. To guide our optimization method, we combine several spatial and temporal energy functions and minimize them using an optimization method.

A. Energy Function

The energy function is based on our representation of the skeleton S described in section IV. All the parameters besides the bone length are variable per frame. The lengths are also variable but stay the same over the entire sequence and are initialized as the average of the local lengths of all frames. This automatically introduces an anthropometric constraint, since bones should not shrink or grow over time. Another nice property of the chosen skeleton representation is that it significantly reduces the number of variables. In order to cope with calibration errors, we also optimize for the two dimensional shift vector given per subject, camera and frame.

We define our energy functional per frame and subject as the following weighted sum of error terms:

$$E(S) = \omega_s E_s + \omega_f E_f + \omega_{DB} E_{DB} + \omega_{rot} E_{rot} + \omega_p E_p + \omega_l E_l \quad (4)$$

Silhouette matching term E_s: The bones of the correct 3D skeleton should project onto the 2D silhouette in all cameras. The error term E_s penalizes the joint positions whose 2D projections are outside the silhouettes:

$$E_s = \frac{1}{|C||J_+|} \sum_{c \in C} \sum_{\mathbf{j} \in J_+} EDT_c(P_c(\mathbf{j})), \quad (5)$$

where C is the set of all cameras that cover a silhouette of this subject. J_+ is the union of the set J of all joints and the points that are exactly in the middle of a bone. The normalized Euclidean distance transform EDT returns for every 2D point in the camera image the distance to the closest point inside the silhouette divided by the larger side of the silhouettes bounding box. This normalization is important to make the error independent of the size of the subject in the camera image which may vary according to the zoom. $P_c(j)$ is the projection to transform the 3D joint $\mathbf{j}$ into camera space taking into account the camera shift.

Silhouette filling error term E_f: Although the silhouette matching term E_s penalizes joints outside the silhouette, there is no restriction on them for where to be placed inside the silhouette. The filling error term E_f prevents them from just shrinking together somewhere inside the torso and especially makes sure that there are also joints in all the extremities:

$$E_f = \frac{1}{|C||R|} \sum_{c \in C} \min_{\mathbf{j} \in J} \sum_{r \in R} dist(P_c^{-1}(r), \mathbf{j}), \quad (6)$$

where R is the set of all grid points from section IV-B that are inside the silhouette. $P_c^{-1}(r)$ transforms such a grid point from camera space of camera c into a ray in world space while $dist()$ describes the distance of a ray to a joint.

Distance to database pose E_{DB}: This was already defined in section IV-C2. It ensures that the final 3D pose is kinematically possible (e.g., the knee joint bends the right way) by taking into advantage the database of correct poses. It implicitely adds anthropometric constraints to our optimization.

Smoothness error terms E_{rot} and E_p: Human motion is generally smooth such that the skeletons of adjacent frames should be similar. This enables us to introduce temporal coherence to the pose optimization. Therefore, E_{rot} penalizes large changes of the internal angles of the skeleton of consecutive frames and E_p penalizes large motion:

$$E_{rot} = \frac{1}{2|J|} \sum_{i=0}^{|J|} (\alpha_i' - \alpha_i)^2 + (\beta_i' - \beta_i)^2 \quad (7)$$

$$E_p = |\mathbf{p}_0 - \mathbf{p}_0'| \quad (8)$$

Table I
THE PARAMETER VALUES THAT WE USED FOR ALL OUR RESULTS, COMPUTED USING OUR AUTOMATIC PARAMETER TUNING SYSTEM

Param.	ω_s	ω_f	ω_{db}	ω_{rot}	ω_p	ω_l	λ_{DB}	λ_t
	9	15	0.05	0.1	1	1	0.15	0.3

where α' and β' are the corresponding angles of the same subject in the previous frame and $\mathbf{p}_0'$ is the global position of the root joint in the previous frame. We also constraint the rotation of the root bone in a similar way, which we omitted here for simplicity.

Length error term E_l: The initialization of the bone lengths is already a good approximation, when handling a sequence of frames. Therefore, we try to keep the optimized pose close to these lengths:

$$E_l = \frac{1}{|J|} \sum_{i=0}^{|J|} (l_i - \hat{l}_i)^2 \quad (9)$$

where l_i is the final bone length and $\hat{l}_i$ is the initial bone length.

B. The Optimization Procedure

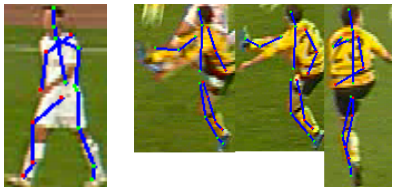
To minimize the energy term in equation 4, we employ a local optimization strategy where we iteratively optimize the variables one by one by performing line search along randomly picked directions [19]. For each variable we select 10 random directions for optimization and we perform 20 global iterations. Due to the inherent non-smooth nature of our objective functions, this method performed better in practice than Levenberg-Marquardt [16].

Figure 6 illustrates the effect of the optimization procedure. The leftmost example shows the influence of the silhouette filling error term: The arm of the player can be brought up or down to reduce the silhouette matching error term, but the silhouette filling error term is only reduced when moving the arm up. Figure 6 shows a clear improvement over the method by Germann et al. [11] which did not include a pose optimization at all and where each pose had to be corrected manually.

VI. Results

We evaluated our system on four sequences of TV-footage from real soccer games with two or three cameras, yielding roughly 1500 poses to process. A subset of the results are shown in figures 1 and 8. In addition, the accompanying video shows also renderings from arbitrary view-points using articulated billboard rendering [11] based on the pose estimations of our algorithm.

Each row in figure 8 shows a set of consecutive poses and each item shows the image of the respective player in all the available cameras. Even with only two cameras and very low resolution images, our algorithm can retrieve good poses in most cases.



(a) (b)

Figure 7. Failure cases. (a) The arms are too close to the body and could not be positioned correctly. (b) A pose that is too far from the database and could not be estimated correctly.

For the optimization functions in equations (4) and (2) we used the parameters shown in table I for all our results. They were found by the following parameter tuning procedure. We annotated manually the 2D poses in two scenes. Then the algorithm was run and the results were automatically compared with the manual annotations. Using this as an error function and the parameters as variables allows for an automatic parameter optimization.

Our pose estimation algorithm takes about 40 seconds per player per frame in a two camera setup and about 60 seconds for a three camera setup. We implemented a parallel version that runs a thread for every player. On an 8 core system this gave a speedup of roughly a factor of 8.

Note that the initial pose estimation does not depend on the pose estimation of the previous frame. Thus, there is no drift and the process can recover from bad pose guesses.

Limitations and Future Work.: Although our current approach leads to good results in many cases, it can fail due to the lack of information provided by a binary silhouette only, particularly when the arms are too close to the body as illustrated in figure 7(a). I.e., several poses can have very similar binary silhouettes. Thus, only using silhouette information is not sufficient to disambiguate the poses. Incorporating optical flow into the optimization procedure could resolve such ambiguities in the future.

Furthermore, the results of our algorithm greatly depend on the pose database. A good database will have a wide range of motions as well as a wide range of views such that the initial guess is close to the correct pose. Figure 7(b) shows an example where there is no similar pose in the database and thus the pose estimation fails. In the future, we would like to find an automatic criterion to quantify a good match of a pose. Good poses could then automatically be added to the database enlarging the space of possible poses.

Another important prior that can be leveraged further is the kinematic information of the human skeleton. Our method already uses some implicit anthropometric constraints, but specific constraints on joint angles could improve this even more.

VII. Conclusions

This paper addresses the problem of multi-view pose estimation of entire sequences in uncontrolled environments. More specifically, we aimed at pose reconstruction using video footage from TV-broadcasts of soccer games. A typical setup can have as few as two or three cameras, which are placed on one side of the field only, with wide baselines and inaccurate camera calibrations as a consequence. Furthermore, the usual wide angle shots result in low resolutions of the players.

To compute accurate 3D poses in such an uncontrolled environment, we rely on a rich database of poses and use the temporal coherence of human motion as a strong prior. In this work, we proposed a data-driven pose estimation algorithm that works in two stages. First, we introduced a novel spatio-temporal search to retrieve a good initial pose based on silhouette matching. A mesh based consistency check resolves for unwanted flips of the limbs, avoiding local minima in the optimization step. The initial estimation is improved using a novel optimization technique that combines spatial and temporal constraints to yield the final pose.

VIII. Acknowledgements

The data is courtesy of Teleclub and LiberoVision. This project is supported by a grant of CTI Switzerland and NSERC. We thank Thomas Oskam for the help with rigging, Tobias Pfaff and Tanja Käser for annotating ground truth data.

References

[1] A. Agarwal and B. Triggs. Recovering 3d human pose from monocular images. *PAMI*, 2006.

[2] M. Andriluka, S. Roth, and B. Schiele. Pictorial Structures Revisited: People Detection and Articulated Pose Estimation. *CVPR*, 2009.

[3] M. Andriluka and S. R. B. Schiele. Monocular 3d pose estimation and tracking by detection. *ECCV*, 2010.

[4] L. Ballan and G. M. Cortelazzo. Marker-less motion capture of skinned models in a four camera set-up using optical flow and silhouettes. In *3DPVT*, 2008.

[5] J.-Y. Bouguet. Pyramidal implementation of the lucas kanade feature tracker: Description of the algorithm. Technical report, Intel Corp., Micropr. Research Labs, 1999.

[6] C. Choi, S.-M. Baek, and S. Lee. Real-time 3d object pose estimation and tracking for natural landmark based visual servo. In *IROS*, 2008.

[7] C. G. L. M. Database, 2010. http://mocap.cs.cmu.edu.

[8] E. de Aguiar, C. Stoll, C. Theobalt, N. Ahmed, H.-P. Seidel, and S. Thrun. Performance capture from sparse multi-view video. In *SIGGRAPH*, 2008.

Figure 8. Result sequences with all camera views shown per frame.

[9] A. A. Efros, A. C. Berg, G. Mori, and J. Malik. Recognizing action at a distance. In *ICCV*, 2003.

[10] V. Ferrari, M. Marin-Jimenez, and A. Zisserman. Progressive search space reduction for human pose estimation. In *CVPR*, 2008.

[11] M. Germann, A. Hornung, R. Keiser, R. Ziegler, S. Würmlin, and M. Gross. Articulated billboards for video-based rendering. *Eurographics*, 2010.

[12] Kinect, 2010. http://www.xbox.com/en-US/kinect.

[13] T. B. Moeslund and E. Granum. A survey of computer vision-based human motion capture. *CVIU*, 2001.

[14] T. B. Moeslund, A. Hilton, and V. Krüger. A survey of advances in vision-based human motion capture and analysis. *CVIU*, 2006.

[15] G. Mori. Guiding model search using segmentation. In *ICCV*, 2005.

[16] J. Mor. The levenberg-marquardt algorithm: Implementation and theory. In *Lecture Notes in Mathematics*, volume 630. 1978.

[17] D. Ramanan, D. Forsyth, and A. Zisserman. Strike a pose: tracking people by finding stylized poses. In *CVPR*, 2005.

[18] D. Ramanan, D. Forsyth, and A. Zisserman. Tracking people by learning their appearance. *PAMI*, 2007.

[19] J. Schreiner, A. Asirvatham, E. Praun, and H. Hoppe. Inter-surface mapping. In *SIGGRAPH*, 2004.

[20] A. Shahrokni, T. Drummond, and P. Fua. Markov-based Silhouette Extraction for Three–Dimensional Body Tracking in Presence of Cluttered Background. In *BMVC*, 2004.

[21] C. Theobalt, E. de Aguiar, M. A. Magnor, H. Theisel, and H.-P. Seidel. Marker-free kinematic skeleton estimation from sequences of volume data. In *VRST*, 2004.

[22] Vicon, 2010. http://www.vicon.com.

[23] D. Vlasic, I. Baran, W. Matusik, and J. Popović. Articulated mesh animation from multi-view silhouettes. In *SIGGRAPH*, 2008.

[24] B. Zou, S. Chen, C. Shi, and U. M. Providence. Automatic reconstruction of 3d human motion pose from uncalibrated monocular video sequences based on markerless human motion tracking. *Pattern Recognition*, 2009.