Motion Capture from Dynamic Orthographic Cameras

Magnus Burenius, Josephine Sullivan, Stefan Carlsson
KTH CSC/CVAP
S-100 44 Stockholm, Sweden
{burenius,sullivan,stefanc}@csc.kth.se

Abstract

We present an extension to the scaled orthographic camera model. It deals with dynamic cameras looking at far away objects. The camera is allowed to change focal length and translate and rotate in 3D. The model we derive says that this motion can be treated as scaling, translation and rotation in a 2D image plane. It is valid if the camera and its target move around in two separate regions that are small compared to the distance between them.

We show two applications of this model to motion capture applications at large distances, i.e. outside a studio, using the affine factorization algorithm. The model is used to motivate theoretically why the factorization can be carried out in a single batch step, when having both dynamic cameras and a dynamic object. Furthermore, the model is used to motivate how the position of the object can be reconstructed by measuring the virtual 2D motion of the cameras. For testing we use videos from a real football game and reconstruct the 3D motion of a footballer as he scores a goal.

Figure 1. Four different frames from a pan tilt camera following a football player. This can be approximated as a dynamic orthographic camera whose 3D rotation can be approximated as 2D translation. As the camera rotates the lines on the pitch looks as if they are being translated.

1. Introduction

This paper discusses the scaled orthographic camera model, a special case of the general projective camera [2]. The scaled orthographic camera is an idealized model that is valid for most cameras when looking at far away objects. Then the projection can be approximated as parallel projection with an isotropic scaling factor. For instance this is true for a lot of footage from outdoor sports like: track and field, football and downhill skiing.

Assuming cameras to be orthographic simplifies calibration as well as 3D reconstruction and motion capture, i.e. 3D motion reconstruction of dynamic objects, typically humans. This is relevant when doing motion capture at large distances, i.e. outside a studio. If the image positions of some dynamic points have been measured in two or more cameras the Affine factorization algorithm [10, 2] can be used to reconstruct the points and the cameras in 3D. In some special cases, like linear low rank or articulated motion, it is even possible to do this with a single camera [2, 6, 11, 13, 14, 7]. However, in this paper we focus on general dynamic objects and multiple dynamic cameras.

In section 2 we present an extension of the scaled orthographic camera. This extended model concerns the dynamics of the camera. Specifically it says that the 3D motion of a dynamic orthographic camera can often be treated as 2D motions in an image plane. This holds if the translation of the camera and the object it looks at are small relative to the distance between them. Then the 3D rotation of the camera can be approximated as 2D rotation and 2D translation. This is a natural extension to orthographic cameras since they are assumed to view distant objects.

We then discuss two applications of this result in section 3. They both regard motion capture using the factorization algorithm which is discussed in section 3.1. The first is about applying factorization to multiple frames in a single batch step for increased accuracy and robustness, as

opposed to doing it independently for each frame. We motivate theoretically why this can be done even if both the object and cameras are dynamic in section 3.2.

The second application is about computing the position of a reconstructed dynamic object (section 3.3). This depends on the absolute motion of the camera, which is difficult to measure for a general projective camera. However, using the result that the motion of orthographic cameras can be treated in 2D this process is simplified.

The related topic of detecting the isotropic scaling, 2D translation and 2D rotations that relates a pair of images is briefly discussed in section 3.3.1. Videos from a real football game are used for testing the motion capture in section 4.

Figure 2. A pan-tilt camera. This can be approximated as a dynamic orthographic camera whose rotation can be approximated as translation. Two different frames are translated and the overlapping parts are blended. The fit is good as can be seen by examining the lines on the pitch.

2. Dynamic Orthographic Camera

Assumptions Consider a camera looking at far away objects. The objects move around and the camera is free to rotate, translate and zoom to follow them. Let $p = (p_x, p_y, p_z)^T$ be the target point of the camera, which controls its pan-tilt rotation. Let the angle θ parametrize a possible roll rotation around the viewing axis. Assume the translational motion of both the camera and its target point are small relative to the distance between the camera and the objects. This can be modeled by placing the camera at position $t = (t_x, t_y, t_z - d)^T$ and letting $d \to \infty$. See figure 4 for the setup of the camera.

Proposition Then the camera can be treated as an orthographic camera and its dynamics can be treated in 2D, i.e. as scaling, rotation and translation in the image plane. Projection of a point in 3D (x, y, z) to the image (u, v) can then be written:

$$\begin{pmatrix} u \\ v \end{pmatrix} = f' \begin{pmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{pmatrix} \begin{pmatrix} x - p_x \\ y - p_y \end{pmatrix} \qquad (1)$$

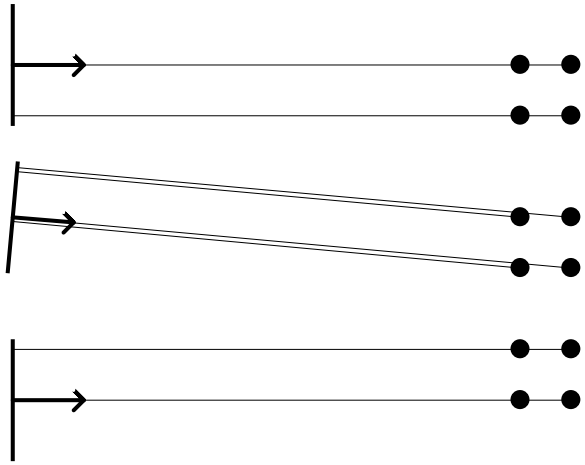

Figure 3. An orthographic camera rotating to follow an object. The top figure shows the original frame with the camera to the left and an object to the right. In the middle figure the object has moved and the camera is rotated to follow it. In the bottom view the object has also moved but the camera is instead translated to follow it. The projected image is approximately the same. If the object is infinitely far away the result will be exactly the same. The rotation of the camera can therefore be approximated as translation.

where f' controls the zooming of the camera and is related to the focal length of the camera, which is assumed to be equal for both image axes. It is also assumed that the image coordinates are given in a coordinate system with the principal point at the origin.

Thus the 3D dynamic of this orthographic camera can be treated in 2D. The roll rotation is a rotation in the image plane and a change of focal length results in a uniform scaling of the image. Both the translation of the camera and its pan-tilt rotation results in a 2D translation in the image. However, this image translation only depends on the target point p. It does not matter how the camera changes to follow it. It could change by translation or pan-tilt rotation. The resulting image translations are equivalent. The dynamics of the camera can thus be seen as isotropic scaling, 2D rotations and 2D translations in a static image plane.

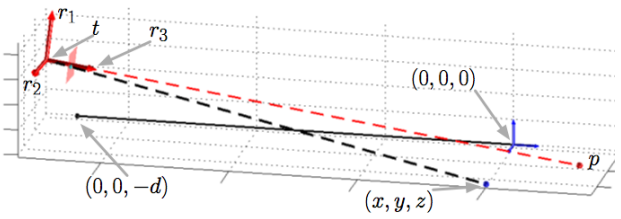

Figure 4. The setup of the camera at position t looking at p and a general point (x, y, z) which is projected by the camera.

Derivation We now derive the result expressed by equation 1 formally:

$$\begin{aligned}
\text{Camera position:} \quad & t = t_0 + \Delta t && (2)\\
& t_0 = -d(0,0,1)^T = -de_3 && (3)\\
& \Delta t = (t_x, t_y, t_z)^T && (4)\\
\text{Target point:} \quad & p = (p_x, p_y, p_z)^T && (5)\\
& n = p - \Delta t = (n_x, n_y, n_z)^T && (6)
\end{aligned}$$

Denote the rotation matrix of the camera as R. Without loss of generality the rotation can be decomposed into a pan-tilt rotation R_{PT}, which determines the viewing axis, and a roll rotation around this axis R_R:

$$\begin{aligned}
R(\theta, t, p) &= R_R(\theta) R_{PT}(t, p) && (7)\\
R_R(\theta) &= \begin{pmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{pmatrix} && (8)\\
R_{PT}(t, p) &= \begin{pmatrix} r_1^T \\ r_2^T \\ r_3^T \end{pmatrix} && (9)
\end{aligned}$$

See the appendix (6) for the full expression of $R_{PT}(t, p)$. If the camera is assumed to have square pixels and the principal point at the origin its calibration matrix is:

$$K = \begin{pmatrix} f & 0 & 0 \\ 0 & f & 0 \\ 0 & 0 & 1 \end{pmatrix} \qquad (10)$$

where f is the focal length of the camera. To derive an orthographic camera we let the camera move infinitely far away from the point it looks at, $d \to \infty$, while at the same time zoom in to have a constant scaling of the objects. Let $f = f'd$:

$$\begin{aligned}
K &= \begin{pmatrix} f & 0 & 0 \\ 0 & f & 0 \\ 0 & 0 & 1 \end{pmatrix} = \begin{pmatrix} f'd & 0 & 0 \\ 0 & f'd & 0 \\ 0 & 0 & 1 \end{pmatrix} = \\
&= d \underbrace{\begin{pmatrix} f' & 0 & 0 \\ 0 & f' & 0 \\ 0 & 0 & 1 \end{pmatrix}}_{K'} \underbrace{\begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & \frac{1}{d} \end{pmatrix}}_{D} = \\
&= K'D \qquad (11)
\end{aligned}$$

The last equality holds since D and dD belongs to the same equivalence class in projective geometry. Note that D and R_R commute, i.e. $DR_R = R_R D$. As the camera moves infinitely far away we get the following camera matrix:

$$\begin{aligned}
P_\infty &= \lim_{d\to\infty} P = \lim_{d\to\infty} KR(I|-t) = \\
&= \lim_{d\to\infty} K'DR_R R_{PT}(I|-t) = \\
&= \lim_{d\to\infty} K'R_R D R_{PT}(I|de_3 - \Delta t) = \\
&= \lim_{d\to\infty} K'R_R D R_{PT} \begin{pmatrix} I & de_3 \end{pmatrix} \begin{pmatrix} I & -\Delta t \\ 0^T & 1 \end{pmatrix} = \\
&= \lim_{d\to\infty} K'R_R D \begin{pmatrix} r_1^T & r_{1,z}d \\ r_2^T & r_{2,z}d \\ r_3^T & r_{3,z}d \end{pmatrix} \begin{pmatrix} I & -\Delta t \\ 0^T & 1 \end{pmatrix} = \\
&= \lim_{d\to\infty} K'R_R \begin{pmatrix} r_1^T & r_{1,z}d \\ r_2^T & r_{2,z}d \\ \frac{r_3^T}{d} & r_{3,z} \end{pmatrix} \begin{pmatrix} I & -\Delta t \\ 0^T & 1 \end{pmatrix} \qquad (12)
\end{aligned}$$

In the final line of equation 12 only one of the matrices depends on d. The limit value of this matrix is computed in the appendix (6). The result is:

$$\begin{aligned}
P_\infty &= K'R_R \begin{pmatrix} 1 & 0 & 0 & -n_x \\ 0 & 1 & 0 & -n_y \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} I & -\Delta t \\ 0^T & 1 \end{pmatrix} = \\
&= K'R_R \begin{pmatrix} 1 & 0 & 0 & -n_x - t_x \\ 0 & 1 & 0 & -n_y - t_y \\ 0 & 0 & 0 & 1 \end{pmatrix} = \\
&= K'R_R \begin{pmatrix} 1 & 0 & 0 & -p_x \\ 0 & 1 & 0 & -p_y \\ 0 & 0 & 0 & 1 \end{pmatrix} \qquad (13)
\end{aligned}$$

The camera thus describes the projection:

$$\begin{aligned}
\begin{pmatrix} u \\ v \\ w \end{pmatrix} &= P_\infty \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix} = \\
&= K'R_R \begin{pmatrix} 1 & 0 & 0 & -p_x \\ 0 & 1 & 0 & -p_y \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix} = \\
&= K'R_R \begin{pmatrix} x - p_x \\ y - p_y \\ 1 \end{pmatrix} \qquad (14)
\end{aligned}$$

The last row turns out to be unnecessary and the transformation can be written using only Cartesian coordinates (eqn. 1).

3. Applications in Motion Capture

When doing 3D reconstruction of far away objects the scaled orthographic camera model is a good approximation. Then the Affine factorization algorithm [10, 2] can be used

to reconstruct points and cameras in 3D, given image correspondences of the points. The result derived in the previous section (eqn. 1) has two useful applications in this context when working with dynamic cameras and dynamic objects:

- For dynamic cameras and non-rigid objects the 3D reconstruction is usually done independently for each frame. However, eqn. 1 can be used to motivate why orthographic cameras can often be seen as static. If this is the case the 3D reconstruction can instead be done in a batch procedure, increasing the accuracy and robustness.
- If dynamic cameras are used for 3D reconstruction of a dynamic object the absolute translation of the reconstructed object is not computed easily. Using eqn. 1 the dynamics of orthographic cameras can be seen as 2D which simplifies the recovering of the external camera parameters and the absolute translation of the object.

We begin by discussing general 3D reconstruction using orthographic cameras and the Affine Factorization algorithm in section 3.1. Then the particular applications are discussed in section 3.2 & 3.3.

3.1. Affine Factorization

Given image positions of some points in at least two cameras the Affine factorization algorithm [10, 2] reconstructs the 3D positions and the camera matrices. Let the column vector $x_{c,j} \in \mathbf{R}^2$ be the image position of point j in camera c and let the corresponding 3D position be $X_j \in \mathbf{R}^3$. The affine factorization algorithm assumes that these points are expressed in coordinate systems with the origins as the centroid of the points. In this way the translations of the cameras are subtracted away and the camera projection can be written in a simple form:

$$x_{c,j} = M_c X_j \tag{15}$$

where M_c is the unknown 2×3 camera matrix of camera c. Since we are assuming the scaled orthographic camera model the camera matrix should be two rows of a rotation matrix multiplied with a scaling factor. The subtraction of the translation is very important for our application. As we will see later rotating orthographic cameras can often be approximated as static due to this. Given the image measurements $x_{c,j}$ we want to find the unknown 3D points X_j and cameras M_c. If we have measured $x_{c,j}$ with some noise we cannot find M_c and X_j that fulfill equation 15 exactly. The factorization algorithm finds the least squares solution to the problem of minimizing the re-projection error:

$$\min_{M_c, X_j} \sum_{c=1}^{C} \sum_{j=1}^{J} \|x_{c,j} - M_c X_j\|^2 \tag{16}$$

3.1.1 Auto-calibration

However, the solution is only unique up to an affine transformation described by the 3×3 matrix A. $M'_c = M_c A^{-1}$ and $X'_j = A X_j$ have the same projection as M_c and X_j, since they have the same product (equation 17).

$$M'_c X'_j = M_c A^{-1} A X_j = M_c X_j \tag{17}$$

The affine reconstruction (X_j, M_c) may be upgraded to a metric reconstruction (X'_j, M'_c) using metric information of the scene and cameras. This process is referred to as auto-calibration [8, 4, 12]. Then the affine transformation A that rectifies the reconstruction is found. The QR-decomposition $A = QR$ decomposes A into an orthogonal matrix Q and an upper triangular matrix R which handles the skew, scale and reflection part of the transformation. We can factor out the scale $s \in (0, \infty)$, the reflection $p \in \{-1, 1\}$ and the skew K from R as well:

$$A = QR = QspK = Qsp \begin{pmatrix} k_1 & k_2 & k_3 \\ 0 & k_4 & k_5 \\ 0 & 0 & 1 \end{pmatrix} \tag{18}$$

General auto-calibration requires computation of all these components.

3.1.2 Translation

By applying the affine factorization algorithm followed by auto-calibration a 3D reconstruction of the scene and the cameras is computed. The 3D points have the mean translation subtracted away though. To reconstruct this as well we proceed in the following way. Let $\bar{x}_c$ denote the mean image position of the points in camera c. The unknown mean 3D position of the points $\bar{X}$ is projected by the already computed camera matrices as:

$$M_c \bar{X} = \bar{x}_c \tag{19}$$

This gives two linear equations per camera and is thus easily solved. By adding the computed mean 3D position $\bar{X}$ to all the reconstructed points X_j their full 3D motion is estimated.

3.1.3 Reconstructing Multiple Frames

Consider the following different cases of 3D reconstruction:

1. **Static scene & static cameras.** In the single frame case the factorization algorithm and auto-calibration can be applied as has just been described.
2. **Static scene & dynamic cameras.** This can be transformed to case 1 by considering the cameras of each frame as new cameras in the original frame.

3. **Rigidly moving scene & dynamic cameras.** This can be transformed to case 1 in the same way as case 2 by considering a single frame and many cameras in a coordinate system that moves with the scene.

4. **Dynamic scene & static cameras.** This can be transformed to case 1 by considering the same point at different frames as different points at the same frame.

5. **Dynamic scene & rigidly moving cameras.** If the cameras move but such that they are static relative to each other, we can consider a coordinate system that moves with the cameras. In this system we have case 4 which can be transformed to case 1.

6. **Dynamic Scene & independently translating cameras.** The mean image translations are subtracted away in the factorization algorithm, i.e. a coordinate system with the origin at the mean position of the points is used. The camera matrices computed in the factorization thus only describe the scaling and rotation of the cameras. Therefore a translating camera can be treated as static in the Factorization algorithm. Therefore this case can be transformed to case 1 in the same way as case 4.

7. **Linearly deformable object & dynamic cameras.** The deformation of the object is decomposed into a low number of linear basis shapes [2, 6]. Works well for deformations that can be linearized, e.g. face expressions. The auto-calibration is more difficult for this case compared to the previous. We do not consider this case in this paper.

8. **Articulated body & dynamic cameras.** Works well if there are many measured image positions for every articulated segment [11, 13, 14, 7]. Not considered in this paper.

9. **Dynamic Scene & dynamic cameras.** Cannot be transformed to case 1 in general. Each frame has to be reconstructed independently.

The first six cases can thus be treated in a similar way. Since the points are considered to be static in some coordinate system, the scaling and rotation will automatically be consistent for the reconstruction. The auto-calibration then only needs to find the skew and reflection. This will be the same for all "real" frames since they are computed in a single "virtual" frame. This results in an accurate and robust computation since all "real" frames are treated in a single batch step, since they share the same points as well as the rectifying affine transformation.

However, the case of a dynamic scene and dynamic cameras is more difficult. Then each frame generally needs to be reconstructed and auto-calibrated independently. The points and the affine rectification matrix will be different for each frame. This leads to a less accurate and robust estimation since there are fewer measurements for each quantity to be computed. In this case the rotation and scaling also needs to be computed for each frame and cannot be ignored in the auto-calibration.

3.2. Accurate & Robust Batch Factorization for Dynamic Scenes & Dynamic Cameras

In section 2 it was shown that dynamic orthographic cameras can often be treated in 2D (eqn. 1). In particular, if the cameras only translate and pan-tilt to follow an object then the camera movement can be considered as just translations. This holds if the distance between the camera and the object is large relative to the translation of the object and the camera. Therefore this case of dynamic scene and dynamic cameras can be treated just as if the cameras were only translating, which can be treated as if the cameras where static, as discussed in the previous section. If the assumption of a large distance between the camera and the object holds this leads to a more accurate and robust factorization. This type of factorization was done in [4] although they did not provide the solid theoretical motivation for why it is applicable.

More generally, consider cameras that are free to translate and rotate in full 3D as well as zooming. If the large distance assumption holds the dynamics of the cameras can be treated as isotropic scaling and 2D translations and rotations in the image plane (eqn. 1). If the 2D rotation and isotropic scaling can be measured from the image it can be compensated for, i.e. the measurements can be transformed to a coordinate system in the image plane that just translates. Then we have the same situation as previously discussed and all frames can be treated in a single batch step increasing the accuracy and robustness. In section 3.3.1 we briefly discuss how to automate the process of measuring scaling, translation and rotation in 2D.

3.3. Reconstructing Translation of an Object using Dynamic Orthographic Cameras

Consider a video of a moving person taken by a dynamic camera. The image motion of the person will depend on the 3D motion of both the person and the camera. However, the image motion of the static background will just depend on the motion of the camera. Thus, by computing the motion of the background the camera motion can be retrieved in principle. For a general dynamic projective camera this relation is complicated. The process can be much simplified if the scaled orthographic camera model is used. We argue that the motion of the background due to the motion of such cameras can often be approximated as 2D translation, rotation and scaling (eqn. 1). In section 3.3.1 we discuss how the measurement of such background motion can be auto-

mated. For now we assume this has been measured.

As described in sections 3.1 & 3.2 the factorization algorithm can be used to reconstruct the cameras and the object. This is done by first transforming the measurements to a coordinate system where the camera only translates. Then this translation is also subtracted away before performing the factorization. If the cameras are static then the 3D translation can be added to the reconstruction as described in section 3.1.2.

But if the cameras are dynamic this approach is not directly applicable. But since the cameras can be treated as only translating we can add the camera translation to the measured image mean position to have them in an absolute coordinate system. Let $m_{c,t}$ denote the image translation of camera c at time t, where the first frame is chosen as zero. Let $\hat{x}_{c,t}$ denote the measured mean image position of the points in camera c at time t, relative the camera translation at the same frame. Let $\bar{x}_{c,t}$ denote the mean image position of the points in a coordinate system that is fixed for all frames which can then be computed as: $\bar{x}_{c,t} = \hat{x}_{c,t} + m_{c,t}$. Then it is possible to proceed as previously described in section 3.1.2 and the full translation can be computed for the 3D reconstruction even though the cameras are dynamic.

3.3.1 Measuring Image Translation, Rotation & Scale

Consider a pair of images that are approximately related by a 2D similarity transformation, i.e. isotropic scaling and translation and rotation in 2D. In this section we briefly discuss how to compute this transformation from the two images. This is related to the well studied problem of taking a set of overlapping images and making a panorama. This can be considered a solved problem if there is not a lot of dynamic objects or motion blur in the images [1]. However, this will typically not be the case when background motion is to be detected in a motion capture application.

Typically when doing 3D reconstruction the first step is to extract a lot of localized features in the images, e.g. SIFT, and then use RANSAC to find corresponding features and the transformation that relates the images [2, 1]. A difficulty of doing this in our application is to differentiate the features belonging to the object from the features of the background. Another difficulty in e.g. a football application is that the background might not have distinct features (fig. 1 & 2).

Another algorithm that is more specialized for measuring 2D similarity transformation is Phase correlation [3, 9, 5]. It utilizes the properties of the Fourier-transform and takes all pixels into account instead of just looking at distinct localized features. A Phase correlation based approach can therefore work as long as the background has some texture even though it lack distinct localized features. Nevertheless, dealing with dynamic objects and motion blur in a robust way is still a complicated and unsolved problem.

4. Experiment

To test the result of section 2 and its applications discussed in section 3 a football game was recorded by three video cameras filming at 25 Hz and a resolution of 1920x1080 pixels. One of the cameras was placed on the stand behind the goal and the other cameras on the stands on each long side. The cameras had static positions but constantly rotated to follow a player and occasionally changed their zooming.

From the recorded video we manually measured the image joint positions for a seven seconds long goal sequence and reconstructed the motion in 3D. See figure 5 for an example of the measured image joint positions. The goal sequence was quite intense with fast actions and the rotation of the cameras resulted in large image translations and motion blur. This is best seen in the supplementary material video.

To test the the application described in section 3.2, i.e. increased accuracy and robustness, the 3D reconstruction was done both in the proposed batch step and independently for each frame as a comparison. As can be seen in the supplementary material video the batch step produces a more accurate and robust reconstruction as predicted. The area that the player moved around in during this sequence seems to be small enough, compared to the distance to the cameras, for our approximation to be valid.

To test the the application described in section 3.3, i.e. reconstructing the translation of an object filmed by dynamic cameras, the translation of the background was first measured. This was done in a semi-automatic way using Phase correlation (section 3.3.1). The translation was manually measured for some key-frames and Phase correlation was used to automatically fill in the gaps. Figure 2 shows two images moved according to the measured virtual camera translation. By looking at the seem between the two images it is seen that the translation approximation works well in practice.

Using the measured background translation the translation of the football player was reconstructed as described in section 3. By comparing with the recorded video we conclude that the reconstructed translation seems to correspond well to the true translation. Figure 6 shows a few frames from the reconstructed 3D motion from two novel view points. However, the result is best seen in the supplementary material video.

5. Conclusion

We have discussed an extension of the useful scaled orthographic camera model. This extension concerns dynamic cameras following an object. If the translation of the camera and the object it tries to follow are small relative to the distance between them then not only can the camera be

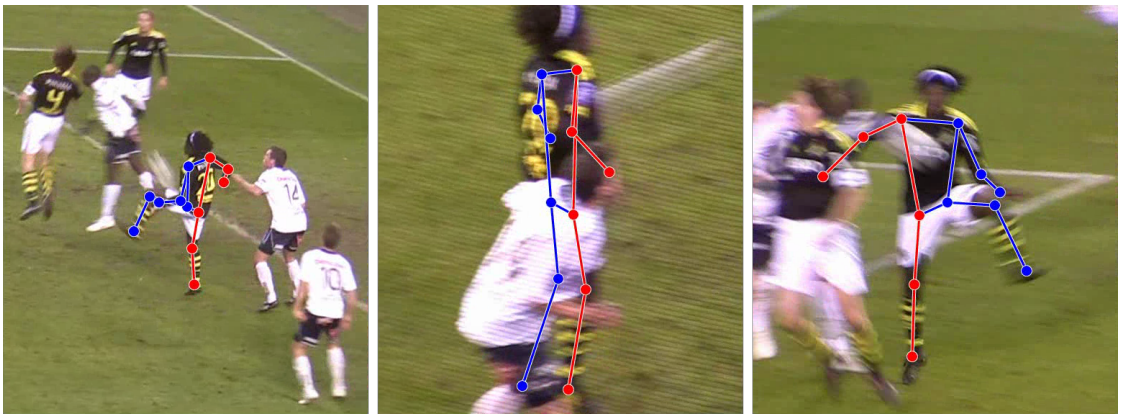

Figure 5. The manually measured image joint positions from a football goal sequence filmed by three rotating cameras.

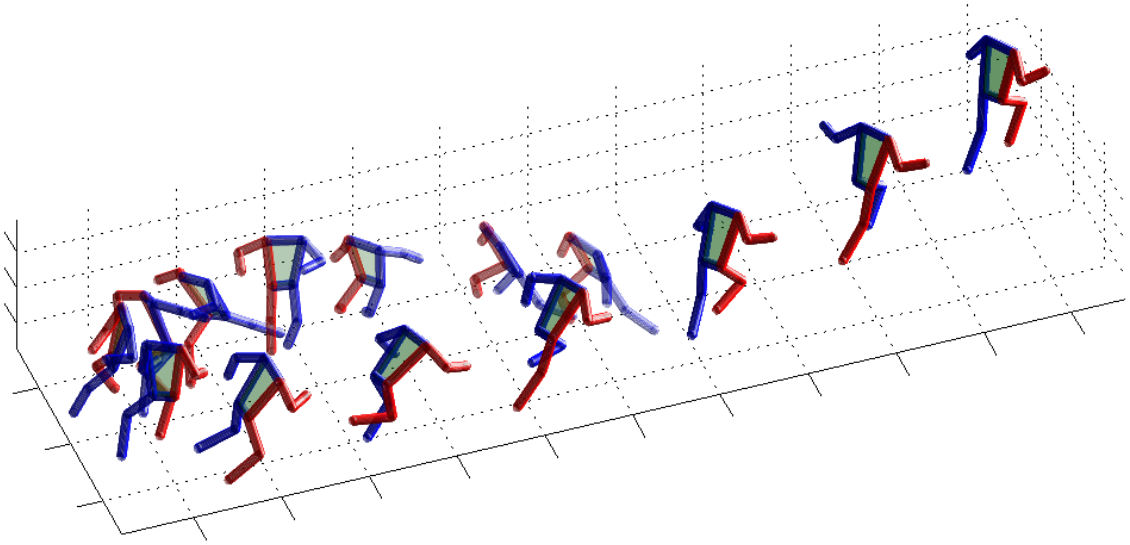

Figure 6. 3D reconstruction of a football goal sequence shown from a novel view point. The result is best seen in color.

approximated as orthographic but its dynamic can be treated in 2D.

This is relevant when doing motion capture at large distances, i.e. outside a studio. In those scenarios the affine factorization algorithm can be used to reconstruct the motion in 3D. If dynamic cameras are used to capture the motion of a dynamic object the factorization is generally done independently for each frame and also the absolute translation of the object is lost.

Using the extended orthographic model we motivated how the factorization can be applied for all frames in a batch procedure for increased accuracy and robustness. We also used it to motivate how the absolute translation of the object can be reconstructed by measuring the approximate 2D motion of the background. Videos from a real football game were used for testing. In our experiments we used manual measurements. A natural and interesting future work is to automate the measuring process.

6. Appendix

Notation for cross product:

$$a \times b = [a]_\times b \tag{20}$$

$$[a]_\times = \begin{pmatrix} 0 & -a_z & a_y \\ a_z & 0 & -a_x \\ -a_y & a_x & 0 \end{pmatrix} \tag{21}$$

The view direction is determined by r_3:

$$\begin{aligned} r_3 &= \frac{p-t}{\|p-t\|} = \frac{p-t_0-\Delta t}{\|p-t_0-\Delta t)\|} = \\ &= \frac{n-t_0}{\|n-t_0\|} = \frac{n+de_3}{\|n+de_3\|} = \\ &= \frac{\frac{n}{d}+e_3}{\|\frac{n}{d}+e_3\|} = \left(\frac{n}{d}+e_3\right) g(d) \end{aligned} \tag{22}$$

$$g(d) = \frac{1}{\|\frac{n}{d}+e_3\|} \tag{23}$$

Since R_{PT} should describe a pan-tilt rotation r_1 should be orthogonal to both r_3 and the unit vector along the y-axis:

$$\begin{aligned} r_1 &= \frac{e_2 \times r_3}{\|e_2 \times r_3\|} = \frac{1}{\|e_2 \times r_3\|} \begin{pmatrix} 0 & 0 & 1 \\ 0 & 0 & 0 \\ -1 & 0 & 0 \end{pmatrix} r_3 = \\ &= \frac{(r_{3,z}, 0, -r_{3,x})^T}{\sqrt{r_{3,z}^2 + r_{3,x}^2}} = (r_{3,z}, 0, -r_{3,x})^T h(d) \quad (24) \\ h(d) &= \frac{1}{\sqrt{r_{3,z}^2 + r_{3,x}^2}} \quad (25) \end{aligned}$$

The remaining row r_2 should be orthogonal to r_1 and r_3:

$$\begin{aligned} r_2 &= -r_1 \times r_3 = \\ &= -h(d) \begin{pmatrix} 0 & r_{3,x} & 0 \\ -r_{3,x} & 0 & -r_{3,z} \\ 0 & r_{3,z} & 0 \end{pmatrix} r_3 = \\ &= (-r_{3,x} r_{3,y}, r_{3,z}^2 + r_{3,x}^2, -r_{3,y} r_{3,z})^T h(d) \quad (26) \end{aligned}$$

As the camera moves infinitely far away we want to compute the following limit value:

$$\lim_{d \to \infty} \begin{pmatrix} r_1^T & r_{1,z} d \\ r_2^T & r_{2,z} d \\ \frac{r_3^T}{d} & r_{3,z} \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & -n_x \\ 0 & 1 & 0 & -n_y \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (27)$$

This was done as follows:

$$\begin{aligned} \lim_{d \to \infty} g(d) &= \lim_{d \to \infty} \frac{1}{\|\frac{n}{d} + e_3\|} = 1 \quad (28) \\ \lim_{d \to \infty} r_3^T &= \lim_{d \to \infty} \left(\frac{n}{d} + e_3\right) g(d) = e_3 = (0,0,1) \quad (29) \\ \lim_{d \to \infty} \frac{r_3^T}{d} &= \lim_{d \to \infty} \frac{e_3}{d} = (0,0,0) \quad (30) \\ \lim_{d \to \infty} h(d) &= \lim_{d \to \infty} \frac{1}{\sqrt{r_{3,z}^2 + r_{3,x}^2}} = 1 \quad (31) \\ \lim_{d \to \infty} r_1^T &= \lim_{d \to \infty} (r_{3,z}, 0, -r_{3,x}) h(d) = (1,0,0) \quad (32) \\ \lim_{d \to \infty} r_2^T &= \lim_{d \to \infty} (-r_{3,x} r_{3,y}, r_{3,z}^2 + r_{3,x}^2, -r_{3,y} r_{3,z}) h(d) = \\ &= (0,1,0) \quad (33) \\ \lim_{d \to \infty} r_{1,z} d &= \lim_{d \to \infty} -r_{3,x} h(d) d = -\frac{n_x}{d} g(d) h(d) d = \\ &= -n_x \quad (34) \\ \lim_{d \to \infty} r_{2,z} d &= \lim_{d \to \infty} -r_{3,y} r_{3,z} h(d) d = \\ &= \lim_{d \to \infty} -\frac{n_y}{d} g(d) (\frac{n_z}{d} + 1) g(d) h(d) d = \\ &= \lim_{d \to \infty} -(\frac{n_y n_z}{d} + n_y) g(d)^2 h(d) = -n_y \quad (35) \end{aligned}$$

References

[1] M. Brown and D. Lowe. Recognising panoramas. In *Computer Vision, 2003. Proceedings. Ninth IEEE International Conference on*, pages 1218 –1225 vol.2, oct. 2003.

[2] R. I. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, ISBN: 0521540518, second edition, 2004.

[3] C. D. Kuglin and D. C. Hines. The phase correlation image alignment method. *Proc. IEEE 1975 Int. Conf. Cybernet. Society*, pages 163 – 165, 1975.

[4] D. Liebowitz and S. Carlsson. Uncalibrated motion capture exploiting articulated structure constraints. *Int. J. Comput. Vision*, 51:171–187, February 2003.

[5] V. Ojansivu and J. Heikkila. Image registration using blur-invariant phase correlation. *Signal Processing Letters, IEEE*, 14(7):449 –452, july 2007.

[6] M. Paladini, A. Bartoli, and L. Agapito. Sequential non-rigid structure-from-motion with the 3d-implicit low-rank shape model. In K. Daniilidis, P. Maragos, and N. Paragios, editors, *Computer Vision – ECCV 2010*, volume 6312 of *Lecture Notes in Computer Science*, pages 15–28. Springer, 2010.

[7] M. Paladini, A. D. Bue, M. Stosic, M. Dodig, J. Xavier, and L. Agapito. Factorization for non-rigid and articulated structure using metric projections. *Computer Vision and Pattern Recognition, IEEE Computer Society Conference on*, 0:2898–2905, 2009.

[8] L. Quan. Self-calibration of an affine camera from multiple views. *Int. J. Comput. Vision*, 19:93–105, July 1996.

[9] B. Reddy and B. Chatterji. An fft-based technique for translation, rotation, and scale-invariant image registration. *Image Processing, IEEE Transactions on*, 5(8):1266 –1271, aug 1996.

[10] C. Tomasi and T. Kanade. Shape and motion from image streams under orthography: a factorization method. *Int. J. Comput. Vision*, 9:137–154, November 1992.

[11] P. Tresadern and I. Reid. Articulated structure from motion by factorization. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, volume 2, pages 1110 – 1115 vol. 2, june 2005.

[12] P. A. Tresadern and I. D. Reid. Camera calibration from human motion. *Image Vision Comput.*, 26:851–862, June 2008.

[13] J. Yan and M. Pollefeys. A factorization-based approach to articulated motion recovery. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, volume 2, pages 815 – 821 vol. 2, june 2005.

[14] J. Yan and M. Pollefeys. A factorization-based approach for articulated nonrigid shape, motion and kinematic chain recovery from video. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 30(5):865 –877, may 2008.