

Optimal Multiple Sprite Generation based on Physical Camera Parameter Estimation

Matthias Kunter^{*a}, Andreas Krutz^a, Mrinal Mandal^b, Thomas Sikora^a

^aCommun. Systems Group, Technische Universität Berlin, Einsteinufer 17, 10587 Berlin, Germany

^bDept. of Electrical & Computer Engineering, University of Alberta, Edmonton, Canada T6G 2V4

ABSTRACT

We present a robust and computational low complex method to estimate the physical camera parameters, intrinsic and extrinsic, for scene shots captured by cameras applying pan, tilt, rotation, and zoom. These parameters are then used to split a sequence of frames into several subsequences in an optimal way to generate multiple sprites. Hereby, optimal means a minimal usage of memory while keeping or even improving the reconstruction quality of the scene background. Since wide angles between two frames of a scene shot cause geometrical distortions using a perspective mapping it is necessary to part the shot into several subsequences. In our approach it is not mandatory that all frames of a subsequence are adjacent frames in the original scene. Furthermore the angle-based classification allows frame reordering and makes our approach very powerful.

Keywords: camera calibration, image mosaicing, multiple sprites, sprite generation

1. INTRODUCTION

The growing computational power of today's computing systems allows modern video analysis and coding tools to apply more complex methods. Therefore, object based approaches are now attracting a great deal of attention. In current hybrid video coding algorithms like MPEG-4, object based schemes are standardized for a long time but reliable all-purpose algorithms providing those objects do emerge only sparsely. A powerful tool to minimize coding costs is the sprite coding tool⁶. A background sprite, also known as video mosaic, is the summarization of all rigid background objects of a video shot into one oversized image^{2,3}. Unfortunately, a universal approach to determine the limits of sprite generation is not available..

In this work, we approach the problem of limiting sprites by providing an efficient algorithm dividing a single video shot into a number of sub-shots based on a robust estimation of intrinsic and extrinsic camera parameters. The partition of the shots and thus a construction of multiple sprites is necessary due to geometrical distortions caused by the perspective image transformation, which is the appropriate mapping for two image planes in 3D space, assuming only very small lens distortions². The perspective mapping is given by

$$\mathbf{x}'' = H(\mathbf{m}) \cdot \mathbf{x} \Rightarrow \begin{pmatrix} x'' \cdot h'' \\ y'' \cdot h'' \\ h'' \end{pmatrix} = \begin{pmatrix} a_1 & a_2 & a_0 \\ b_1 & b_2 & b_0 \\ c_1 & c_2 & 1 \end{pmatrix} \cdot \begin{pmatrix} x \\ y \\ 1 \end{pmatrix}, \quad (1)$$

where $(x,y)^T$ are the coordinates of the reference image and $(x'',y'')^T$ the corresponding points in the image to register. The vector $\mathbf{m}=[a_0..a_2,b_0..b_2,c_1,c_2]$ is the parameter vector. Note that (1) is written in homogeneous coordinates, thus $k \cdot H \equiv H$ for all $k \in \mathbb{R} \setminus \{0\}$. If the mapping angle for a point correspondence approaches 90° , i.e. $c_1 x + c_2 y \approx -1$, the sprite size grows towards infinity. Hence it is obvious that the sprite has to be parted. The mapping $H(\mathbf{m})$ in equation (1), also known as homography, can be decomposed into a matrix F of internal camera parameters and a rotation matrix R

$$\begin{pmatrix} x'' \cdot h'' \\ y'' \cdot h'' \\ h'' \end{pmatrix} = F \cdot R \cdot \begin{pmatrix} x \\ y \\ 1 \end{pmatrix} = \begin{pmatrix} f & 0 & o_x \\ 0 & f & o_y \\ 0 & 0 & 1 \end{pmatrix} \cdot R_{\varphi_z} R_{\varphi_x} R_{\varphi_y} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix}. \quad (2)$$

* kunter@nue.tu-berlin.de; phone +49 30 314-24571; fax +49 30 314-22514; www.nue.tu-berlin.de

Without loss of generality we can omit the coordinates of principle point $(o_x, o_y)^T$ while defining the image middle as the point of origin. Thus, F can be reduced to a diagonal matrix containing focal length f_n . R is a rotation matrix composed of three rotations about y -, x -, and z -axis using Euler angles.

In our work the video shot is divided in an optimal sense, which means that we obtain a minimum memory cost for the multi-sprites. In order to reach this goal, we determine the physical camera parameters especially the focal length f_i and rotation angles $\varphi_{y,i}$ and $\varphi_{x,i}$ and derive a discrete criteria for the estimation of the number of sub-sprites.

Estimation of camera parameters is an important task in image and video processing. The objective of the estimation techniques is to compute the intrinsic and/or extrinsic parameters of the camera, which was used to shoot a particular video. The estimated parameters can be used in many applications such as content analysis and 3D video⁵.

1.1 Multiple Sprites

Farin¹ proposed a technique for construction of multiple sprites with minimal coding costs based on the estimation of the size of the surrounding rectangle. In order to find the best solution, he determines the bounding box size for every pair of frames i, k , being first and last frame of a sub-sprite while testing every reference frame F_r ($i \leq r \leq k$). Tracing back the cost values from the last frame one obtains the optimal sequence partitioning for sprite construction. However, the algorithm still works sequentially, i.e. all frames that compose a sub-sprite are always adjacent frames of the original shot. Additionally, the algorithm itself remains structurally complex even if the cost computation can be simplified.

In our approach we rely on a very comprehensible physical camera parameter based sequence partitioning. If the camera angle between two views of a scene becomes too wide, i.e. the perspective distortion causes huge sprite image sizes, the overall angle (pan and/or tilt) will be parted into M angles of equal size. Thus, the partition algorithm yields in the calculation of M_{opt} minimizing a simple cost function depending on the camera parameter set. One advantage of our approach is the possible reordering of the frames. Therefore, unlike other methods, frames of different temporal positions can compose one sub-sprite.

1.2 Calibration based approach

Figure 1 shows the flowchart of our proposed multiple sprite generation technique. In a first state we calculate the perspective transformation to register every frame F_i of a sequence with respect to the reference frame F_0 while generating a preliminary sprite as proposed in our earlier work⁴. If it is impossible to create one preliminary sprite for the whole sequence we generate several sprites with the last possible registered frame as new reference frame F_r . Second, intrinsic and extrinsic camera parameters are estimated robustly. Then, according to the calibration results we part the video sequence into M subsequences and construct the multiple sprites S_0 to S_{M-1} .

The remainder of the paper is organized as follows. A quick overview of the preliminary sprite generation technique is given in section 2. The estimation of physical camera parameters and the shot partition into several sequences is described in section 3. Section 4 explains the final construction of the multiple sprites. In section 5 we show our experimental results where the quality of our calibration technique is assessed and effectiveness of the multiple sprite generation scheme is measured. Section 6 concludes the paper.

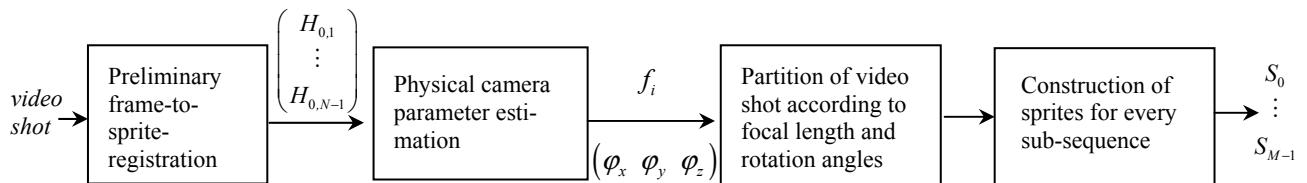


Fig. 1. Flowchart of the proposed calibration based multiple sprite generation approach

2. IMAGE REGISTRATION AND PRELIMINARY SPRITE GENERATION

As shown in Fig. 1, preliminary sprite generation is the first step in our proposed technique. In this section, we present the details of the image registration and sprite generation used in our paper. The flowchart of our sprite generation approach is shown in Fig. 2. The process can be divided into two parts, the image registration and the blending process. In the image registration process optimal transformation parameters for warping the actual frame I_n into the reference

coordinate system of Sprite S_{n-1} are calculated. Since the convergence of a high order transformation calculation is very critical we apply hierarchical parameter estimation starting with feature-based robust estimation of the affine transformation parameters ($a_0...a_2, b_0...b_2, c_1=c_2=0$) between consecutive frames I_n and I_{n-1} . As robust estimation procedure the Monte Carlo type *random sample consensus* (RANSAC)⁸ algorithm is used. These *short-term* parameters are then optimized using Levenberg-Marquardt gradient energy minimization. The energy function is represented by

$$E(n) = \frac{1}{2} \cdot \frac{1}{N_\Omega} \cdot \sum_{(x,y) \in \Omega} (I_{n-1}(x,y) - I_n(x'',y''))^2. \quad (3)$$

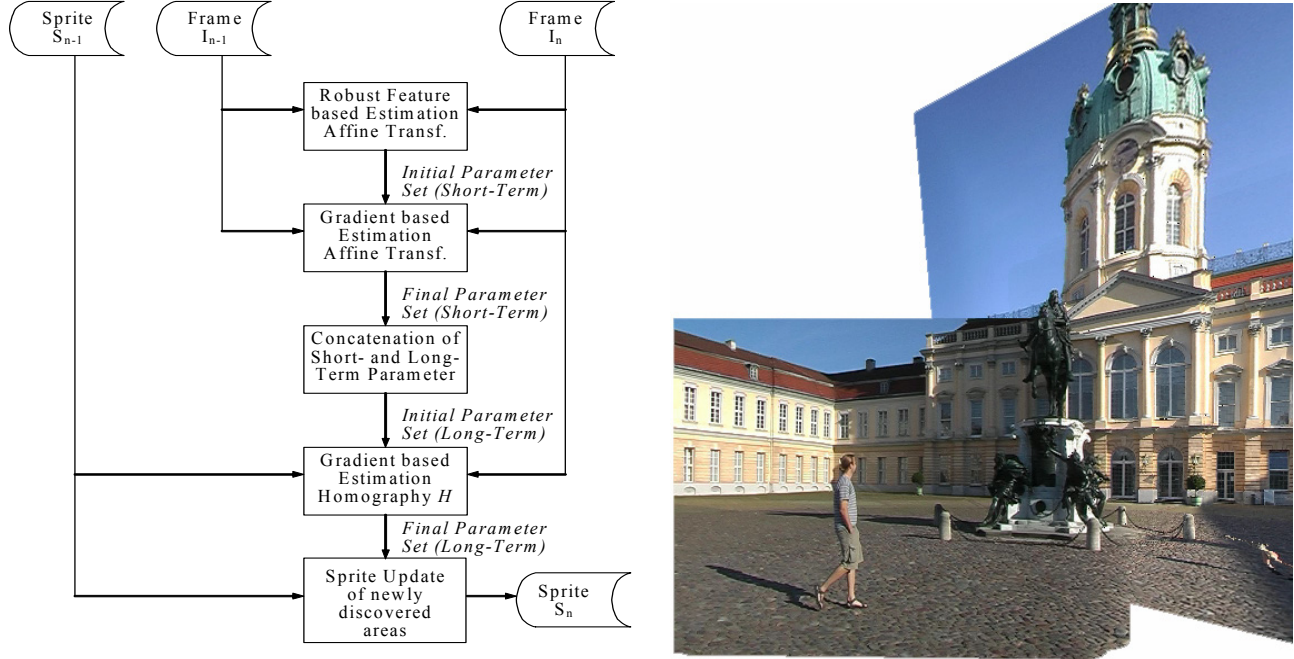


Fig. 2. Flowchart of the sprite construction process (left) and part of preliminary sprite of sequence “Charlottenburg” (right) – see artifacts of the independently moving foreground object

The concatenation of short-term transformation parameters $\mathbf{m}_{n,\text{affine}}$ with previously computed *long-term* parameters $\mathbf{m}_{n-1,\text{pers}}$ using

$$H_{0,n} = H_{n-1,n} \cdot H_{0,n-1} \quad (4)$$

is refined by a direct frame-to-sprite estimation of the Homography $H_{0,n}$ applying Levenberg-Marquardt again. Thus, an open-loop based accumulation of registration errors is prevented.

In order to do the direct frame-to-sprite estimation a preliminary sprite is generated and updated for every registered frame. This is achieved by simply blending newly discovered content into the sprite. It is clear that independently moving foreground objects will distort the visual image quality. However, due to robust registration the parameter estimation will not be affected. Result of this process is a set of homographies $\{H_{0,1} \dots H_{0,N-1}\}$ that relate every frame I_n with $n=0, \dots, N-1$ to the reference image I_0 and can be written as

$$H_{0,n} = H_n \cdot H_0^{-1} = F_n \cdot R_{0,n} \cdot F_0^{-1} = \begin{pmatrix} f_n & 0 & 0 \\ 0 & f_n & 0 \\ 0 & 0 & 1 \end{pmatrix} \cdot R_{\varphi_z} R_{\varphi_x} R_{\varphi_y} \begin{pmatrix} f_0^{-1} & 0 & 0 \\ 0 & f_0^{-1} & 0 \\ 0 & 0 & 1 \end{pmatrix}, \quad (5)$$

where $R_{0,n}$ defines the rotation matrix between view 0 and view n .

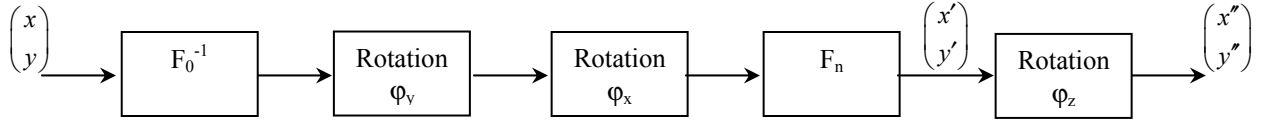


Fig. 3. Scheme of the decomposition of homography $H(m)$ into several physical based extrinsic and intrinsic parameter matrices. Note that vector \mathbf{x}' and \mathbf{x}'' lie in the same image plane and have the same vector norm: $\|\mathbf{x}'\|=\|\mathbf{x}''\|$.

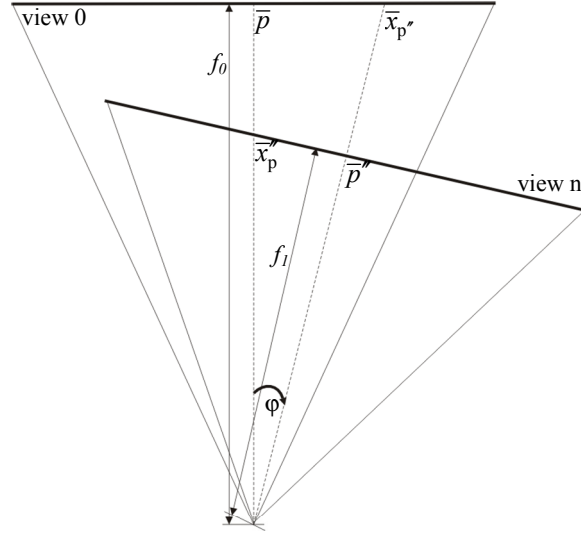


Fig. 4. On the derivation of ratio $\alpha_{0,n}$ between the focal lengths of two views of a scene captured with rotating and zooming camera

3. ESTIMATION OF CAMERA PARAMETERS AND SHOT PARTITIONING

The use of Euler angles for the decomposition of homographies can be difficult because of possible singularities having impact on the numerical stability⁷. On the other hand the order of the three independent rotations (y-axis, x-axis, and z-axis) can be chosen arbitrarily. Thus, we can write $H_{0,n}$ as

$$H_{0,n} = F_n \cdot R_{\varphi_z} \cdot R_{\varphi_x} \cdot R_{\varphi_y} \cdot F_0^{-1} = R_{\varphi_z} \cdot F_n \cdot R_{\varphi_x} \cdot R_{\varphi_y} \cdot F_0^{-1}. \quad (6)$$

Figure 3 shows the decomposed transformation scheme. The final transformation about the z-axis can be regarded as in-plane rotation. Hence, for computation based on vector norms $\|\cdot\|$ in 2D Euclidean space this rotation can be neglected. The in-plane rotation has also no influence for the principal points $\mathbf{p}'=(0,0,1)^T$ and $\mathbf{p}''=(0,0,1)^T$.

3.1 Camera Parameter Estimation

Since for common cameras the principal point $\mathbf{p}=(x_p, y_p, 1)^T=(0,0,1)$ lies in the image middle the focal ratio can be computed as

$$\alpha_{0,n} = \frac{f_0}{f_n} = \frac{\|\mathbf{x}_{p''}\|/\tan \varphi}{\|\mathbf{x}_{p'}\|/\tan \varphi} = \frac{\|\mathbf{x}_{p''}\|}{\|\mathbf{x}_{p'}\|}, \quad (7)$$

where $\mathbf{x}_{p''}$ is the image of the principal point \mathbf{p}'' of view n in view 0 and $\mathbf{x}_{p'}$ the image of the principal point \mathbf{p} of view 0 in view n . See Figure 4 for derivation of equation (7). Having the ratio α of focal length for all image pairs $0, n$ ($1 \leq n \leq N-1$) the value for the focal length f_0 is computed as median value of all real solutions applying Szelisky's linear method². Rewriting equation (5) we obtain

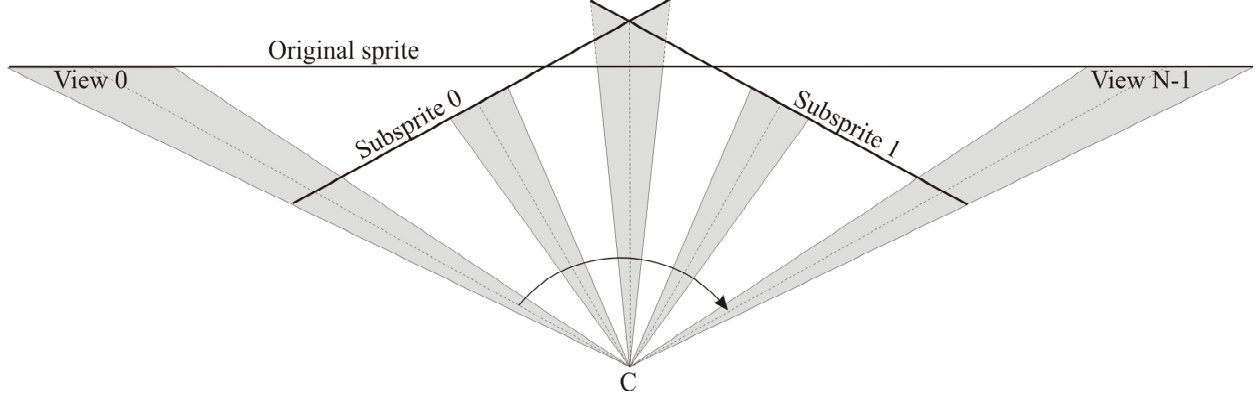


Fig. 5. Partition of one sprite into two subsprites for constant focal length (constant FOV) along horizontal dimension. See that the overall size of the two subsprites is smaller than the original sprite due to geometrical distortions. The reference frame is chosen as the middle frame with respect to the view's absolute angle. The field of view (FOV) is marked as gray.

$$H_{0,n} = \frac{1}{\alpha_{0,n}} \begin{pmatrix} r_{00} & r_{01} & f_0 r_{02} \\ r_{10} & r_{11} & f_0 r_{12} \\ r_{20} \alpha_{0,n} / f_0 & r_{21} \alpha_{0,n} / f_0 & r_{22} \alpha_{0,n} \end{pmatrix} = \begin{pmatrix} h_{00} & h_{01} & h_{02} \\ h_{10} & h_{11} & h_{12} \\ h_{20} & h_{21} & h_{22} \end{pmatrix}. \quad (8)$$

Taking into account the orthogonality constraint for row vectors and column vectors of R as well as the vector norm constraint for rows and columns we can arrange 12 independent equations for the calculation of focal length f_0 . Note that we obtain 12 equations for every homography $H_{0,n}$ with $n=1, \dots, N-1$. From all real solutions we choose the median as final value. As our experimental results show, this method yields accurate results for a minimum number of frames in a scene, which lies between 15 and 25 frames depending on the sequences.

The angles of the rotations are then estimated directly using the images of both principal points. Inverting equation (6) yields in

$$h \cdot \mathbf{x}_{p^*} = F_0 R_{\varphi_y}^{-1} R_{\varphi_x}^{-1} F_n^{-1} R_{\varphi_z}^{-1} \mathbf{p}'' = F_0 R_{\varphi_y}^{-1} R_{\varphi_x}^{-1} F_n^{-1} \mathbf{p}' = F_0 \begin{pmatrix} \sin \varphi_y \cos \varphi_y \\ \sin \varphi_x \\ \cos \varphi_y \cos \varphi_y \end{pmatrix}, \quad (9)$$

which follows in

$$\mathbf{x}_{p^*} = \begin{pmatrix} f_0 \tan \varphi_y \\ f_0 \frac{\tan \varphi_x}{\cos \varphi_y} \\ 1 \end{pmatrix}. \quad (10)$$

After calculating angles φ_y and φ_x we can determine the image of principal point \mathbf{p} in the reference image before and after the final z-rotation

$$\begin{aligned} h' \cdot \mathbf{x}'_p &= F_n R_{\varphi_x} R_{\varphi_y} F_0^{-1} \cdot \mathbf{p} \\ h'' \cdot \mathbf{x}''_p &= H_{0,n} \cdot \mathbf{p} \end{aligned} \quad (11)$$

φ_z is now calculated as difference of the absolute angles of vector \mathbf{x}_p' and \mathbf{x}_p'' with the horizontal line.

$$\begin{aligned} \varphi_z &= \varphi_z'' - \varphi_z' \text{ with} \\ \tan \varphi_z'' &= \frac{y''_p}{x''_p} \text{ and } \tan \varphi_z' = \frac{y'_p}{x'_p}. \end{aligned} \quad (12)$$

3.2 Shot Partitioning

To split a video shot into several sequences, we first compute angle division for the rotation angle (φ_y or φ_x) with the maximum overall rotation $\Delta\varphi_{max}$. We minimize cost function C with respect to the number of angles M in order to find the number of sprites to be generated for one rotation plane.

$$C(\Delta\varphi_{max}, M) = \sum_{i=0}^{M-1} f_{max,i} \cdot 2 \tan\left(\frac{\Delta\varphi_{max}}{2M} + \frac{FOV}{2}\right) \quad (13)$$

Factor $f_{max,i}$ describes the maximum focal length within the set of frames classified by a certain angle range $i \cdot \Delta\varphi_{max}/N \leq \varphi - \varphi_{min} \leq (i+1) \cdot \Delta\varphi_{max}$. The field of view (FOV) is the average aperture angle for the same subset of frames. An example for subdivision along one dimension is given in Figure 5. After subdividing the sequence in one rotation plane (pan or tilt), we apply equation 3 for the perpendicular plane as well. Thus, we handle the sequence division as a separable problem, which is only an approximation for the non-separable perspective mapping. However, for small angles $\Delta\varphi_{max}/N$ the introduced error is negligible. Beside its low computational complexity this approach has the advantage that views that are relative close in space are classified into one subprite even if they are temporal divided by views classified into another subprite.

4. MULTIPLE SPRITE CONSTRUCTION

In this section we describe the eventual generation of the multiple sprites for the different subsequences of the video shot. It can be achieved in two steps. First the exact registration into the particular reference frame coordinate system is achieved by direct estimation using the LM-energy minimization framework of section 2. As reference image always the frame I_r with rotation angle φ closest to the medium angle of a subsequence is used. Since this frame is not the frame with the biggest zoom the reference frame coordinate system is scaled by

$$s = \frac{f_{max}^i}{f_r}, \quad (14)$$

where f_{max}^i is the maximum focal length within a subsequence. Thus, we prevent resolution degradation due to subsampling of images during the warping process.

In a second step we robustly blend the contents of all transformed and warped frames of a subsequence into the final sprite. In order to remove artifacts of independently moving foreground objects temporal filtering is performed. Under all pixel candidates we consider only those pixels which hold

$$\left| I_n(T_{r \rightarrow n}(x_p, y_p)) - m \right| \leq A \cdot \underset{\forall \tau}{\text{median}} \left| I_\tau(T_{r \rightarrow \tau}(x_p, y_p)) - m \right| \quad (15)$$

with $m = \underset{\forall v}{\text{median}} \left(I_v(T_{r \rightarrow v}(x_p, y_p)) \right)$.

$T_{r \rightarrow n}(\cdot)$ represents the transformation from the reference frame to a frame at time n . In order to minimize the lowpass effect as a result of pixel interpolation, out of these candidates the pixel with smallest distance d_p to integer position in the original frame is chosen. To enhance the background reconstruction quality additionally correlation analysis can be conducted, which unfortunately would also increase the algorithms complexity.

5. EXPERIMENTAL RESULTS

In this section we first demonstrate the exactness of our calibration technique (section 5.1) while comparing the estimated camera parameter for a synthetic scene with the ones used for rendering the scene. In section 5.2 we depict the results for sprite partitioning for real scenes.

5.1 Sequence with ground truth – Calibration accuracy

Figures 7 and 8 compare the curves of the estimated camera parameters with the original ones of the rendered scene “TUB-room” (Fig. 6). The preliminary sprite is also shown in Figure 7. As the results show, focal length and rotation angles are estimated very exactly. The maximum relative error for the focal length is less than 2% and the maximum

angle deviation is about 2 degree. To strengthen our parameter estimation result, we compare the vector norms $\|\mathbf{x}'_p\|$ and $\|\mathbf{x}''_p\|$ (images of principal point \mathbf{p} in the reference frame) obtained in two different ways according to equation (11). Smaller values for the difference between these two absolute values indicate a more exact estimation. The diagram in Figure 7 depicts this distance differences $\|\mathbf{x}'_p\| - \|\mathbf{x}''_p\|$ over all frames of the sequence. The maximal deviation is only about $4 \cdot 10^{-14}$ pixels.



Fig. 6. Exemplary frames of rendered sequence “TUB-room”, frame 0, 50, 100, 150 (left to right – 44% of original size).

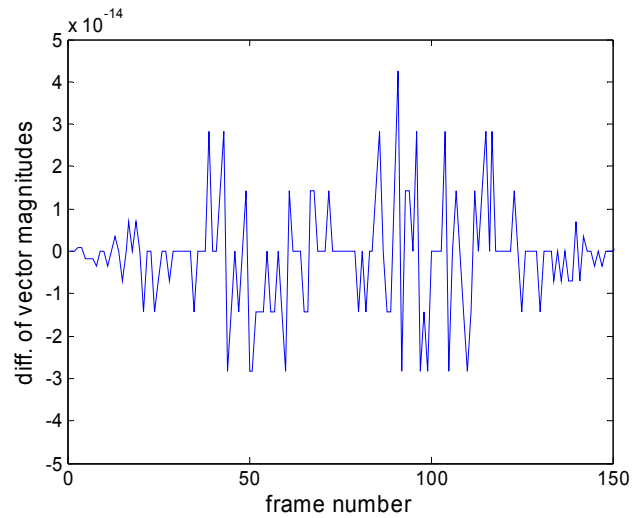


Fig. 7. Preliminary sprite of synthetic sequence “TUB-room” for calibration accuracy assessment (left); Difference $\|\mathbf{x}'_p\| - \|\mathbf{x}''_p\|$ for vector norm of images \mathbf{x}'_p and \mathbf{x}''_p of principal point \mathbf{p} (view 0), sequence “TUB-room” (right) – small values indicate an accurate estimation of rotation angles φ_y and φ_x and focal length f_n with respect to homography $H_{0,n}$ and its inverse $H_{n,0} = (H_{0,n})^{-1}$

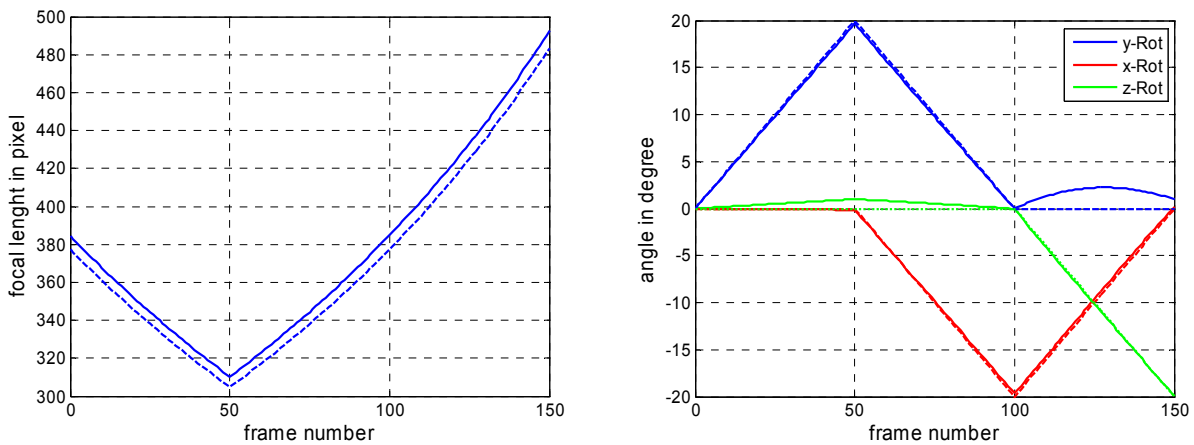


Fig. 8. Comparison of estimation (solid) with ground truth values (dashed) for focal length (left) and rotation angles (right)

5.2 Sequences without ground truth

We calculated extrinsic and intrinsic camera parameters for the well-known “Stefan” sequence. Since the panning angle is too wide to construct only one preliminary sprite a second one was generated using the last registered frame as reference frame. The left diagram of Figure 9 shows the obtained focal length for this sequence compared to the result calculated with a commercial application (Boujou)⁹. The relative difference is about 8.5%. Note that the commercial result is no real ground truth data. The rotation angles (Figure 9 right) are very similar to those estimated with nonlinear methods in Farin’s approach⁷. As indication of the calibration accuracy of this method as well the vector length of the reference principle point images is compared for $\|x'_p\|$ and $\|x''_p\|$ (Figure 12). The maximum difference is about $1.5 \cdot 10^{-13}$ pixels.

The sequence was then split into three subsequences according to equation (13) based on the horizontal rotation. Since only very small vertical rotation appears no further partitioning along the perpendicular dimension was processed. In Figure 9 the partitioning borders and the assignment of frames to the three sprites is depicted. The sprite obtained by a one-sprite-approach and the generated multiple sprites are shown in Figures 10 and 11. From the size of the surrounding rectangles it can be seen that memory cost is much higher for the single sprite even if not all 300 frames of the sequence are registered.

Additionally, we compared the compressed file sizes for the one-sprite-approach (only 264 frames) and the multiple-sprite-approach. The amount of data was reduced by 77.4% (Table 1) in the compressed domain at quality level 75 for JPEG compression. Thus we can approve the results for multiple sprite memory cost minimization found in¹.

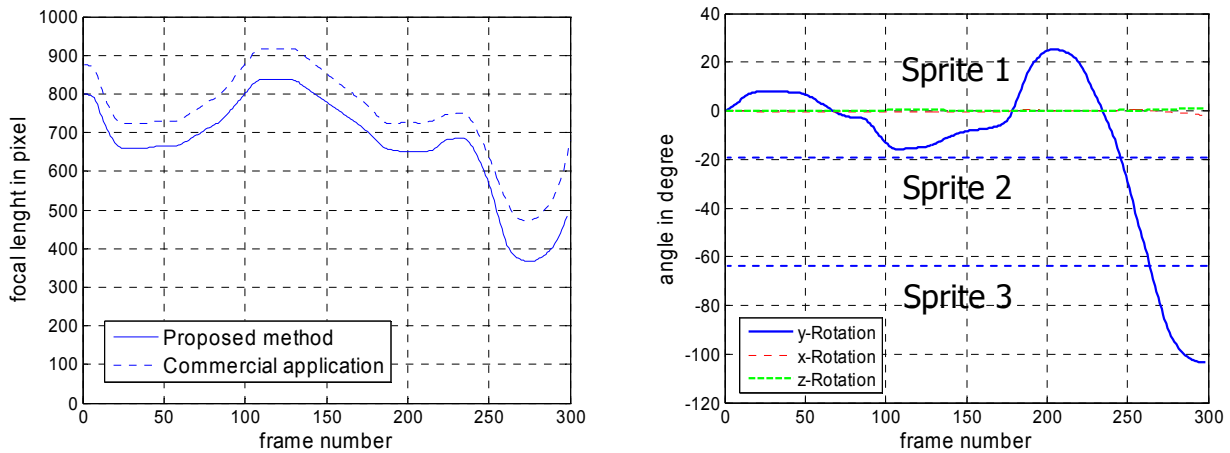


Fig. 9. Focal length (left) and rotation angles with partitions (right) for sequence “Stefan”

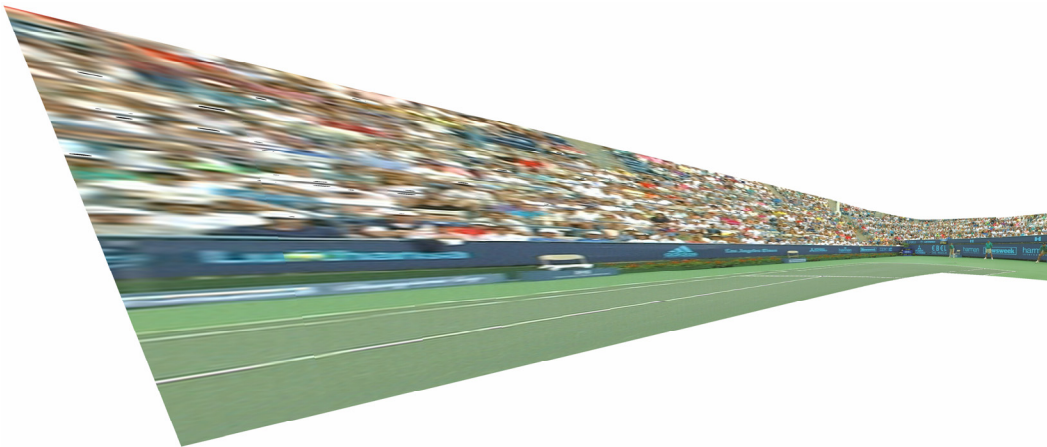


Fig. 10. Single-sprite-approach - “Stefan” frame 0 to 263 (9% of orig. size: 6921x2592pel).

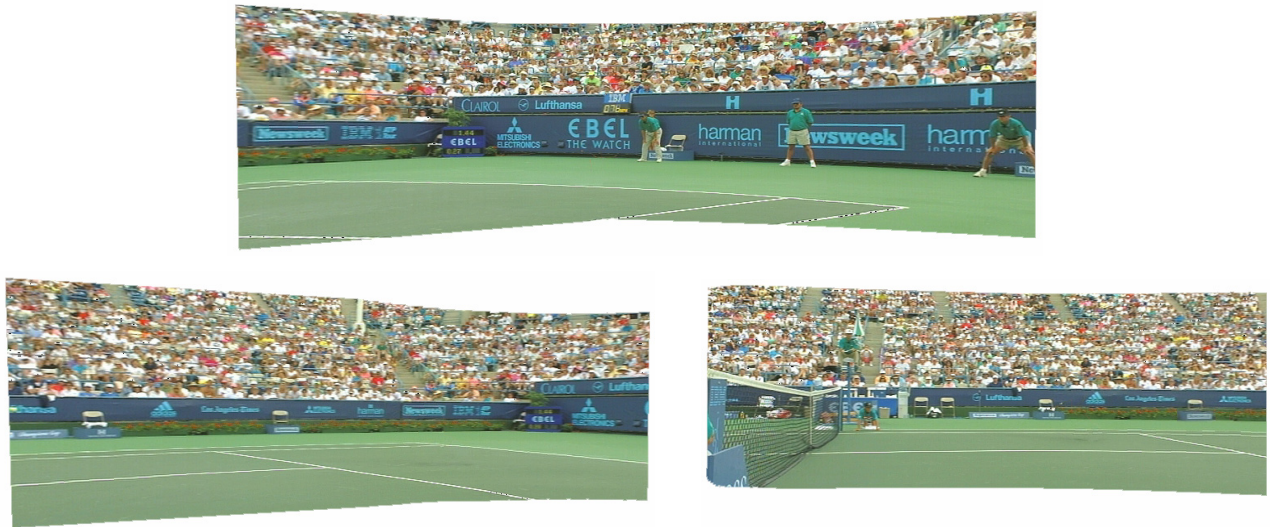


Fig. 11. Multiple-sprite-approach – “Stefan” (35% of original size: 1010x317 pel, frame 0 to 244 – upper; 799x317 pel, frame 245 to 261 – lower left; 679x261 pel, frame 262 to 299 – lower right).

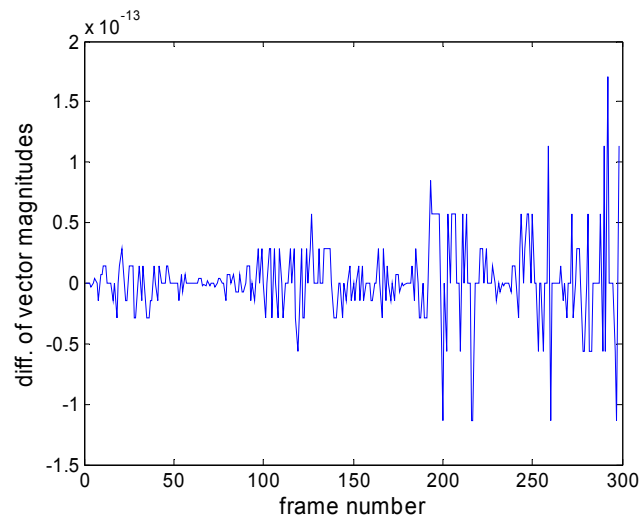


Fig. 12. Difference of vector magnitudes of reference principle point images (sequence “Stefan”)

File sizes for JPEG compression	
Image	File Size
<i>Single-sprite-approach (Quality level 75)</i>	<i>694 kByte</i>
<i>Sprite 1 - multi-sprite-approach (QL 75)</i>	<i>63 kByte</i>
<i>Sprite 2 - multi-sprite-approach (QL 75)</i>	<i>49 kByte</i>
<i>Sprite 3 - multi-sprite-approach (QL 75)</i>	<i>45 kByte</i>

Tab. 1. Comparison of file sizes for single and multi-sprites

6. SUMMARY AND CONCLUSION

We presented a geometry based approach for generation of multiple sprites for scenes recorded with cameras applying pan, tilt, roll, and zoom. In order to derive a rule for subdivision of the video shots, fast and effective intrinsic and extrinsic camera parameter estimation was conducted. Because it is only applied for angle based shot partitioning the calibration does not have to satisfy maximal accuracy constraints. More important for this approach are robustness and low complexity. However, results have shown that our estimation of camera parameters is very close to other calibration techniques. The construction of multiple sprites minimizes the memory cost for the background which is an important matter for sprite based video compression algorithms. Moreover, the reconstruction quality can be preserved or even improved in case of changing background objects. The main advantage over other multiple sprite techniques is the possibility of clustering of temporal non-successive frames into one subsprite.

ACKNOWLEDGMENT

This work was developed within 3DTV (FP6-PLT-511568-3DTV), a European Network of Excellence funded under the European Commission IST FP6 programme.

REFERENCES

1. D. Farin, P. H.N. de With, W. Effelsberg, "Minimizing MPEG-4 Sprite Coding-Cost Using Multiple Sprites," in *SPIE Visual Comm. and Image Proc. (VCIP)*, vol. 5380, Jan 2004.
2. R. Szelisky, H.-Y. Shum, "Creating full view panoramic image mosaics and environment maps", in *Computer Graphics, Annual Convergence Series*, vol. 31, 1997.
3. A. Smolic, T. Sikora, and J.-R. Ohm, "Long-Term Global Motion Estimation and Its Application for Sprite Coding, Content Description, and Segmentation," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 9(8), pp. 1227-1242, Dec. 1998.
4. M. Kunter, J. Kim, T. Sikora, "Super-resolution Mosaicing using Embedded Hybrid Recursive Flow-based Segmentation," *Int. Conf. on Information, Communications and Signal Processing (ICICS '05)*, Dec. 2005.
5. A. Iketani, et al., "Super-resolved video mosaicing for documents based on extrinsic camera parameter estimation," in *Proc. of the 7th Asian Conference on Computer Vision, (ACCV)*, vol. 3852, 2006.
6. T. Sikora, "The MPEG-4 video standard verification model," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 15, pp. 19-31, 1997.
7. D. Farin, P. H. N. de With, "Estimating Physical Camera Parameters based on Multi-Sprite Motion Estimation," in *SPIE Image and Video Communications and Processing*, vol. 5685, January 2005.
8. M. A. Fischler and R. C. Bolles, "Random sample consensus: a paradigm for model fitting with applications to image analysis and auto-mated cartography," *Commun. ACM*, vol. 24, no. 6, pp. 381-393, 1981.
9. <http://www.2d3.com/>