Visual modeling with a hand-held camera

MARC POLLEFEYS

Department of Computer Science, University of North Carolina,

Chapel Hill, NC 27599-3175

marc@cs.unc.edu

LUC VAN GOOL, MAARTEN VERGAUWEN, FRANK VERBIEST,

KURT CORNELIS AND JAN TOPS

Center for Processing of Speech and Images,

Katholieke Universiteit Leuven,

Kasteelpark Arenberg 10, B-3001 Leuven, Belgium

Luc.VanGool@esat.kuleuven.ac.be

Maarten.Vergauwen@esat.kuleuven.ac.be

Frank.Verbiest@esat.kuleuven.ac.be

Kurt.Cornelis@esat.kuleuven.ac.be

Jan.Tops@esat.kuleuven.ac.be

REINHARD KOCH

Institut für Informatik und Praktische Mathematik,

Christian-Albrechts-Universität Kiel,

Hermann-Rodewald-Str. 3, D-24098 Kiel, Germany

rk@mip.informatik.uni-kiel.de

Abstract

In this paper a complete system to build visual models from camera images is presented. The system can deal with uncalibrated image sequences acquired with a hand-held camera. Based on tracked or matched features the relations between multiple views are computed. From this both the structure of the scene and the motion of the camera are retrieved. The ambiguity on the reconstruction is restricted from projective to metric through self-calibration. A flexible multi-view stereo matching scheme is used to obtain a dense estimation of the surface geometry. From the computed data different types of visual models are constructed. Besides the traditional geometry- and image-based approaches, a combined approach with view-dependent geometry and texture is presented. As an application fusion of real and virtual scenes is also shown.

keywords: Visual modeling, Structure-from-Motion, Projective reconstruction, Self-calibration, Multi-view stereo matching, Dense reconstruction, 3D reconstruction, Image-based rendering, Augmented video, hand-held camera.

1 Introduction

During recent years a lot of effort was put in developing new approaches for modeling and rendering visual scenes. A few years ago the main applications of 3d modeling in vision were robot navigation and visual inspection. Nowadays however the emphasis has changed. There is more and more demand for 3D models in computer graphics, virtual reality and communication. This results in a change in the requirements. The visual quality of the models becomes the main point of attention. There is an important demand for simple and flexible acquisition procedures. Therefore calibration should be absent or restricted to a minimum. Many new applications also require robust low cost acquisition systems. This stimulates the use of consumer photo- or video cameras.

In this paper we present an approach that can be used to obtain several types of visual models from images acquired with an uncalibrated camera. The user acquires the images by freely moving the camera around an object or scene. Neither the camera motion nor the camera settings have to be known a priori. The presented approach can generate a textured 3D surface model or alternatively render new views using a combined geometry- and image-based approach that uses view-dependent texture and geometry. The system can also be used to combine virtual objects with real video, yielding augmented video sequences.

Other approaches for extracting 3D shape and texture from image sequences acquired with a freely moving camera have been proposed. The approach of Tomasi and Kanade [59] used an affine factorization method to extract 3D from image sequences. An important restriction of this system is the assumption of orthographic projection. Another type of approach starts from an approximate 3D model and camera poses and refines the model based on images (e.g. *Facade* proposed by Debevec et al. [8]). The advantage is that less images are required. On the other hand a preliminary model must be available which mostly limits the approach to man-made environments. This approach also combines geometry- and image-based techniques, however only the texture is view-dependent.

The approach presented here combines many ideas and algorithms that have been developed in recent years. This paper aims at consolidating these results by bringing them together and

showing how they can be combined to yield a complete visual modeling approach. In Section 1.1 notations and background are given. The rest of the paper is then organized as follows. Section 2 discusses feature extraction and matching and the computation of the multi-view relations. Section 3 deals with the structure and motion recovery, including self-calibration. In Section 4 the approach to obtain dense depth maps is presented and in Section 5 the construction of the different visual models is discussed. The paper is concluded in Section 6.

1.1 Notations and background

In this section we briefly introduce some of the geometric concepts used throughout the paper. A more in depth description can be found in [23, 13]. A *perspective camera* is modeled through the projection equation

$$\mathtt{m} \sim \mathbf{P}\mathtt{M} \tag{1}$$

where $\sim$ represents the equality up to a non-zero scale factor, $\mathtt{M}$ is a 4-vector that represents 3D world point in homogeneous coordinates, similarly $\mathtt{m}$ is a 3-vector that represents a corresponding 2D image point and $\mathbf{P}$ is a 3×4 projection matrix. In a metric or Euclidean frame $\mathbf{P}$ can be factorized as follows

$$\mathbf{P} = \mathbf{K}\mathbf{R}^\top[\,\mathbf{I}\,|\text{-}\mathtt{t}] \text{ where } \mathbf{K} = \begin{bmatrix} f & s & u \\ & rf & v \\ & & 1 \end{bmatrix} \tag{2}$$

contains the intrinsic camera parameters, $\mathbf{R}$ is a rotation matrix representing the orientation and $\mathtt{t}$ is a 3-vector representing the position of the camera. The intrinsic camera parameter f represents the focal length measured in width of pixels, r is the aspect ratio of pixels, (u, v) represent the coordinates of the principal point and s is a term accounting for the skew. In general s can be assumed zero. In practice, the principal point is often close to the center of the image and the aspect ratio r close to one. In many cases the camera does not perfectly satisfy the perspective projection model and distortions have to be taken into account, the most important being radial distortion. In practice, it is often sufficient to model the radial distortion as follows:

$$\mathtt{m} \sim \mathbf{P}(\mathtt{M}) = \mathbf{K}R(\mathbf{R}^\top[\,\mathbf{I}\,|\text{-}\mathtt{t}]\mathtt{M}) \text{ with } R(\mathtt{x}) = (1 + K_1(x^2 + y^2))[x\,y\,0]^\top + [0\,0\,1]^\top \tag{3}$$

where K_1 indicates the amount of radial distortion that is present in the image. For high accuracy applications more advanced models can be used [68, 54].

In this paper the notation $d(.,.)$ will be used to indicate the Euclidean distance between entities in the images.

two view geometry The point $\mathtt{m}'$ corresponding to the point $\mathtt{m}$ in another image is bound to be on the projection of its line of sight $\mathtt{l}' \sim \mathbf{F}\mathtt{m}$ where $\mathbf{F}$ is the *fundamental matrix* for the two views under consideration. Therefore, the following equation should be satisfied for all corresponding points:

$$\mathtt{m}'^{\top}\mathbf{F}\mathtt{m} = 0 \quad . \tag{4}$$

The fundamental matrix has rank 2 and the right and left null-space of $\mathbf{F}$ corresponds to the epipoles. The epipoles $\mathtt{e}$ and $\mathtt{e}'$ are the projections of the projection center of one image in the other image. The fundamental matrix can be obtained from two projection matrices $\mathbf{P}$ and $\mathbf{P}'$ as

$$\mathbf{F} = \left(\mathbf{P}'^{\top}\right)^{\dagger}\mathbf{P}^{\top}[\mathtt{e}]_{\times} \tag{5}$$

where the epipole $\mathtt{e} = \mathbf{P}\mathtt{C}'$ with $\mathtt{C}'$ the solution of $\mathbf{P}'\mathtt{C}' = 0$.

Homographies These can be used to transfer image points that corresponds to 3D points that are on a specific plane from one image to the other, i.e. $\mathtt{m}' \sim \mathbf{H}\mathtt{m}$ where $\mathbf{H}$ is the homography that corresponds to that plane (for the two views under consideration). There is an important relationship between such homographies and the fundamental matrix:

$$\mathbf{F} \sim [\mathtt{e}']_{\times}\mathbf{H} \text{ and } \mathbf{H} = [\mathtt{e}']_{\times}\mathbf{F} - \mathtt{e}'\mathtt{a}^{\top} \tag{6}$$

with $[\mathtt{e}']_{\times}$ an anti-symmetric matrix representing the vector product with the epipole and with $\mathtt{a}$ a vector related to the plane. Homographies for a plane $\mathtt{L} = [a\,b\,c\,d]^{\top}$ can also be obtained from projection matrices as

$$\mathbf{H}^{\mathtt{L}}_{ii'} = \mathbf{H}_{\mathtt{L}i'}\mathbf{H}^{-1}_{\mathtt{L}i} \text{ with } \mathbf{H}_{\mathtt{L}i'} = \mathbf{P}_{i'}\begin{bmatrix} d\mathbf{I} \\ [a\,b\,c] \end{bmatrix} \tag{7}$$

From 3 points $\mathtt{M}_1, \mathtt{M}_2$ and $\mathtt{M}_3$ a plane is obtained as the right null space of $[\mathtt{M}_1\,\mathtt{M}_2\,\mathtt{M}_3]^{\top}$.

comparing images regions Image regions are typically compared using sum-of-square-differences (SSD) or zero-mean normalized cross-correlation (ZNCC). Consider a window W in image I and a corresponding region $\mathbf{T}(W)$ in image J. The *dissimilarity* between two image regions based on SSD is given by

$$D = \int \int_W \left[J(\mathbf{T}(x,y)) - I(x,y) \right]^2 w(x,y) dxdy \tag{8}$$

where $w(x,y)$ is a weighting function that is defined over W. Typically, $w(x,y) = 1$ or it is a Gaussian. The *similarity* measure between two image regions based on ZNCC is given by

$$S = \frac{\int \int_W (J(\mathbf{T}(x,y)) - \bar{J}).(I(x,y) - \bar{I}) w(x,y) dxdy}{\sqrt{\int \int_W (J(\mathbf{T}(x,y)) - \bar{J})^2 w(x,y) dxdy}.\sqrt{\int \int_W (I(x,y) - \bar{I})^2 w(x,y) dxdy}} \tag{9}$$

with $\bar{J} = \int \int_W J(\mathbf{T}(x,y)) dxdy$ and $\bar{I} = \int \int_W I(x,y) dxdy$ the mean image intensity in the considered region. Note that this last measure is invariant to global intensity and contrast changes over the considered regions.

2 Relating images

Starting from a collection of images or a video sequence the first step consists in relating the different images to each other. This is not an easy problem. A restricted number of corresponding points is sufficient to determine the geometric relationship or *multi-view constraints* between the images. Since not all points are equally suited for matching or tracking (e.g. a pixel in a homogeneous region), the first step consist of selecting a number of interesting points or *feature points*. Some approaches also use other features, such as lines or curves, but these will not be discussed here. Depending on the type of image data (i.e. video or still pictures) the feature points are tracked or matched and a number of potential correspondences are obtained. From these the multi-view constraints can be computed. However, since the correspondence problem is an ill-posed problem, the set of corresponding points can be contaminated with an important number of wrong matches or *outliers*. In this case, a traditional least-squares approach will fail and therefore a robust method is needed. Once the multi-view constraints have been obtained

they can be used to guide the search for additional correspondences. These can then be used to further refine the results for the multi-view constraints.

2.1 Feature extraction and matching

One of the most important requirements for a feature point is that it can be differentiated from its neighboring image points. If this were not the case, it wouldn't be possible to match it uniquely with a corresponding point in another image. Therefore, the neighborhood of a feature should be sufficiently different from the neighborhoods obtained after a small displacement.

A second order approximation of the dissimilarity, as defined in Eq. (8), between a image window W and a slightly translated image window is given by

$$d(\Delta x, \Delta y) = \begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix} \mathbf{M} \begin{bmatrix} \Delta x & \Delta y \end{bmatrix} \text{ with } \mathbf{M} = \int\int_W \begin{bmatrix} \frac{\partial I}{\partial x} \\ \frac{\partial I}{\partial y} \end{bmatrix} \begin{bmatrix} \frac{\partial I}{\partial x} & \frac{\partial I}{\partial y} \end{bmatrix} w(x,y) dx dy \tag{10}$$

To ensure that no displacement exists for which D is small, the eigenvalues of $\mathbf{M}$ should both be large. This can be achieved by enforcing a minimal value for the smallest eigenvalue [53] or alternatively for the following corner response function $R = \det \mathbf{M} - k(\text{trace } \mathbf{M})^2$ [17] where k is a parameter set to 0.04 (a suggestion of Harris). In the case of tracking this is sufficient to ensure that features can be tracked from one video frame to the next. In this case it is natural to use the tracking neighborhood to evaluate the quality of a feature (e.g. a 7×7 window with $w(x, y) = 1$). Tracking itself is done by minimizing Eq. 8 over the parameters of $\mathbf{T}$. For small steps a translation is sufficient for $\mathbf{T}$. To evaluate the accumulated difference from the start of the track it is advised to use an affine motion model.

In the case of separate frames as obtained with a still camera, there is the additional requirement that as much image points originating from the same 3D points as possible should be extracted. Therefore, only local maxima of the corner response function are considered as suitable features. Sub-pixel precision can be achieved through quadratic approximation of the neighborhood of the local maxima. A typical choice for $w(x)$ in this case is a Gaussian with $\sigma = 0.7$. Matching is typically done by comparing small, e.g. 7×7, windows centered around

the feature through SSD or ZNCC. This measure is only invariant to image translations and can therefore not cope with too large variations in camera pose.

To match images that are more widely separated, it is required to cope with a larger set of image variations. Exhaustive search over all possible variations is computationally intractable. A more interesting approach consists of extracting a more complex feature that not only determines the position, but also the other unknowns of a local similarity [51] or affine transformation [36, 65].

2.2 Two view geometry computation

Even for an arbitrary geometric structure, the projections of points in two views contain some structure. Finding back this structure is not only interesting to retrieve information on the relative pose between the two views [11, 18], but also to eliminate mismatches and to facilitate the search for additional matches. This structure is corresponds to the epipolar geometry and is mathematically expressed by the fundamental matrix. Given a number of corresponding points Eq. (4) can be used to compute $\mathbf{F}$. This equation can be rewritten in the following form:

$$\begin{bmatrix} xx' & yx' & x' & xy' & yy' & y' & x & y & 1 \end{bmatrix} \mathbf{f} = 0 \tag{11}$$

with $\mathtt{m} = [x\, y\, 1]^\top, \mathtt{m}' = [x'y'1]^\top$ and $\mathbf{f}$ a vector containing the elements of the fundamental matrix. Stacking 8 or more of these equations allows to linearly solve for the fundamental matrix. Even for 7 corresponding points the one parameter family of solutions obtained by solving the linear equations can be restricted to 1 or 3 solutions by enforcing the cubic rank-2 constraint $\det(\mathbf{F}_1 + \lambda\mathbf{F}_2) = 0$. As pointed out by Hartley [20] it is important to normalize the image coordinates before solving the linear equations. Otherwise the columns of Eq. (11) would differ by several orders of magnitude and the error would concentrate on the coefficients corresponding to the smaller columns. This normalization can be achieved by transforming the image center to the origin and scaling the images so that the coordinates have a standard deviation of unity. More advanced approaches have been proposed, e.g. [37], but in practice the simple approach is sufficient to initialize a non-linear minimization. The result of the linear equations can be refined

by minimizing the following criterion [69]:

$$\mathcal{C}(\mathbf{F}) = \sum \left(d(\mathtt{m}', \mathbf{F}\mathtt{m})^2 + d(\mathtt{m}, \mathbf{F}^\top \mathtt{m}')^2 \right) \tag{12}$$

This criterion can be minimized through a Levenberg-Marquard algorithm [47]. An even better approach consists of computing a maximum-likelihood estimation (MLE) by minimizing the following criterion:

$$\mathcal{C}(\mathbf{F}, \hat{\mathtt{m}}, \hat{\mathtt{m}}') = \sum \left(d(\hat{\mathtt{m}}, \mathtt{m})^2 + d(\hat{\mathtt{m}}', \mathtt{m}')^2 \right) \text{ with } \hat{\mathtt{m}}'^\top \mathbf{F} \hat{\mathtt{m}} = 0 \tag{13}$$

Although in this case the minimization has to be carried out over a much larger set of variables, this can be achieved efficiently by taking advantage of the sparsity of the problem (see Section 3.3).

To compute the fundamental matrix from a set of matches that were automatically obtained from a pair of real images, it is important to explicitly deal with outliers. If the set of matches is contaminated with even a small set of outliers, the result of the above method can become unusable. This is typical for all types of least-squares approaches (even non-linear ones). The problem is that the quadratic penalty (which is optimal for Gaussian noise) allows for a single outlier that is very far away from the true solution to completely bias the final result.

The approach that is used to cope with this problem is the RANSAC algorithm that was proposed by Fischler and Bolles [14]. A minimal subset of the data is randomly selected and the solution obtained from it is used to segment the remainder of the dataset in "inliers" and "outliers". If the initial set contains no outliers, it can be expected that an important number of inliers will support the solution, otherwise the initial subset is probably contaminated with outliers. This procedure is repeated until a satisfying solution is obtained. This can for example be defined as a probability in excess of 95% that a good subsample was selected. The expression for this probability is $\Gamma = 1 - (1 - \gamma^p)^m$ with γ the fraction of inliers, and p the number of features in each sample and m the number of trials (see Rousseeuw [48]).

Once the epipolar geometry has been computed it can be used to guide the matching procedure toward additional matches. At this point only features that are close to the epipolar line should be considered for matching. Table 1 summarizes the robust approach to the determination of the two-view geometry.

Step 1. Compute a set of potential matches

Step 2. While $\Gamma(\#inliers, \#samples) < 95\%$ do

step 2.1 select minimal sample (7 matches)

step 2.2 compute solutions for F

step 2.3 determine inliers

step 3. Refine F based on all inliers

step 4. Look for additional matches

step 5. Refine F based on all correct matches

Table 1: Overview of the two-view geometry computation algorithm.

3 Structure and motion recovery

In the previous section it was seen how different views could be related to each other. In this section the relation between the views and the correspondences between the features will be used to retrieve the structure of the scene and the motion of the camera. This problem is called *Structure and Motion*.

The approach that is proposed here extends [1, 30] by being fully projective and therefore not dependent on the quasi-euclidean initialization. This was achieved by carrying out all measurements in the images. This approach provides an alternative for the triplet-based approach proposed in [15]. An image-based measure that is able to obtain a qualitative distance between viewpoints is also proposed to support initialization and determination of close views (independently of the actual projective frame).

At first two images are selected and an initial reconstruction frame is set-up. Then the pose of the camera for the other views is determined in this frame and each time the initial reconstruction is refined and extended. In this way the pose estimation of views that have no common features

with the reference views also becomes possible. Typically, a view is only matched with its predecessor in the sequence. In most cases this works fine, but in some cases (e.g. when the camera moves back and forth) it can be interesting to also relate a new view to a number of additional views. Once the structure and motion has been determined for the whole sequence, the results can be refined through a projective bundle adjustment. Then the ambiguity will be restricted to metric through self-calibration. Finally, a metric bundle adjustment is carried out to obtain an optimal estimation of the structure and motion.

3.1 Initial structure and motion

The first step consists of selecting two views that are suited for initializing the sequential structure and motion computation. On the one hand it is important that sufficient features are matched between these views, on the other hand the views should not be too close to each other so that the initial structure is well-conditioned. The first of these criteria is easy to verify, the second one is harder in the uncalibrated case. The image-based distance that we propose is the median distance between points transferred through an average planar-homography and the corresponding points in the target image:

$$\text{median}\{d(\mathbf{H}\mathtt{m}_j, \mathtt{m}'_j)\} \tag{14}$$

This planar-homography $\mathbf{H}$ is determined as follows from the matches between the two views:

$$\mathbf{H} = [\mathtt{e}]_\times \mathbf{F} + \mathtt{e}\mathtt{a}^\top_{min} \text{ with } \mathtt{a}_{min} = \text{argmin}_\mathtt{a} \sum_i d(([\mathtt{e}]_\times \mathbf{F} + \mathtt{e}\mathtt{a}^\top)\mathtt{m}_j, \mathtt{m}'_j)^2 \tag{15}$$

In practice the selection of the initial frame can be done by maximizing the product of the number of matches and the image-based distance defined above. When features are matched between sparse views, the evaluation can be restricted to consecutive frames. However, when features are tracked over a video sequence, it is important to consider views that are further apart in the sequence.

In the case of a video sequence where consecutive frames are very close together the computation of the epipolar geometry is ill conditioned. To avoid this problem we propose to only consider properly selected key-frames for the structure and motion recovery. If it is important to

compute the motion for all frames, such as for insertion of virtual objects in a video sequence (see Section 5.3), the pose for in-between frames can be computed afterward. We propose to use model selection [61] to select the next key-frame only once the epipolar geometry model explains the tracked features better than the simpler homography model[1].

Initial frame Two images of the sequence are used to determine a reference frame. The world frame is aligned with the first camera. The second camera is chosen so that the epipolar geometry corresponds to the retrieved $\mathbf{F}_{12}$:

$$\begin{aligned} \mathbf{P}_1 &= [\ \mathbf{I}_{3\times 3} \mid 0_3\ ] \\ \mathbf{P}_2 &= [\ [\mathtt{e}_{12}]_\times \mathbf{F}_{12} + \mathtt{e}_{12}\mathtt{a}^\top \mid \sigma \mathtt{e}_{12}\ ] \end{aligned} \tag{16}$$

Eq. (16) is not completely determined by the epipolar geometry (i.e. $\mathbf{F}_{12}$ and $\mathtt{e}_{12}$), but has 4 more degrees of freedom (i.e. $\mathtt{a}$ and σ). $\mathtt{a}$ determines the position of the reference plane (i.e. the plane at infinity in an affine or metric frame) and σ determines the global scale of the reconstruction. The parameter σ can simply be put to one or alternatively the baseline between the two initial views can be scaled to one. In [1] it was proposed to set the coefficient of $\mathtt{a}$ to ensure a quasi-Euclidean frame, to avoid too large projective distortions. This was needed because not all parts of the algorithms where strictly projective. For the structure and motion approach proposed in this paper $\mathtt{a}$ can be arbitrarily set, e.g. $\mathtt{a} = [0\,0\,0]^\top$.

Initializing structure Once two projection matrices have been fully determined the matches can be reconstructed through triangulation. Due to noise the lines of sight will not intersect perfectly. In the uncalibrated case the minimizations should be carried out in the images and not in projective 3D space. Therefore, the distance between the reprojected 3D point and the image points should be minimized:

$$d(\mathtt{m}_1, \mathbf{P}_1\mathtt{M})^2 + d(\mathtt{m}_2, \mathbf{P}_2\mathtt{M})^2 \tag{17}$$

[1] In practice, to avoid selecting too many key-frames, we propose to pick a key-frame at the last frame for which $n_i \geq 0.9 n_e$ with n_i the number of valid tracks and n_e the number of valid tracks when the epipolar geometry model overtakes the homography model.

It was noted by Hartley and Sturm [21] that the only important choice is to select in which epipolar plane the point is reconstructed. Once this choice is made it is trivial to select the optimal point from the plane. A bundle of epipolar planes has only one parameter. In this case the dimension of the problem is reduced from 3-dimensions to 1-dimension. Minimizing the following equation is thus equivalent to minimizing equation (17).

$$d(\mathtt{m}_1, \mathtt{l}_1(\alpha))^2 + d(\mathtt{m}_2, \mathtt{l}_2(\alpha))^2 \tag{18}$$

with $\mathtt{l}_1(\alpha)$ and $\mathtt{l}_2(\alpha)$ the epipolar lines obtained in function of the parameter α describing the bundle of epipolar planes. It turns out (see [21]) that this equation is a polynomial of degree 6 in α. The global minimum of equation (18) can thus easily be computed. In both images the point on the epipolar line $\mathtt{l}_1(\alpha)$ and $\mathtt{l}_2(\alpha)$ closest to the points $\mathtt{m}_1$ resp. $\mathtt{m}_2$ is selected. Since these points are in epipolar correspondence their lines of sight meet in a 3D point.

3.2 Updating the structure and motion

The previous section dealt with obtaining an initial reconstruction from two views. This section discusses how to add a view to an existing reconstruction. First the pose of the camera is determined, then the structure is updated based on the added view and finally new points are initialized.

projective pose estimation For every additional view the pose toward the pre-existing reconstruction is determined, then the reconstruction is updated. This is illustrated in Figure 1. The first step consists of finding the epipolar geometry as described in Section 2.2. Then the matches which correspond to already reconstructed points are used to infer correspondences between 2D and 3D. Based on these the projection matrix $\mathbf{P}_k$ is computed using a robust procedure similar to the one laid out in Table 1. In this case a minimal sample of 6 matches is needed to compute $\mathbf{P}_k$. A point is considered an inlier if there exists a 3D point that projects sufficiently close to all associated image points. This requires to refine the initial solution of $\mathbf{M}$ based on all observations, including the last. Because this is computationally expensive (remember that this has to be done for each generated hypothesis), it is advised to use a modified version of RANSAC that cancels

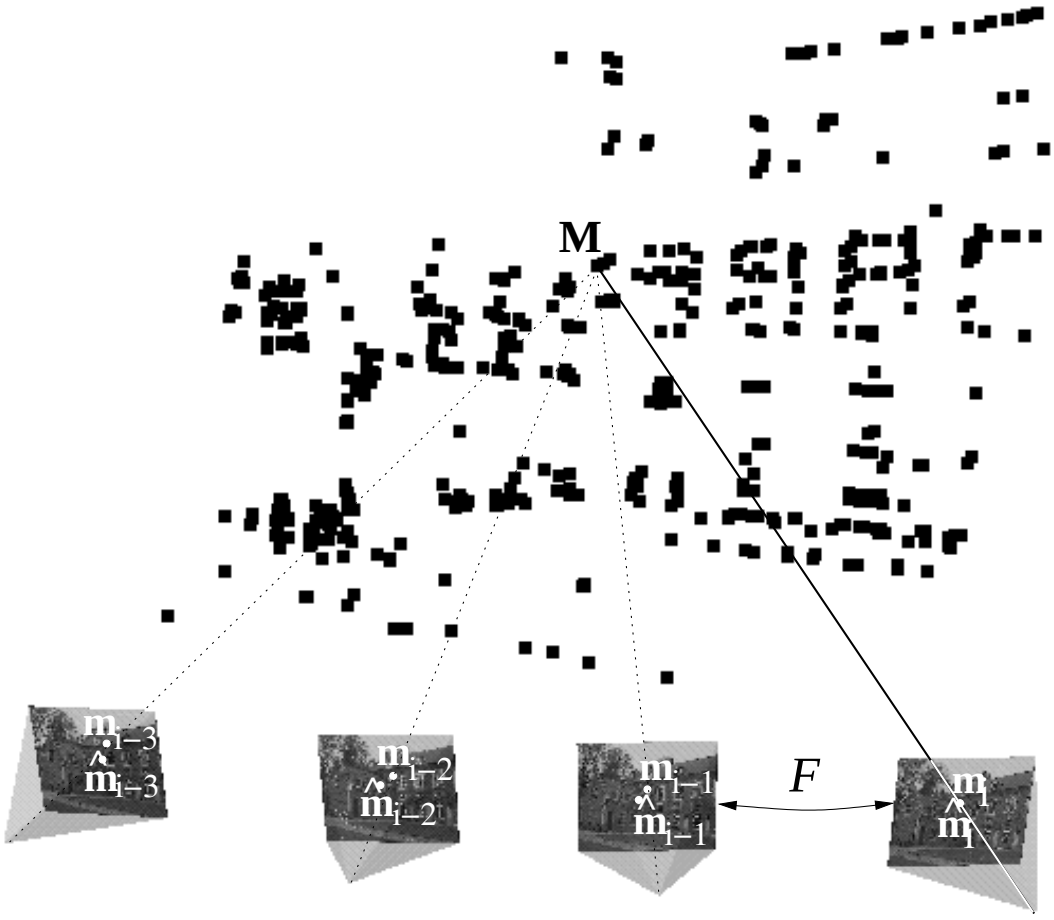

Figure 1: Image matches $(\mathtt{m}_{i-1}, \mathtt{m}_i)$ are found as described before. Since the image points, $\mathtt{m}_{i-1}$, relate to object points, $\mathtt{M}_i$, the pose for view k can be computed from the inferred matches $(\mathtt{M}, \mathtt{m}_i)$. A point is accepted as an inlier if a solution for $\hat{\mathbf{M}}$ exist for which $d(P\hat{\mathbf{M}}, \mathbf{m}_i) < 1$ for each view k in which $\mathbf{M}$ has been observed.

the verification of unpromising hypothesis [4]. Once P_k has been determined the projection of already reconstructed points can be predicted, so that some additional matches can be obtained. This means that the search space is gradually reduced from the full image to the epipolar line to the predicted projection of the point.

This procedure only relates the image to the previous image. In fact it is implicitly assumed that once a point gets out of sight, it will not come back. Although this is true for many sequences, this assumptions does not always hold. Assume that a specific 3D point got out of sight, but that it is visible again in the last two views. In this case a new 3D point will be instantiated. This will not immediately cause problems, but since these two 3D points are unrelated for the system, nothing enforces their position to correspond. For longer sequences where the camera is moved back and forth over the scene, this can lead to poor results due to accumulated errors.

The solution that we propose is to match all the views that are close with the actual view (as described in Section 2.2). For every close view a set of potential 2D-3D correspondences is obtained. These sets are merged and the camera projection matrix is estimated using the same robust procedure as described above, but on the merged set of 2D-3D correspondences [30, 49].

Close views are determined as follows. First a planar-homography that explains best the image-motion of feature points between the actual and the previous view is determined (using Eq. 15). Then, the median residual for the transfer of these features to other views using homographies corresponding to the same plane are computed (see Eq. (14)). Since the direction of the camera motion is given through the epipoles, it is possible to limit the selection to the closest views in each direction. In this case it is important to take orientation into account [22, 33] to differentiate between opposite directions.

Refining and extending structure The structure is refined using an iterated linear reconstruction algorithm on each point. Eq. (1) can be rewritten to become linear in $\mathtt{M}$:

$$\begin{aligned} \mathtt{P}_3\mathtt{M}x - \mathtt{P}_1\mathtt{M} &= 0 \\ \mathtt{P}_3\mathtt{M}y - \mathtt{P}_2\mathtt{M} &= 0 \end{aligned} \tag{19}$$

with $\mathtt{P}_i$ the i-th row of $\mathbf{P}$ and (x, y) being the image coordinates of the point. An estimate of $\mathtt{M}$ is computed by solving the system of linear equations obtained from all views where a

corresponding image point is available. To obtain a better solution the criterion $\sum d(\mathbf{P}\mathtt{M}, \mathtt{m})^2$ should be minimized. This can be approximately obtained by iteratively solving the following weighted linear equations (in matrix form):

$$\frac{1}{\mathtt{P}_3\tilde{\mathtt{M}}} \begin{bmatrix} \mathtt{P}_3 x - \mathtt{P}_1 \\ \mathtt{P}_3 y - \mathtt{P}_2 \end{bmatrix} \mathtt{M} = 0 \tag{20}$$

where $\tilde{\mathtt{M}}$ is the previous solution for $\mathtt{M}$. This procedure can be repeated a few times. By solving this system of equations through SVD a normalized homogeneous point is automatically obtained. If a 3D point is not observed the position is not updated. In this case one can check if the point was seen in a sufficient number of views to be kept in the final reconstruction. This minimum number of views can for example be put to three. This avoids to have an important number of outliers due to spurious matches.

Of course in an image sequence some new features will appear in every new image. If point matches are available that were not related to an existing point in the structure, then a new point can be initialized as in section 3.1.

After this procedure has been repeated for all the images, one disposes of camera poses for all the views and the reconstruction of the interest points. In the further modules mainly the camera calibration is used. The reconstruction itself is used to obtain an estimate of the disparity range for the dense stereo matching.

3.3 Refining structure and motion

Once the structure and motion has been obtained for the whole sequence, it is recommended to refine it through a global minimization step. A maximum likelihood estimation can be obtained through *bundle adjustment* [63, 54]. The goal is to find the parameters of the camera view $\mathbf{P}_k$ and the 3D points $\mathtt{M}_j$ for which the mean squared distances between the observed image points $\mathtt{m}_{ij}$ and the reprojected image points $\mathbf{P}_i(\mathtt{M}_i)$ is minimized. The camera projection model should also take radial distortion into account. For m views and n points the following criterion should

be minimized:

$$\min_{\mathbf{P}_i, \mathtt{M}_j} \sum_{i=1}^{m} \sum_{j=1}^{n} d(\mathtt{m}_{ij}, \mathbf{P}_i(\mathtt{M}_j))^2 \tag{21}$$

If the errors on the localization of image features are independent and satisfy a zero-mean Gaussian distribution then it can be shown that bundle adjustment corresponds to a maximum likelihood estimator. This minimization problem is huge, e.g. for a sequence of 20 views and 100 points/view, a minimization problem in more than 6000 variables has to be solved (most of them related to the structure). A straight-forward computation is obviously not feasible. However, the special structure of the problem can be exploited to solve the problem much more efficiently [63, 54]. The key reason for this is that a specific residual is only dependent on one point and one camera, which results in a very sparse structure for the normal equations.

To conclude this section an overview of the algorithm to retrieve structure and motion from a sequence of images is given. Two views are selected and a projective frame is initialized. The matched corners are reconstructed to obtain an initial structure. The other views in the sequence are related to the existing structure by matching them with their predecessor. Once this is done the structure is updated. Existing points are refined and new points are initialized. When the camera motion implies that points continuously disappear and reappear it is interesting to relate an image to other close views. Once the structure and motion has been retrieved for the whole sequence, the results can be refined through bundle adjustment. The whole procedure is summarized in Table 2.

3.4 Upgrading to metric

The reconstruction obtained as described in the previous sections is only determined up to an arbitrary projective transformation. This might be sufficient for some robotics or inspection applications, but certainly not for visualization. Therefore we need a method to upgrade the reconstruction to a metric one (i.e. determined up to an arbitrary Euclidean transformation and a scale factor). This can be done by imposing some constraints on the intrinsic camera parameters. This approach that is called *self-calibration* has received a lot of attention in recent years. Mostly

Step 1. Match or track points over the whole image sequence.

Step 2. Initialize the structure and motion recovery

- step 2.1. Select two views that are suited for initialization.
- step 2.2. Relate these views by computing the two view geometry.
- step 2.3. Set up the initial frame.
- step 2.4. Reconstruct the initial structure.

Step 3. For every additional view

- step 3.1. Infer matches to the structure and compute the camera pose using a robust algorithm.
- step 3.2. Refine the existing structure.
- step 3.3. (optional) For already computed views which are "close"
  - 3.4.1. Relate this view with the current view by finding feature matches and computing the two view geometry.
  - 3.4.2. Infer new matches to the structure based on the computed matches and add these to the list used in step 3.1.

Refine the pose from all the matches using a robust algorithm.

- step 3.5. Initialize new structure points.

Step 4. Refine the structure and motion through bundle adjustment.

Table 2: Overview of the projective structure and motion algorithm.

self-calibration algorithms are concerned with unknown but constant intrinsic camera parameters [12, 19, 45, 25, 62]. Some algorithms for varying intrinsic camera parameters have also been proposed [44, 26]. In some cases the motion of the camera is not general enough to allow for self-calibration to uniquely recover the metric structure and an ambiguity remains. More details can be found in [56] for constant intrinsics and in [57, 41, 27] for varying intrinsics.

The approach that is presented here was originally proposed in [42] and later adapted to take a priori information on the intrinsic camera parameters into account which reduces the problem of critical motion sequences.

The image of the absolute conic One of the most important concepts for self-calibration is the *absolute conic* and its projection in the images. The simplest way to represent the absolute conic is through the dual absolute quadric Ω^* [62]. In a Euclidean coordinate frame $\Omega^* = \text{diag}(1,1,1,0)$ and one can easily verify that it is invariant to similarity transformations. Inversely, it can also be shown that a transformation that leaves the dual quadric $\Omega^* = \text{diag}(1,1,1,0)$ unchanged is a similarity transformation. For a projective reconstruction Ω^* can be represented by a 4×4 rank-3 symmetric positive semi-definite matrix. According to the properties mentioned above a transformation that transforms $\Omega^* \rightarrow \text{diag}(1,1,1,0)$ will bring the reconstruction within a similarity transformation of the original scene, i.e. yield a metric reconstruction.

The projection of the dual absolute quadric in the image is described by the following equation:

$$\omega^* \sim \mathbf{P}\Omega^*\mathbf{P}^\top \quad . \tag{22}$$

It can be easily verified that in a Euclidean coordinate frame the image of the absolute quadric is directly related to the intrinsic camera parameters:

$$\omega^* \sim \mathbf{K}\mathbf{K}^\top \tag{23}$$

Since the images are independent of the projective basis of the reconstruction, Eq. (23) is always valid and constraints on the intrinsics can be translated to constraints on Ω^*.

linear self-calibration The approach proposed in this paper is inspired from [42], however, some important improvements were made. A priori knowledge about the parameters is introduced in the linear computations [46]. This reduces the problems with critical motion sequences [56, 41].

The first step consists of normalizing the projection matrices. The following normalization is proposed:

$$\mathbf{P}_N = \mathbf{K}_N^{-1}\mathbf{P} \text{ with } \mathbf{K}_N = \begin{bmatrix} w+h & 0 & \frac{w}{2} \\ & w+h & \frac{h}{2} \\ & & 1 \end{bmatrix} \tag{24}$$

where w and h are the width, resp. height of the image. After the normalization the focal length should be of the order of unity and the principal point should be close to the origin. The above normalization would scale a focal length of a 60mm lens to 1 and thus focal lengths in the range of 20mm to 180mm would end up in the range $[1/3, 3]$. The aspect ratio is typically also around 1 and the skew can be assumed 0 for all practical purposes. Making these a priori knowledge more explicit and estimating reasonable standard deviations one could for example get $f \approx rf \approx 1 \pm 3$, $u \approx v \approx 0 \pm 0.1$, $r \approx 1 \pm 0.1$ and $s = 0$. It is now interesting to investigate the impact of this knowledge on ω^*:

$$\omega^* \sim \mathbf{K}\mathbf{K}^\top = \begin{bmatrix} f^2+s^2+u^2 & srf+uv & u \\ srf+uv & r^2f^2+v^2 & v \\ u & v & 1 \end{bmatrix} \approx \begin{bmatrix} 1 \pm 9 & \pm 0.01 & \pm 0.1 \\ \pm 0.01 & 1 \pm 9 & \pm 0.1 \\ \pm 0.1 & \pm 0.1 & 1 \end{bmatrix} \tag{25}$$

and $\omega^*_{22}/\omega^*_{11} \approx 1 \pm 0.2$. The constraints on the left-hand side of Eq. (22) should also be verified on the right-hand side (up to scale). The uncertainty can be take into account by weighting the equations accordingly.

$$\begin{aligned} \tfrac{1}{9\nu}\left(P_1\Omega^*{P_1}^\top - P_3\Omega^*{P_3}^\top\right) &= 0 \\ \tfrac{1}{9\nu}\left(P_2\Omega^*{P_2}^\top - P_3\Omega^*{P_3}^\top\right) &= 0 \\ \tfrac{1}{0.2\nu}\left(P_1\Omega^*{P_1}^\top - P_2\Omega^*{P_2}^\top\right) &= 0 \\ \tfrac{1}{0.1\nu}\left(P_1\Omega^*{P_2}^\top\right) &= 0 \\ \tfrac{1}{0.1\nu}\left(P_1\Omega^*{P_3}^\top\right) &= 0 \\ \tfrac{1}{0.01\nu}\left(P_2\Omega^*{P_3}^\top\right) &= 0 \end{aligned} \tag{26}$$

with P_i the ith row of $\mathbf{P}$ and ν a scale factor that is initially set to 1 and later on to $P_3\tilde{\Omega}^*{P_3}^\top$ with $\tilde{\Omega}^*$ the result of the previous iteration. Since Ω^* is a symmetric 4×4 matrix it is parametrized through 10 coefficients. An estimate of the dual absolute quadric Ω^* can be obtained by solving the above set of equations for all views through linear least-squares. The rank-3 constraint should be imposed by forcing the smallest singular value to zero. This scheme can be iterated until the ν factors converge (typically after a few iterations). In most cases, however, the first iteration is sufficient to initialize the metric bundle adjustment. The upgrading transformation $\mathbf{T}$ can be obtained from diag $(1, 1, 1, 0) = \mathbf{T}\Omega^*\mathbf{T}^\top$ by decomposition of Ω^*.

The metric structure and motion is then obtained as

$$\mathbf{P}_M = \mathbf{P}\mathbf{T}^{-1} \text{ and } \mathtt{M}_M = \mathbf{T}\mathtt{M} \tag{27}$$

refinement This initial metric reconstruction should then further be refined through bundle adjustment to obtain the best possible results. While some approaches suggest an intermediate non-linear refinement of the self-calibration, our experience shows that this is in general not necessary of one uses the self-calibration approach presented in the previous paragraph (as well as an initial correction of the radial distortion). For this bundle adjustment procedure the camera projection model should explicitly represent the constraints on the camera intrinsics. These constraints can both be hard constraints (imposed through parametrization) or soft constraints (imposed by including an additional term in the minimization criterion). A good typical choice of constraints for a photo camera consists of imposing a constant focal length (if no zoom was used), a constant principal point and radial distortion, an aspect ratio of one and the absence of skew. However, for a camcorder/video camera it is important to estimate the (constant) aspect ratio as this can significantly differ from one.

4 Dense surface estimation

With the camera calibration given for all viewpoints of the sequence, we can proceed with methods developed for calibrated structure from motion algorithms. The feature tracking algorithm already delivers a sparse surface model based on distinct feature points. This however is not

sufficient to reconstruct geometrically correct and visually pleasing surface models. This task is accomplished by a dense disparity matching that estimates correspondences from the images by exploiting additional geometrical constraints. The dense surface estimation is done in a number of steps. First image pairs are rectified to the standard stereo configuration. Then disparity maps are computed through a stereo matching algorithm. Finally a multi-view approach integrates the results obtained from several view pairs.

4.1 Rectification

Since the calibration between successive image pairs was computed, the epipolar constraint that restricts the correspondence search to a 1-D search range can be exploited. Image pairs are warped so that epipolar lines coinciding with the image scan lines. The correspondence search is then reduced to a matching of the image points along each image scan-line. This results in a dramatic increase of the computational efficiency of the algorithms by enabling several optimizations in the computations.

For some motions (i.e. when the epipole is located in the image) standard rectification based on planar homographies is not possible and a more advanced procedure should be used. The approach used in the presented system was proposed in [43]. The method combines simplicity with minimal image size and works for all possible motions. The key idea is to use polar coordinates with the epipole as origin. Corresponding lines are given through the epipolar geometry. By taking the orientation [33] into account the matching ambiguity is reduced to half epipolar lines. A minimal image size is achieved by computing the angle between two consecutive epipolar lines that correspond to rows in the rectified images to have the worst case pixel on the line preserve its area. To avoid image degradation, both correction of radial distortion and rectification are performed in a single resampling step.

Some examples A first example comes from the *castle* sequence. In Figure 2 an image pair and the associated rectified image pair are shown. A second example was filmed with a hand-held digital video camera in the Béguinage in Leuven. Due to the narrow streets only forward

Figure 2: Original image pair (left) and rectified image pair (right).

motion is feasible. In this case the full advantage of the polar rectification scheme becomes clear since this sequence could not have been handled through traditional planar rectification. An example of a rectified image pair is given in Figure 3. Note that the whole left part of the rectified images corresponds to the epipole. On the right side of this figure a model that was obtained by combining the results from several image pairs is shown.

4.2 Stereo matching

The goal of a dense stereo algorithm is to compute corresponding pixel for every pixel of an image pair. After rectification the correspondence search is limited to corresponding scanlines. As illustrated in Fig 4, finding the correspondences for a pair of scanlines can be seen as a path search problem. Besides the epipolar geometry other constraints, like preserving the order of neighboring pixels, bidirectional uniqueness of the match, and detection of occlusions can be exploited. In most cases it is also possible to limit the search to a certain disparity range (an estimate of this range can be obtained from the reconstructed 3D feature points). Besides these constraints, a stereo algorithm should also take into account the similarity between corresponding points and the continuity of the surface. It is possible to compute the optimal path taking all the constraints into account using dynamic programming [6, 10, 66]. While many other stereo approaches are available, we use this one because it provides a good trade-off between quality and speed. Computation of a disparity map between two video frame with a disparity range of hundred, including polar rectification, takes less than a minute on a PC. By treating pixels independently stereo can be performed much faster (even real-time), but at the expense of quality.

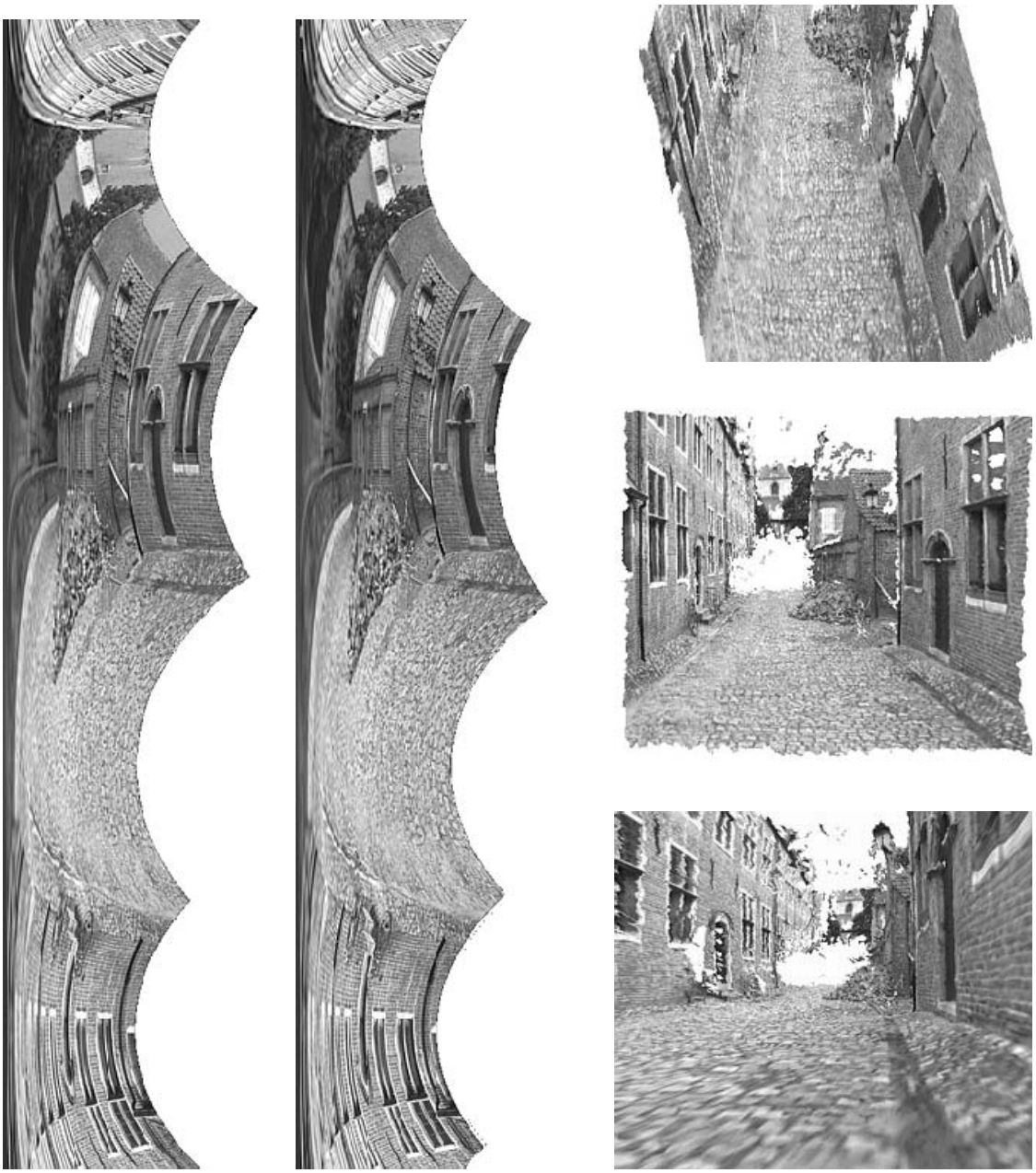

Figure 3: Rectified image pair (left) and some views of the reconstructed street model (right).

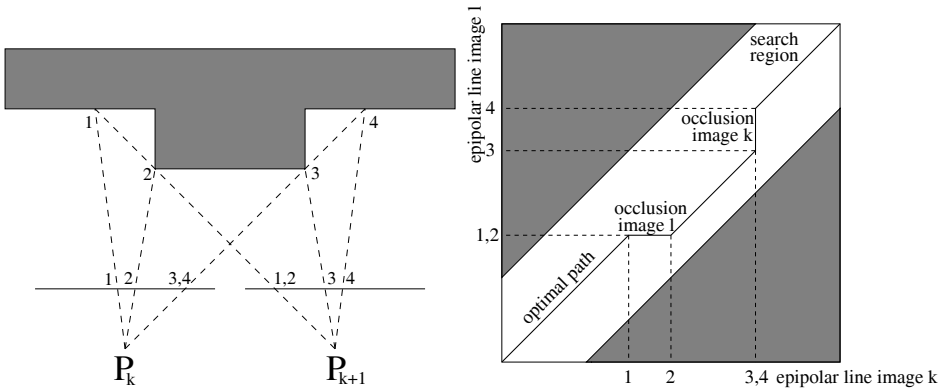

Figure 4: Illustration of the ordering constraint (left), dense matching as a path search problem (right).

Some recent stereo approaches perform a global optimization that also takes continuity across scanlines into account and therefore achieve better results, but are much slower. A good taxonomy of stereo algorithms can be found in [50]. Notice that because of the modular design of our 3D reconstruction pipeline, it is simple to substitute one stereo algorithm for another.

4.3 Multi-view linking

The pairwise disparity estimation allows to compute image to image correspondence between adjacent rectified image pairs, and independent depth estimates for each camera viewpoint. An optimal joint estimate is achieved by fusing all independent estimates into a common 3D model. The fusion can be performed in an economical way through controlled correspondence linking (see Figure 5). A point is transferred from one image to the next image as follows:

$$\mathtt{m}' = R'^{-1}(R(\mathtt{m}) + D(R(\mathtt{m}))) \tag{28}$$

with $R(.)$ and $R'(.)$ functions that map points from the original image into the rectified image and $D(.)$ a function that corresponds to the disparity map. When the depth obtained from the new image point $\mathtt{m}'$ is outside the confidence interval the linking is stopped, otherwise the result is fused with the previous values through a Kalman filter. This approach is discussed into more detail in [29]. This approach combines the advantages of small baseline and wide baseline stereo.

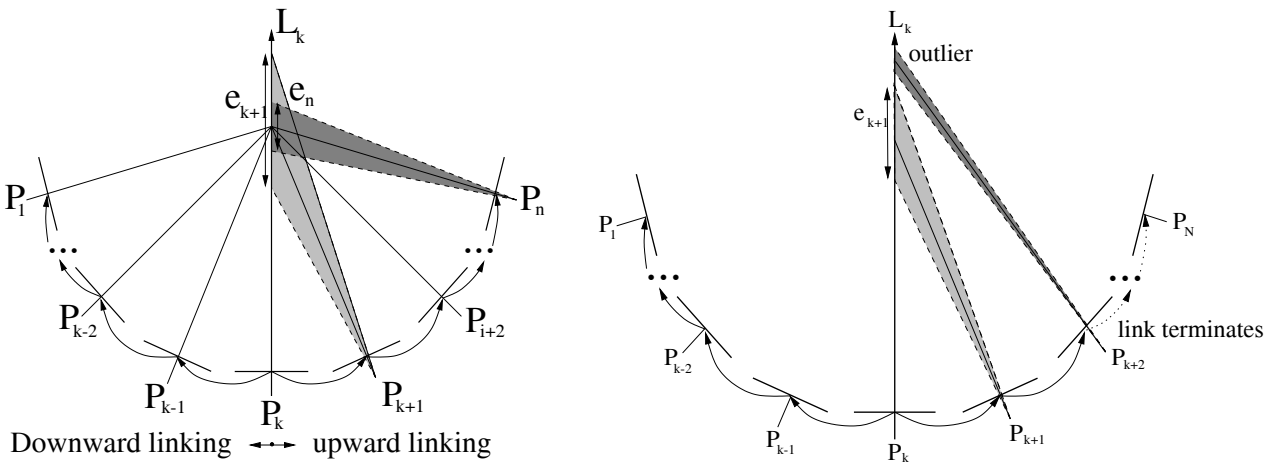

Figure 5: Depth fusion and uncertainty reduction from correspondence linking (left), linking stops when an outlier is encountered (right).

It can provide a very dense depth map by avoiding most occlusions. The depth resolution is increased through the combination of multiple viewpoints and large global baseline while the matching is simplified through the small local baselines. Due to multiple observations of a single surface points the texture can be enhanced and noise and highlights can be removed.

Starting from the computed structure and motion alternative approaches such as sum-of-sum-of-square-differences (SSSD) [40] or space-carving [32] could also be used. The advantages of an approach that only uses pairwise matching followed by multi-view linking, is that it is more robust to changes in camera exposure, non-Lambertian surfaces, passers-by, etc. This is important for obtaining good quality results using hand-held camera sequences recorded in an uncontrolled environment.

Some results The quantitative performance of correspondence linking can be tested in different ways. One measure already mentioned is the visibility of an object point. In connection with correspondence linking, we have defined visibility V as the number of views linked to the reference view. Another important feature of the algorithm is the density and accuracy of the depth maps. To describe its improvement over the 2-view estimator, we define the fill rate F and the average relative depth error E as additional measures. The 2-view disparity estimator

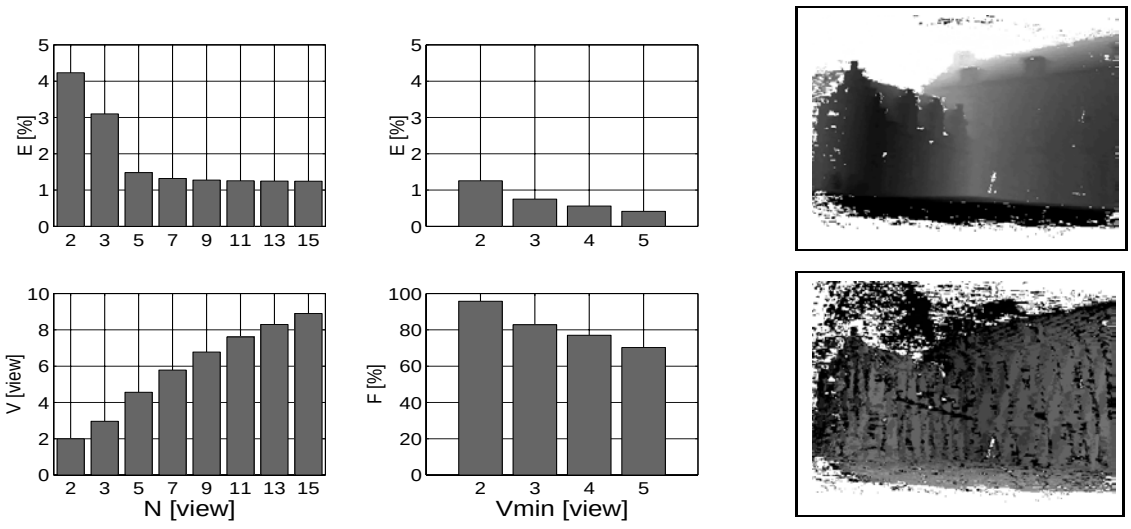

Figure 6: Statistics of the castle sequence. Influence of sequence length N on visibility V and relative depth error E. (left) Influence of minimum visibility V_{min} on fill rate F and depth error E for $N = 11$ (center). Depth map (above: dark=near, light=far) and error map (below: dark=large error, light=small error) for $N = 11$ and $V_{min} = 3$ (right).

is a special case of the proposed linking algorithm, hence both can be compared on an equal basis. Figure 6 displays visibility and relative depth error for sequences from 2 to 15 images of the *castle* sequence, chosen symmetrically around the reference image. The average visibility V shows that for up to 5 images nearly all images are utilized. For 15 images, at average 9 images are linked. The amount of linking is reflected in the relative depth error that drops from 5% in the 2 view estimator to about 1.2% for 15 images.

Linking two views is the minimum case that allows triangulation. To increase the reliability of the estimates, a surface point should be observed in more than two images. We can therefore impose a minimum visibility V_{min} on a depth estimate. This will reject unreliable depth estimates effectively, but will also reduce the fill rate of the depth map. The graphs in figure 6(center) show the dependency of the fill rate and depth error on minimum visibility for the sequence length N=11. The fill rate drops from 92% to about 70%, but at the same time the depth error is reduced to 0.5% due to outlier rejection. The depth map and the relative error distribution over the depth map is displayed in Figure 6(right). The error distribution shows a periodic structure

that in fact reflects the quantization uncertainty of the disparity resolution when it switches from one disparity value to the next.

5 Visual scene representations

In the previous sections a dense structure and motion recovery approach was given. This yields all the necessary information to build different types of visual models. In this section several types of models will be considered. First, the construction of texture-mapped 3D surface models is discussed. Then, a combined image- and geometry-based approach is presented that can render models ranging from pure plenoptic to view-dependent texture and geometry models. Finally, the possibility of fusion of real and virtual scenes in video is also treated. These different cases will now be discussed in more detail.

5.1 3D surface reconstruction

The 3D surface is approximated by a triangular mesh to reduce geometric complexity and to tailor the model to the requirements of computer graphics visualization systems. A simple approach consists of overlaying a 2D triangular mesh on top of one of the images and then build a corresponding 3D mesh by placing the vertices of the triangles in 3D space according to the values found in the corresponding depth map. To reduce noise it is recommended to first smooth the depth image (the kernel can be chosen of the same size as the mesh triangles). The image itself can be used as texture map (the texture coordinates are trivially obtained as the 2D coordinates of the vertices).

It can happen that for some vertices no depth value is available or that the confidence is too low. In these cases the corresponding triangles are not reconstructed. The same happens when triangles are placed over discontinuities. This is achieved by selecting a maximum angle between the normal of a triangle and the line-of-sight through its center (e.g. 85 degrees). This simple approach works very well on the dense depth maps as obtained through multi-view linking. The surface reconstruction approach is illustrated in Figure 7. The texture can be enhanced through

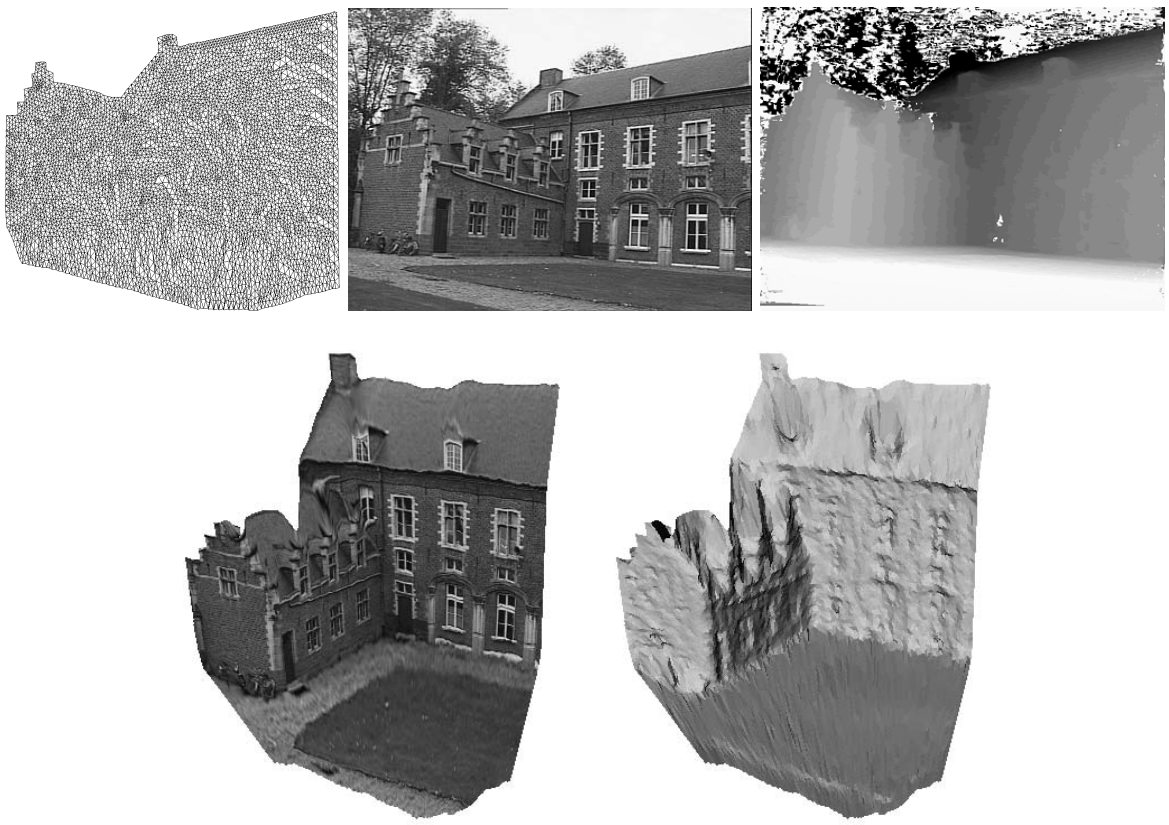

Figure 7: Surface reconstruction approach (top): A triangular mesh is overlaid on top of the image. The vertices are back-projected in space according to the depth values. From this a 3D surface model is obtained (bottom)

the multi-view linking scheme. A median or robust mean of the corresponding texture values is computed to discard imaging artifacts like sensor noise, specular reflections and highlights[39]. Ideally, to avoid artifacts on stretched triangles (such as the ground in Fig. 7) projective texture mapping has to be used in stead of the standard affine mapping [9].

To reconstruct more complex shapes it is necessary to combine results from multiple depth maps. The simplest approach consists of generating separate models independently and then loading them together in the graphics system. Since all depth-maps can be located in a single metric frame, registration is not an issue. For more complex scenes different meshes can be integrated into a single surface representation. Different approaches have been proposed to deal with this problem. These can broadly be classified in surface-based approaches [64, 55] and

volumetric approaches [7, 67]. In our approach we have implemented [7] to obtain an implicit representation of the surface, followed by the marching cubes algorithm to obtain an explicit mesh representation [35] and finally we apply a variant of [52] to simplify the mesh.

Example A first example was recorded using a consumer camcorder (Sony TRV900). A 20 second shot was made of a Medusa head located on the entablature of a monumental fountain in the ancient city of Sagalassos (Turkey). The head itself is about $30cm$ across. Using progressive-scan frames of 720×576 are obtained, an example is shown on the upper-left of Figure 8. Key-frames are automatically selected and the structure of the tracked features and the motion and calibration of the camera is computed, see upper-right of Fig. 8. It is interesting to notice that for this camera the aspect ratio is actually not 1, but around 1.09 which can be observed by comparing the upper-left and the lower-left image in Fig. 8 (notice that it is the real picture that is unnaturally stretched vertically). The next stage consisted of computing a dense surface representation. To this effect stereo matching was performed for all pairs of consecutive key-frames. Using our multi-view linking approach a dense depth map was computed for a central frame and the corresponding image was applied as a texture. Several views of the resulting model are shown in Fig. 8. The shaded view allows to observe the high-quality of the recovered geometry. We have also performed a more quantitative evaluation of the results. The accuracy of the reconstruction should be considered at two levels. Errors on the camera motion and calibration computations can result in a global bias on the reconstruction. From the results of the bundle adjustment we can estimate this error to be of the order of $3mm$ for points on the reconstruction. The depth computations indicate that 90% of the reconstructed points have a relative error of less than $1mm$. Note that the stereo correlation uses a 7×7 window which corresponds to a size of $5mm \times 5mm$ on the object and therefore the measured depth will typically correspond to the dominant visual feature within that patch.

Our second example was also recorded on the archaeological site of Sagalassos. We took a sequence of about 25 photographs of the excavations of a Roman villa at different moments in time so that we could reconstruct the different layers of the stratigraphy. In this case a reconstruction based on a single depth map is not feasible and we have used the volumetric approach

Figure 8: Reconstruction of ancient Medusa head: video frame and recovered structure and motion for key-frames (top), textured and shaded view of 3D reconstruction (middle), frontal view and detailed view (bottom).

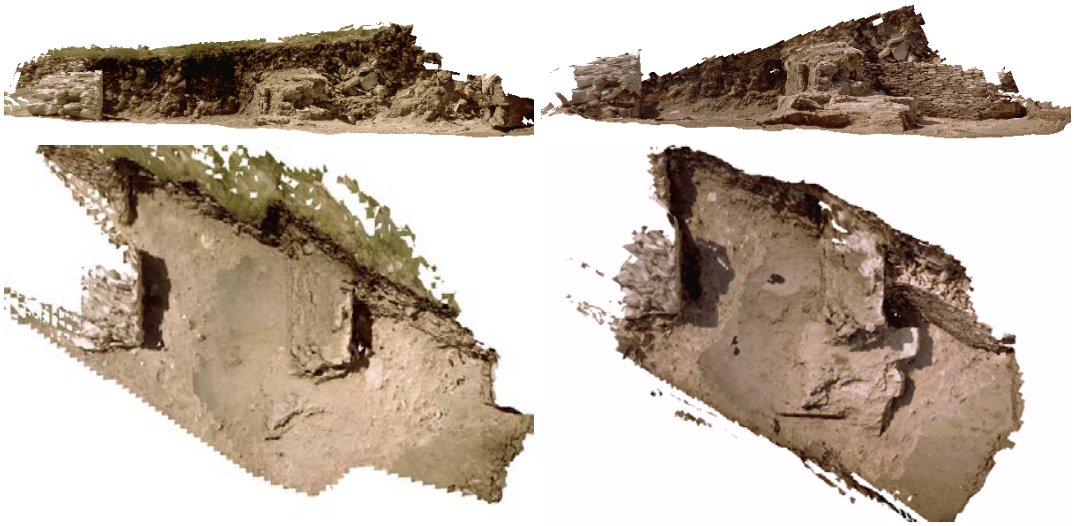

Figure 9: Excavation of Roman villa: front and top view of two different stratigraphic layers.

explained above to extract a single 3D mesh representing the complete stratigraphic layer within the sector of interest. Two of these layers are shown in Figure 9. The possibility to acquire 3D records using cheap consumer products, the absence of calibration procedure and the flexibility and limited time needed for acquisition make our approach particularly suitable for archaeologists.

5.2 Combined image- and geometry-based scene visualization

In this section a different approach is taken to the visualization of 3D scenes. An image-based approach is proposed that can efficiently deal with hand-held camera images. An underlying view-dependent geometry is used to minimize artifacts during visualization. This approach avoids the need for a globally consistent 3D surface representation. This enables to render realistic views of more complex scenes and to reproduce visual effects such as highlights and reflections during rendering. A more in depth discussion of this approach can be found in the following papers [31, 30, 24].

For rendering new views two major concepts are known in literature. The first one is the geometry based concept. The scene geometry is reconstructed from a stream of images and a single texture is synthesized which is mapped onto this geometry. For this approach, a limited set

of camera views is sufficient, but view-dependent effects such as specularities can not be handled appropriately. This approach was discussed in the previous section. The second major concept is image-based rendering. This approach models the scene as a collection of views all around the scene without an exact geometrical representation [34]. New (virtual) views are rendered from the recorded ones by interpolation. Optionally approximate geometrical information can be used to improve the results [16]. It was shown that this can greatly reduce the required amount of images [3].

Plenoptic modeling and rendering In [38] the appearance of a scene is described through all light rays (2D) that are emitted from every 3D scene point, generating a 5D radiance function. Subsequently two equivalent realizations of the plenoptic function were proposed in form of the lightfield [34], and the lumigraph [16]. They handle the case when the observer and the scene can be separated by a surface. Hence the plenoptic function is reduced to four dimensions. The radiance is represented as a function of light rays passing through the separating surface. To create such a plenoptic model for real scenes, a large number of views is taken. These views can be considered as a collection of light rays with according color values. They are discrete samples of the plenoptic function. The light rays which are not represented have to be interpolated from recorded ones considering additional information on physical restrictions. Often real objects are supposed to be Lambertian, meaning that one point of the object has the same radiance value in all possible directions. This implies that two viewing rays have the same color value, if they intersect at a surface point. If specular effects occur, this is not true any more. Two viewing rays then have similar color values if their direction is similar and if their point of intersection is near the real scene point which originates their color value. To render a new view we suppose to have a virtual camera looking at the scene. We determine those viewing rays which are nearest to those of this camera. The nearer a ray is to a given ray, the greater is its support to the color value.

The original 4D lightfield [34] data structure employs a two-plane parameterization. Each light ray passes through two parallel planes with plane coordinates (s, t) and (u, v). The (u, v)-plane is the *viewpoint plane* in which all camera focal points are placed on regular grid points.

The (s,t)-plane is the *focal plane*. New views can be rendered by intersecting each viewing ray of a virtual camera with the two planes at (s,t,u,v). The resulting radiance is a look-up into the regular grid. For rays passing in between the (s,t) and (u,v) grid coordinates an interpolation is applied that will degrade the rendering quality depending on the scene geometry. In fact, the lightfield contains an implicit geometrical assumption, i.e. the scene geometry is planar and coincides with the focal plane. Deviation of the scene geometry from the focal plane causes image degradation (i.e. blurring or ghosting). To use hand-held camera images, the solution proposed in [16] consists of *rebinning* the images to the regular grid. The disadvantage of this rebinning step is that the interpolated regular structure already contains inconsistencies and ghosting artifacts because of errors in the scantily approximated geometry. During rendering the effect of ghosting artifacts is repeated so duplicate ghosting effects occur.

Rendering from recorded images Our goal is to overcome the problems described in the last section by relaxing the restrictions imposed by the regular lightfield structure and to render views directly from the calibrated sequence of recorded images with use of local depth maps. Without loosing performance the original images are directly mapped onto one or more planes viewed by a virtual camera.

To obtain a high-quality image-based scene representation, we need many views from a scene from many directions. For this, we can record an extended image sequence moving the camera in a zigzag like manner. The camera can cross its own moving path several times or at least gets close to it. To obtain a good quality structure-and-motion estimation from this type of sequence it is important to use the extensions proposed in Section 3.2 to match close views that are not predecessors or successors in the image stream. To allow to construct the local geometrical approximation depth maps should also be computed as described in the previous section.

Fixed plane approximation In a first approach, we approximate the scene geometry by a single plane `L` by minimizing the least square error. We map all given camera images onto plane `L` and view it through a virtual camera. This can be achieved by directly mapping the coordinates x_i, y_i of image i onto the virtual camera coordinates $[x_V\, y_V\, 1]^\top = \mathbf{H}_{iV}[x_i\, y_i\, 1]^\top$.

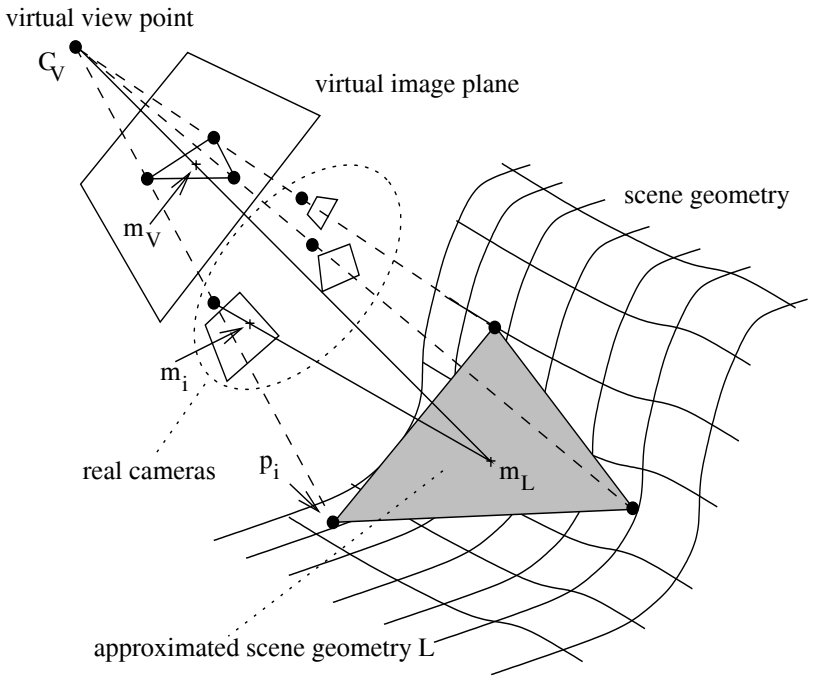

Figure 10: Drawing triangles of neighboring projected camera centers and approximating geometry by one plane for the whole scene, for one camera triple or by several planes for one camera triple.

Therefore we can perform a direct look-up into the originally recorded images and determine the radiance by interpolating the recorded neighboring pixel values. This technique is similar to the lightfield approach [34] which implicitly assumes the focal plane as the plane of geometry. Thus to construct a specific view we have to interpolate between neighboring views. Those views give the most support to the color value of a particular pixel whose projection center is close to the viewing ray of this pixel. This is equivalent to the fact that those views whose projected camera centers are close to its image coordinate give the most support to a specified pixel. We restrict the support to the nearest three cameras (see Figure 10). We project all camera centers into the virtual image and perform a 2D Delaunay triangulation. Then the neighboring cameras of a pixel are determined by the corners of the triangle which this pixel belongs to. Each triangle is drawn as a sum of three triangles. For each camera we look up the color values in the original image like described above and multiply them with weight 1 at the corresponding vertex and with weight 0

at both other vertices. In between, the weights are interpolated linearly similar to the Gouraud shading. Within the triangle the sum of weights is 1 at each point. The total image is built up as a mosaic of these triangles.

View-dependent geometry approximation The results can be improved if depth maps are available. A depth map provides the distance between the projection center and the scene for every pixel. To avoid a runtime penalty a diffusion algorithm is used to fill in undefined depth values. From the depth maps we can calculate the approximating plane of geometry for each triangle of the actual view. This is simple because the points in the original images corresponding to the triangle vertices are easily computed (i.e. the projection of the virtual camera center in the original views). The 3D point $\mathtt{M}_i$ which corresponds to view i can be calculated as

$$\mathtt{M}_i = D_i(\mathbf{P}_i \mathtt{C}_V) \frac{\mathtt{C}_i - \mathtt{C}_V}{\|\mathtt{C}_i - \mathtt{C}_V\|} + \mathtt{C}_i \tag{29}$$

where $D_i(.)$ is a function representing the depthmap (if the virtual camera moves in front of the real cameras the sign of the first term has to be reversed). We can interpret the points $\mathtt{M}_i$ as the intersection of the line $\overline{\mathtt{C}_V \mathtt{C}_i}$ with the scene geometry. Knowing the 3D coordinates of triangle corners, we can define a plane through them and apply the same rendering technique as described above.

Finally, if inside the triangle the scene is not well approximated by a plane, the triangle can be subdivided into four sub-triangles by splitting the three sides into two parts, each. For each of these sub-triangles, a separate approximative plane is calculated in the above manner. Note that the computation of the depth becomes slightly more complicated because for points other than the epipoles the correspondence between view i and v depends on the depth which has therefore to be computed iteratively. Further subdivision can be done in the same manner to improve accuracy. In fact, using the same approach extrapolation also becomes feasible. Conceptually it is sufficient to add the four image corners to the points for which the Delaunay triangulation is computed and then use subdivision to further refine the approximation. In practice, however, results degrade rapidly as one moves away from the original viewpoints. Adding the image corners in the triangulation can also be used to solve the degeneracy that occurs when the virtual camera

passes through the mesh formed by the real camera positions. This extrapolation approach was proposed by Buehler et al [2].

Hardware accelerated rendering We have implemented our approach so that it takes advantage of the available graphics hardware. In doing so real-time performance is easily achieved. Triangulation and depth look-up are performed on the CPU, while the GPU's projective texture mapping and alpha blending functionality are used to build up the image. The level of subdivision can be fixed, or subdivision can be performed adaptively depending on the observed depth variation within a triangle.

In this section, we have shown how the proposed approach for modeling from images could easily be extended to allow the rendering of novel views using a plenoptic or view-dependent texture/geometry representation. The quality of rendered images can be varied by adjusting the resolution of the considered scene geometry. This approach has been described in [31, 30, 24].

Example We have tested our approaches with an image sequence of 187 images showing an office scene. Figure 11 (top-left) shows one particular image. A digital consumer video camera was swept freely over a cluttered scene on a desk, covering a viewing surface of about $1m^2$. Figure 11 (top-right) shows the calibration result. A result of a rendered view is shown in the middle-left part of the figure. The middle-right part illustrates the success of the extended structure-and-motion algorithm. Features that are lost are picked up again when they reappear in the images. In the lower part of Figure 11 a detail of a view is shown for the different methods. In the case of one global plane (left image), the reconstruction is sharp where the approximating plane intersects the actual scene geometry. The reconstruction is blurred where the scene geometry diverges from this plane. In the case of local planes (middle image), at the corners of the triangles, the reconstruction is almost sharp, because there the scene geometry is considered directly. Within a triangle, ghosting artifacts occur where the scene geometry diverges from the particular local plane. If these triangles are subdivided (right image) these artifacts are reduced further.

Figure 11: Top: Image of the *desk* sequence and sparse structure-and-motion result (left) and view rendered using adaptive subdivision (right). Middle: Details of rendered images showing multiple levels of geometric refinement. Bottom: Rendering of a view-dependent effect.

5.3 Combining real and virtual scenes

Another interesting possibility offered by the presented approach is to combine real and virtual scene elements. This allows us to augment real environments with virtual objects. A first approach consists of virtualizing the real environment and then to place virtual objects in it. This can readily be done using the techniques presented in Section 5.1. An example is shown in Figure 12. The landscape of Sagalassos (an archaeological site in Turkey) was modeled from a dozen photographs taken from a nearby hill. Virtual reconstructions of ancient monuments have been made based on measurements and hypotheses of archaeologists. Both could then be combined in a single virtual world.

Figure 12: Virtualized landscape of Sagalassos combined with virtual reconstructions of monuments.

Augmenting video footage Another challenging application consists of seamlessly merging virtual objects with real video. In this case the ultimate goal is to make it impossible to differ-

entiate between real and virtual objects. Several problems need to be overcome before achieving this goal. Among them are the rigid registration of virtual objects into the real environment, the problem of mutual occlusion of real and virtual objects and the extraction of the illumination distribution of the real environment in order to render the virtual objects with this illumination model.

Here we will concentrate on the first of these problems, although the computations described in the previous section also provide most of the necessary information to solve for occlusions and other interactions between the real and virtual components of the augmented scene. Accurate registration of virtual objects into a real environment is still a challenging problems. Systems that fail to do so will fail to give the user a real-life impression of the augmented outcome. Since our approach does not use markers or a-priori knowledge of the scene or the camera, this allows us to deal with video footage of unprepared environments or archive video footage. More details on this approach can be found in [5]. The software Boujou and MatchMover commercialized by 2D3 and RealViz respectively (and inspired by Oxford and INRIA respectively) follow a similar approach.

An important difference with the applications discussed in the previous sections is that in this case all frames of the input video sequence have to be processed while for 3D modeling often a sparse set of views is sufficient. Therefore, in this case features should be tracked from frame to frame. As already mentioned in Section 3.1 it is important that the structure is initialized from frames that are sufficiently separated. Another key component is the bundle adjustment. It does not only reduce the frame to frame jitter, but removes the largest part of the error that the structure and motion approach accumulates over the sequence. According to our experience it is very important to extend the perspective camera model with at least one parameter for radial distortion to obtain an undistorted metric structure (this will clearly be demonstrated in the example). Undistorted models are required to position larger virtual entities correctly in the model and to avoid drift of virtual objects in the augmented video sequences. Note however that for the rendering of the virtual objects the computed radial distortion can most often be ignored (except for sequences where radial distortion is immediately noticeable from single images).

examples This example was recorded at Sagalassos in Turkey, where the footage of the ruins of an ancient fountain was taken. The *fountain* video sequence consists of 250 frames. A large part of the original monument is missing. Based on results of archaeological excavations and architectural studies, it was possible to generate a virtual copy of the missing part. Using the proposed approach the virtual reconstruction could be placed back on the remains of the original monument, at least in the recorded video sequence. This material is of great interest to the archaeologists, not only for education and dissemination, but also for fund raising to achieve a real restoration of the fountain. The top part of Figure 13 shows a top view of the recovered structure before and after bundle-adjustment. Besides the larger reconstruction error it can also be noticed that the non-refined structure is slightly bent. This effect mostly comes from not taking the radial distortion into account in the initial structure recovery. Therefore, a bundle adjustment that did not model radial distortion would not yield satisfying results. In the rest of Figure 13 some frames of the augmented video are shown.

6 Conclusion

In this paper a complete system for visual modeling with a hand-held camera was presented. The system combines different components that gradually retrieve all the information that is necessary to construct visual models from images. Automatically extracted features are tracked or matched between consecutive views and multi-view relations are robustly computed. Based on this the projective structure and motion is determined and subsequently upgraded to metric through self-calibration. Bundle-adjustment is used to refine the results. Then, image pairs are rectified and matched using a stereo algorithm and dense and accurate depth maps are obtained by combining measurements of multiple pairs. For a five image sequence of video resolution, the computation of the metric structure and motion, including bundle adjustment and self-calibration, takes about 20 seconds. Computing pairwise stereo takes less than a minute per pair and performing multi-view linking for the complete sequence takes another 2 minutes, so that the complete automated processing of such a sequence requires 6-7 minutes on a standard PC.

From the computed results different types of visual models can be constructed. First the

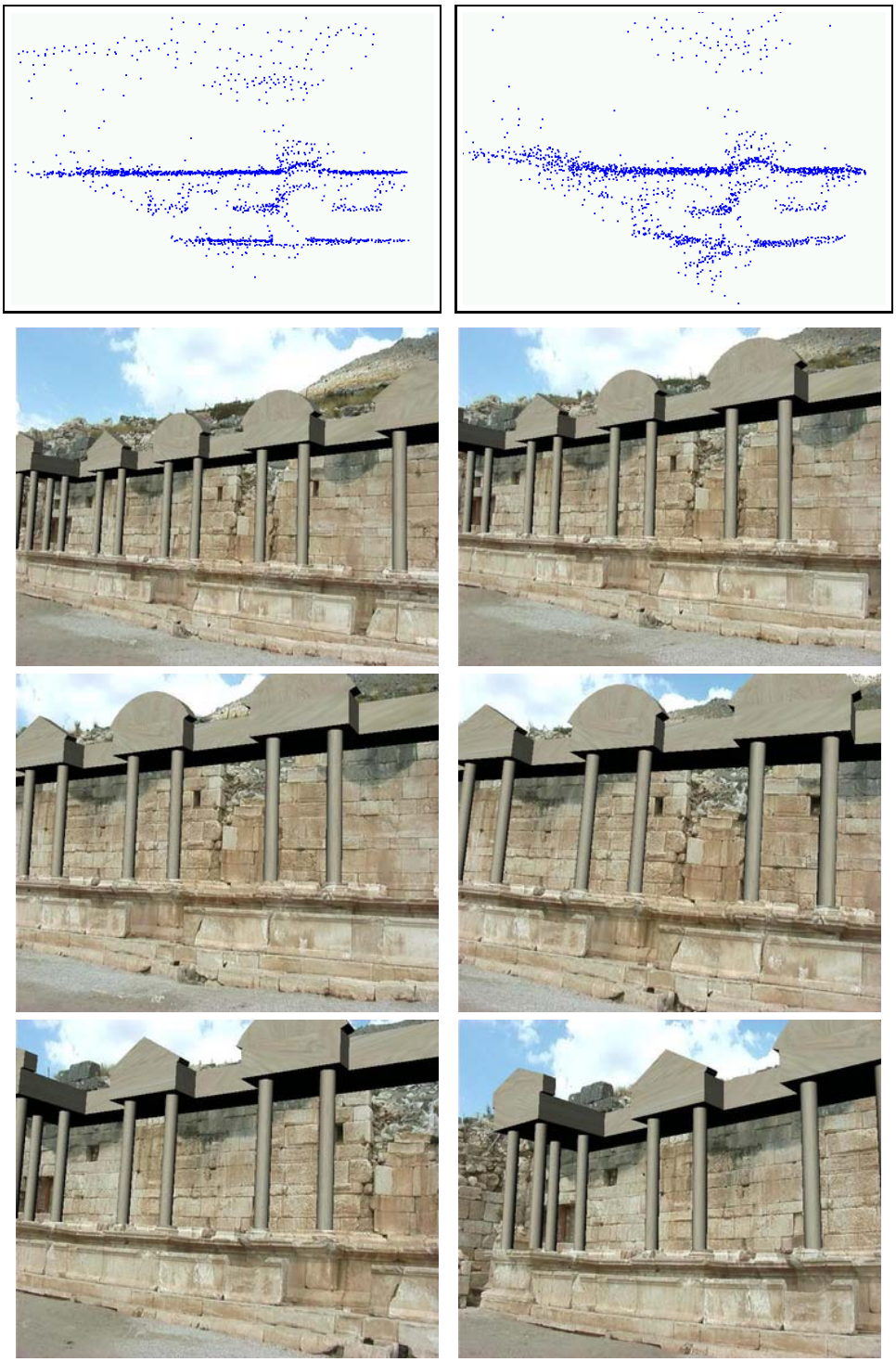

Figure 13: Fusion of real and virtual fountain parts. Top: Top view of the recovered structure-and-motion with and without taking radial distortion into account in the bundle adjustment. Bottom: 6 of the 250 frames of the fused video sequence

traditional approach that consists of constructing a textured 3D mesh was presented, then an image-based approach was described and extended to yield both view-dependent geometry and texture. This last approach allows to efficiently capture visually complex scenes. It was also shown that the proposed approach could be used to combine real and virtual scenes in video sequences.

Acknowledgment

We acknowledge the financial support of the FWO project G.0223.01, the European IST projects Vibes, Murale and InView, as well as of the NSF IIS 0237533 grant. Part of this research was carried out in collaboration with Benno Heigl and colleagues from the university of Erlangen-Nürnberg. The GPU-based implementation of our combined image- and geometry-based rendering was done by Alexander Thomas. We are also grateful to Marc Waelkens and his colleagues for their assistance and for allowing us to record data on the archaeological site of Sagalassos.

References

[1] P. Beardsley, A. Zisserman and D. Murray, "Sequential Updating of Projective and Affine Structure from Motion", *International Journal of Computer Vision* (23), No. 3, Jun-Jul 1997, pp. 235-259.

[2] C. Buehler and M. Bosse and L. McMillan and S. Gortler and M. Cohen, "Unstructured Lumigraph Rendering", In Proceedings ACM SIGGRAPH 2001, pp. 425-432, August 2001.

[3] J.-X. Chai, X. Tong, S.-C. Chan, H.-Y. Shum, "Plenoptic Sampling", Proc. Siggraph, pp.307-318, 2000.y

[4] O. Chum and J. Matas. "Randomized ransac with td,d test". In P. Rosin and D. Marshall, (Eds.), Proceedings of the British Machine Vision Conference, volume 2, pages 448-457, London, UK, September 2002. BMVA.

[5] K. Cornelis, M. Pollefeys, M. Vergauwen and L. Van Gool, "Augmented Reality from Uncalibrated Video Sequences", In M. Pollefeys, L. Van Gool, A. Zisserman, A. Fitzgibbon (Eds.), *3D Structure from Images - SMILE 2000*, Lecture Notes in Computer Science, Vol. 2018, pp.150-167. Springer-Verlag, 2001.

[6] Cox, I., Hingorani, S., Rao, S., 1996, A Maximum Likelihood Stereo Algorithm, Computer Vision and Image Understanding, Vol. 63, No. 3.

[7] B. Curless and M. Levoy, "A Volumetric Method for Building Complex Models from Range Images" *Proc. SIGGRAPH '96*, pp. 303–312, 1996.

[8] P. Debevec, C. Taylor and J. Malik, "Modeling and Rendering Architecture from Photographs: A Hybrid Geometry- and Image-Based Approach", *Proc. SIGGRAPH'96*, pp. 11–20, 1996.

[9] P. Debevec, G. Borshukov and Y. Yu. "Efficient View-Dependent Image-Based Rendering with Projective Texture-Mapping". In 9th Eurographics Rendering Workshop, Vienna, Austria, June 1998.

[10] Falkenhagen, L., 1997, Hierarchical Block-Based Disparity Estimation Considering Neighbourhood Constraints. Proceedings International Workshop on SNHC and 3D Imaging, Rhodes, Greece, pp.115-122.

[11] O. Faugeras, "What can be seen in three dimensions with an uncalibrated stereo rig", *Computer Vision - ECCV'92*, Lecture Notes in Computer Science, Vol. 588, Springer-Verlag, pp. 563-578, 1992.

[12] O. Faugeras, Q.-T. Luong and S. Maybank. "Camera self-calibration: Theory and experiments", *Computer Vision - ECCV'92*, Lecture Notes in Computer Science, Vol. 588, Springer-Verlag, pp. 321-334, 1992.

[13] O. Faugeras, Q.-T. Luong, T. Papadopoulo, *The geometry of multiple images* MIT press, 2001.

[14] M. Fischler and R. Bolles, “RANdom SAmpling Consensus: a paradigm for model fitting with application to image analysis and automated cartography”, *Commun. Assoc. Comp. Mach.*, 24:381-95, 1981.

[15] A. Fitzgibbon and A. Zisserman, “Automatic camera recovery for closed or open image sequences”, *Computer Vision – ECCV’98*, vol.1, Lecture Notes in Computer Science, Vol. 1406, Springer-Verlag, 1998. pp.311-326, 1998.

[16] S. Gortler, R. Grzeszczuk, R. Szeliski and M. F. Cohen, “The Lumigraph”, *Proc. SIGGRAPH ’96*, pp 43–54, ACM Press, New York, 1996.

[17] C. Harris and M. Stephens, “A combined corner and edge detector”, *Fourth Alvey Vision Conference*, pp.147-151, 1988.

[18] R. Hartley, R. Gupta, and T. Chang. “Stereo from uncalibrated cameras”. Proc. Conference Computer Vision and Pattern Recognition, pp. 761-764, 1992.

[19] R. Hartley, “Euclidean reconstruction from uncalibrated views”, in : J.L. Mundy, A. Zisserman, and D. Forsyth (eds.), *Applications of Invariance in Computer Vision*, Lecture Notes in Computer Science, Vol. 825, Springer-Verlag, pp. 237-256, 1994.

[20] R. Hartley, “In defense of the eight-point algorithm”. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 19(6):580-593, June 1997.

[21] R. Hartley and P. Sturm, “Triangulation”, *Computer Vision and Image Understanding*, 68(2):146-157, 1997.

[22] R.Hartley, Chirality *International Journal of Computer Vision*, 26(1):41-61, January 1998.

[23] R. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision.* Cambridge University Press, 2000.

[24] B. Heigl, R. Koch, M. Pollefeys, J. Denzler and L. Van Gool, Plenoptic Modeling and Rendering from Image Sequences taken by Hand-held Camera, Proc. DAGM’99, pp.94-101.

[25] A. Heyden and K. Åström, "Euclidean Reconstruction from Constant Intrinsic Parameters" *Proc. 13th International Conference on Pattern Recognition*, IEEE Computer Soc. Press, pp. 339-343, 1996.

[26] A. Heyden and K. Åström, "Euclidean Reconstruction from Image Sequences with Varying and Unknown Focal Length and Principal Point", *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, IEEE Computer Soc. Press, pp. 438-443, 1997.

[27] F. Kahl, "Critical Motions and Ambiuous Euclidean Reconstructions in Auto-Calibration", *Proc. ICCV*, pp.469-475, 1999.

[28] Koch, R., 1996, Automatische Oberflachenmodellierung starrer dreidimensionaler Objekte aus stereoskopischen Rundum-Ansichten, PhD thesis, University of Hannover, Germany, also published as Fortschritte-Berichte VDI, Reihe 10, Nr.499, VDI Verlag, 1997.

[29] R. Koch, M. Pollefeys and L. Van Gool, Multi Viewpoint Stereo from Uncalibrated Video Sequences. *Proc. European Conference on Computer Vision*, pp.55-71. Freiburg, Germany, 1998.

[30] R. Koch, M. Pollefeys, B. Heigl, L. Van Gool and H. Niemann. "Calibration of Hand-held Camera Sequences for Plenoptic Modeling", *Proc.ICCV'99 (international Conference on Computer Vision)*, pp.585-591, Corfu (Greece), 1999.

[31] R. Koch, B. Heigl, M. Pollefeys, L. Van Gool and H. Niemann, "A Geometric Approach to Lightfield Calibration", *Proc. CAIP99*, LNCS 1689, Springer-Verlag, pp.596-603, 1999.

[32] K. N. Kutulakos and S. M. Seitz, "A Theory of Shape by Space Carving," International Journal of Computer Vision, Vol. 38, No. 3, July 2000, pp. 199-218.

[33] S. Laveau and O. Faugeras, "Oriented Projective Geometry for Computer Vision", in : B. Buxton and R. Cipolla (eds.), *Computer Vision - ECCV'96*, Lecture Notes in Computer Science, Vol. 1064, Springer-Verlag, pp. 147-156, 1996.

[34] M. Levoy and P. Hanrahan, “Lightfield Rendering”, *Proc. SIGGRAPH ’96*, pp 31–42, ACM Press, New York, 1996.

[35] W. Lorensen and H. Cline. “Marching cubes: A high resolution 3D surface construction algorithm”. Computer Graphics (Proceedings of SIGGRAPH 87), 21(4):163–169, July 1987.

[36] D. Lowe, “Object recognition from local scale-invariant features”, *Proc. International Conference on Computer Vision*, 1999, pp. 1150-1157.

[37] B. Matei and P. Meer, “A general method for errors-in-variables problems in computer vision”, Proc. CVPR 2000, , IEEE Computer Society Conference on Computer Vision and Pattern Recognition, vol. 2, pp. 18–25, IEEE Computer Society Press, Los Alamitos, CA, (Hilton Head Island, South Carolina, June 13-15, 2000), 2000.

[38] L. McMillan and G. Bishop, “Plenoptic modeling: An image-based rendering system”, *Proc. SIGGRAPH’95*, pp. 39-46, 1995.

[39] E. Ofek, E. Shilat, A. Rappopport and M. Werman, “Highlight and Reflection Independent Multiresolution Textures from Image Sequences”, *IEEE Computer Graphics and Applications*, vol.17 (2), March-April 1997.

[40] M. Okutomi and T. Kanade. ”A multiple-baseline stereo.” IEEE Transactions on Pattern Analysis and Machine Intelligence, 15(4):353–363, 1993.

[41] M. Pollefeys, “Self-calibration and metric 3D reconstruction from uncalibrated image sequences”, Ph.D. dissertation, ESAT-PSI, K.U.Leuven, 1999.

[42] M. Pollefeys, R. Koch and L. Van Gool, “Self-Calibration and Metric Reconstruction in spite of Varying and Unknown Internal Camera Parameters”, *Proc. International Conference on Computer Vision*, Narosa Publishing House, pp.90-95, 1998.

[43] M. Pollefeys, R. Koch and L. Van Gool, ”A simple and efficient rectification method for general motion”, *Proc.ICCV’99 (international Conference on Computer Vision)*, pp.496-501, Corfu (Greece), 1999.

[44] M. Pollefeys, R. Koch and L. Van Gool. “Self-Calibration and Metric Reconstruction in spite of Varying and Unknown Internal Camera Parameters”, International Journal of Computer Vision, 32(1), 7-25, 1999.

[45] M. Pollefeys and L. Van Gool, “Stratified self-calibration with the modulus constraint”, *IEEE transactions on Pattern Analysis and Machine Intelligence*. Vol 21, No.8, pp.707-724, 1999.

[46] M. Pollefeys, F. Verbiest, L. Van Gool, ”Surviving dominant planes in uncalibrated structure and motion recovery”, A. Heyden, G. Sparr, M. Nielsen, P. Johansen (Eds.) Computer Vision - ECCV 2002, 7th European Conference on Computer Vision, Lecture Notes in Computer Science, Vol.2351, pp. 837-851. y

[47] W. Press, S. Teukolsky and W. Vetterling, *Numerical recipes in C: the art of scientific computing*, Cambridge university press, 1992.

[48] P. Rousseeuw, *Robust Regression and Outlier Detection*, Wiley, New York, 1987.

[49] H.. Sawhney, S. Hsu, R. Kumar, “Robust Video Mosaicing through Topology Inference and Local to Global Alignment”, Computer Vision - ECCV’98: Proc. 5th European Conference on Computer Vision, Vol. II, Lecture Notes in Computer Science, Springer-Verlag, pp. 103-119, 1998.y

[50] D. Scharstein and R. Szeliski. “A Taxonomy and Evaluation of Dense Two-Frame Stereo Correspondence Algorithms”. Internation Journal of Computer Vision 47(1/2/3):7-42, April-June 2002.

[51] C. Schmid and R. Mohr, “Local grayvalue invariants for image retrieval”, *IEEE Transaction on Pattern Analysis and Machine Intelligence*, 19, 5 (1997), pp. 530 534.

[52] W. Schroeder, J. Zarge, and W. Lorensen. “Decimation of triangle meshes”. Computer Graphics (Proceedings of SIGGRAPH 92), 26(2):65–70, July 1992.

[53] J. Shi and C. Tomasi, “Good Features to Track”, *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR’94)*, pp. 593 - 600, 1994.

[54] C. Slama, *Manual of Photogrammetry*, American Society of Photogrammetry, Falls Church, VA, USA, 4th edition, 1980.

[55] M. Soucy and D. Laurendeau, ”A general surface approach to the integration of a set of range views,” IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 17, no. 4, pp. 344–358, Apr. 1995.

[56] P. Sturm, “Critical Motion Sequences for Monocular Self-Calibration and Uncalibrated Euclidean Reconstruction”, *Proc. 1997 Conference on Computer Vision and Pattern Recognition*, IEEE Computer Soc. Press, pp. 1100-1105, 1997.

[57] P. Sturm, “Critical Motion Sequences for Monocular Self-Calibration and Uncalibrated Euclidean Reconstruction”, *Proc. 1997 Conference on Computer Vision and Pattern Recognition*, IEEE Computer Soc. Press, pp. 1100-1105, 1997.

[58] P. Sturm, “Critical Motion Sequences for the Self-Calibration of Cameras and Stereo Systems with Variable Focal Length”, In T. Pridmore and D. Elliman, editors, Pro- ceedings of the tenth British Machine Vision Conference, Nottingham, England, pages 63-72. British Machine Vision Association, pp. 63-72, 1999.

[59] C. Tomasi and T. Kanade, “Shape and motion from image streams under orthography: A factorization approach”, *International Journal of Computer Vision*, 9(2):137-154, 1992.

[60] P. Torr, *Motion Segmentation and Outlier Detection*, PhD Thesis, Dept. of Engineering Science, University of Oxford, 1995.

[61] P. Torr, A. Fitzgibbon, A. Zisserman, “Maintaining Multiple Motion Model Hypotheses Through Many Views to Recover Matching and Structure”. Proc. ICCV, pp. 485-491, 1998.

[62] B. Triggs, “The Absolute Quadric”, *Proc. 1997 Conference on Computer Vision and Pattern Recognition*, IEEE Computer Soc. Press, pp. 609-614, 1997.

[63] B. Triggs, P. McLauchlan, R. Hartley, A. Fiztgibbon, “Bundle Adjustment – A Modern Synthesis”, In B. Triggs, A. Zisserman, R. Szeliski (Eds.), *Vision Algorithms: Theory and Practice*, LNCS Vol.1883, pp.298-372, Springer-Verlag, 2000.

[64] G. Turk and M. Levoy ”Zippered Polygon Meshes from Range Images” Proceedings of SIGGRAPH ’94 pp. 311-318.

[65] T. Tuytelaars and L. Van Gool ”Wide Baseline Stereo based on Local, Affinely invariant Regions” *British Machine Vision Conference*, pp. 412-422, 2000.

[66] G. Van Meerbergen, M. Vergauwen, M. Pollefeys, L. Van Gool. ”A Hierarchical Symmetric Stereo Algorithm Using Dynamic Programming”, International Journal on Computer Vision 47(1/2/3): 275-285, 2002.

[67] M. Wheeler, Y. Sato, and K. Ikeuchi. “Consensus Surfaces for Modeling 3D Objects from Multiple Range Images.” *Sixth International Conference on Computer Vision*, pages 917–924, 1998.

[68] R. Willson, *Modeling and Calibration of Automated Zoom Lenses*, Ph.D. thesis, Department of Electrical and Computer Engineering, Carnegie Mellon University, January 1994.

[69] Z. Zhang, R. Deriche, O. Faugeras and Q.-T. Luong, “A robust technique for matching two uncalibrated images through the recovery of the unknown epipolar geometry”, *Artificial Intelligence Journal*, Vol.78, pp.87-119, October 1995.

List of Figures

List of Tables