Dense Stereo Correspondence for Uncalibrated Images in Multiple View Reconstruction

Willie Brink*, Daniek Joubert and François Singels
Applied Mathematics
Department of Mathematical Sciences
University of Stellenbosch, South Africa
*Email: wbrink@sun.ac.za

***Abstract*—We propose a method for combining dense stereo matching on calibrated images with automatic camera motion estimation, in order to generate dense reconstructions from uncalibrated image sequences. In doing so we explain how standard stereo matching can be extended, and for the estimation of motion we also introduce a technique for dealing with scale ambiguity and loop closure (a solution to the latter is termed déjà vu correction). Results from experiments are given and discussed, and we find that the proposed method can provide useful 3D information from uncalibrated image sequences.**

I. Introduction

Multiple view reconstruction is concerned with the building of a three-dimensional geometric model of some physical object, given a sequence of images taken of the object from various viewpoints. Techniques for retrieving and measuring shape information are useful for a wide variety of applications in autonomous robotics, industrial design and prototyping, human-computer interaction, augmented reality, medical imaging, archaeology, and many others. Digital cameras are often chosen to be used as capturing devices in these systems, particularly for their passiveness, portability, relatively low cost, high capturing speed and the potential richness of information in the output.

Most methods for performing object reconstruction from multiple images can be categorized broadly into shape-based methods and point-based methods. Shape-based methods attempt to extract the contours of an object from each view and combine these to build a 3D shape that is consistent with all of them [1]. Point-based methods search for corresponding points on the surface of the observed scene in two or more images and, if the relative translation and rotation of the camera between those images are known or can be established, then point locations in 3D are attainable through a triangulation procedure [2]. We will focus mainly on point-based methods but, towards the end of the paper, show that the incorporation of some shape information can be useful.

Point-based methods can be subdivided further into pixel-based methods and feature-based methods. The former attempts to find for every single pixel in an image a matching point in the other image, if it is visible in both views. Feature-based methods first detect salient, and therefore reliably matchable, features in every image independently and then match them across the different views. Pixel-based methods produce dense reconstructions, while feature-based ones would typically yield a sparser but more accurate set of points.

As mentioned, the motion parameters (translation and rotation) of each camera are needed. In many cases we can assume them to be known, which would imply that some form of calibration had to be performed [3] or that the movement of the camera was highly controlled [4]. Images for which the camera motions are known are said to be calibrated. In other cases, however, the images may not be calibrated. Agarwal *et al.* [5], for example, considered the intriguing problem of reconstructing parts of the city of Rome using a vast number of uncalibrated images from Flickr. In the uncalibrated case the camera motion parameters need to be estimated from the observed images before 3D structure of the object can be determined. A typical approach here is to match salient features in different images and, based on their 2D disparities between the views, infer relative camera motion.

Since camera motion estimation relies on feature detection and matching, approaches for reconstructing objects from uncalibrated images are normally feature-based. Dense reconstruction methods, on the other hand, are usually reserved for cases where calibrated images are available. In this paper we propose a method that essentially combines the two approaches in an effort to generate dense reconstructions from uncalibrated images. Effective means of combining the two have received relatively little attention in the literature to date, with the exception of work by Lhuillier and Quan [6].

We first provide some background of stereo geometry and matching in the calibrated case, then discuss a method for estimating camera motion parameters in the uncalibrated case. We propose a means to deal with the scale ambiguity that is inherently present and describe a simple way of performing so-called déjà vu correction on the estimated camera matrices. Our combination of motion estimation and stereo matching is explained, and results are presented from experiments on a test data sequence.

II. Stereo matching on calibrated images

Stereo vision is a widely studied approach that uses images captured by two synchronized cameras in order to infer depth of the observed scene. The problem of generating dense reconstructions from a pair of calibrated images amounts to finding for every pixel in the one image a matching pixel in

the other image. Although this problem has received much attention for a number of decades [7], it remains challenging.

This section describes in brief some geometric properties that can be exploited in order to constrain the search for correspondences, and the method of hierarchical dynamic programming that we find to be a good compromise between accuracy and speed.

A. Stereo geometry

Figure 1 depicts a typical stereo setup in which two cameras with optical centers $\mathbf{c}_1$ and $\mathbf{c}_2$ view a point $\mathbf{X}$ in 3D space. The point $\mathbf{X}$ projects onto the image plane of the first camera at point $\mathbf{x}_1$, and onto the second image plane at point $\mathbf{x}_2$. Note that the image planes are drawn in front of the optical centers, merely for ease of understanding.

In order to formulate the projections mathematically, we define two camera matrices as

$$\mathbf{P}_1 = \mathbf{K}_1\mathbf{R}_1\left[\mathbf{I} \mid -\mathbf{c}_1\right], \quad \mathbf{P}_2 = \mathbf{K}_2\mathbf{R}_2\left[\mathbf{I} \mid -\mathbf{c}_2\right]. \tag{1}$$

Here $\mathbf{R}_i$ is a 3×3 rotation matrix, and $\mathbf{c}_i$ a 3×1 translation vector, that relates the coordinate system of camera i with that of the world (in which $\mathbf{X}$ is defined). The 3×3 matrices $\mathbf{K}_1$ and $\mathbf{K}_2$ contain internal parameters of the two cameras which include focal lengths, possible offsets in image center and skewness factors [8]. It then follows that

$$\mathbf{x}_1 = \mathbf{P}_1\mathbf{X}, \quad \mathbf{x}_2 = \mathbf{P}_2\mathbf{X}, \tag{2}$$

where $\mathbf{x}_1$ and $\mathbf{x}_2$ are specified in 3D homogeneous coordinates, and $\mathbf{X}$ in 4D homogeneous coordinates.

Clearly, if for $\mathbf{x}_1$ the match $\mathbf{x}_2$ can be obtained, and if $\mathbf{P}_1$ and $\mathbf{P}_2$ are known, $\mathbf{X}$ can be determined. In the calibrated case, where $\mathbf{P}_1$ and $\mathbf{P}_2$ are known, the crucial problem of finding matching points between the two images remains.

Note in Fig. 1 that the plane passing through points $\mathbf{c}_1$, $\mathbf{c}_2$ and $\mathbf{x}_1$, called the epipolar plane, also passes through $\mathbf{x}_2$. This implies that the sought-after match for a point $\mathbf{x}_1$ in the first image must lie on the straight line, called the epipolar line, defined by the intersection of the epipolar plane and the second image plane. Moreover, this line is completely specifiable from $\mathbf{x}_1$ and the positions of the two camera centers.

The process of image rectification builds upon these search constraints by attempting to projectively transform the images in such a way that the epipolar lines are perfectly parallel and horizontal, as this would narrow the search for the match of a point in the one image down to a single image row in the other image. The image planes must therefore be transformed so that they are coplanar and parallel to the line through $\mathbf{c}_1$ and $\mathbf{c}_2$, and we need to find a suitable rotation matrix $\mathbf{R}_n$ and intrinsic calibration matrix $\mathbf{K}_n$ that will do this.

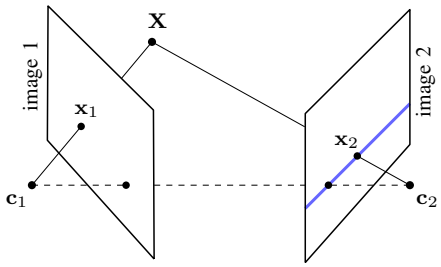


Fig. 1. A typical stereo configuration where a point $\mathbf{X}$ projects to coordinates $\mathbf{x}_1$ and $\mathbf{x}_2$ in two images respectively. The search for a match for $\mathbf{x}_1$ can be constrained to a single line through image 2.

$\mathbf{K}_n$ can be chosen arbitrarily, but a simple choice would be the average of $\mathbf{K}_1$ and $\mathbf{K}_2$. The rows of $\mathbf{R}_n$ are calculated as

$$\mathbf{r}_1 = \frac{\mathbf{c}_2 - \mathbf{c}_1}{||\mathbf{c}_2 - \mathbf{c}_1||}, \quad \mathbf{r}_2 = \frac{\mathbf{k} \times \mathbf{r}_1}{||\mathbf{k} \times \mathbf{r}_1||}, \quad \mathbf{r}_3 = \mathbf{r}_1 \times \mathbf{r}_2, \tag{3}$$

where $\mathbf{k}$ is the unit vector in the direction of the principal ray of camera 1, such that

$$\mathbf{R}_n = \begin{bmatrix}\mathbf{r}_1 & \mathbf{r}_2 & \mathbf{r}_3\end{bmatrix}^T. \tag{4}$$

The following transformation matrices are defined:

$$\mathbf{T}_1 = \mathbf{K}_n\mathbf{R}_n\mathbf{R}_1^T\mathbf{K}_1^{-1}, \quad \mathbf{T}_2 = \mathbf{K}_n\mathbf{R}_n\mathbf{R}_2^T\mathbf{K}_2^{-1}, \tag{5}$$

which implies that $\mathbf{x}_1' = \mathbf{T}_1\mathbf{x}_1$ provides the coordinates in the rectified image corresponding to $\mathbf{x}_1$ in the original image. Similarly, $\mathbf{x}_2' = \mathbf{T}_2\mathbf{x}_2$ is the rectified version of $\mathbf{x}_2$.

Since point correspondences between rectified images are known to occur on corresponding rows, the match for a pixel in one image can be represented by a single value, called its disparity. It is simply the horizontal shift that takes the pixel's current position to the position of its match.

An important issue to take note of is that of occlusions, which occur when some object or feature is visible in one image but not in the other. Assuming camera 1 is to the left of camera 2, we distinguish between left-occlusions (occluded from the left camera's point of view) and right-occlusions (occluded from the right camera's point of view).

B. Hierarchical dynamic programming (HDP)

When images can be rectified (i.e. when the images are calibrated) the stereo correspondence problem becomes a matter of matching every pair of coinciding rows pixel-wise in the two images. In order to accomplish this some way of measuring the dissimilarity between two pixels is needed. The smaller such a dissimilarity, the more likely it should be that two pixels are a good match. Options range from simple absolute differences, which would be quick to calculate but not particularly reliable, to more computationally taxing methods designed to yield better results [9].

In order to perform the matching between two image rows, we choose a hierarchical approach to dynamic programming because of the good balance between accuracy and computational efficiency. The method is explained in some detail in [10] and [11], and we provide a brief description here.

Given two rows of pixel values, one from each image, the first step would be to build a disparity space image (DSI). It is a matrix containing dissimilarity values for every possible disparity in some pre-specified range. Figure 2 shows a DSI for two synthetic image rows, where the absolute difference was chosen simply for illustration purposes.

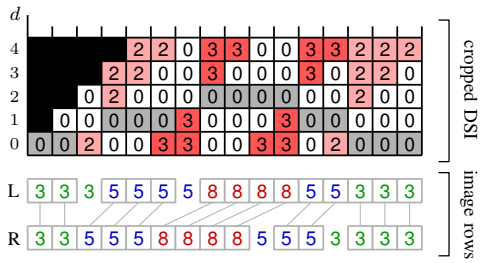


Fig. 2. A disparity space image (DSI) created from two synthetic image rows. The true matches are indicated in grey.

It is worth stressing that only non-negative disparities are considered at this stage, as it is assumed that camera 1 is to the left of camera 2 and their optical axes are parallel.

The aim is now to find some minimum cost path through the DSI, which then yields an optimal set of matches between the two image rows. A further restriction is put in place that prohibits the path from backtracking. It is referred to as the ordering constraint and implies that if one object appears to the left of another in image 1, it will also be to the left of that other object in image 2. This is not always true, if for example thin objects close to the cameras are present, but renders the optimization problem far more tractable.

Dynamic programming (DP) can be called upon for solving this optimization problem. It divides the problem into smaller subproblems recursively and can be implemented quite efficiently. A further significant decrease in execution time can be obtained by performing hierarchical DP. The images are downsampled several times, DP is applied to the lowest sampling level, its result is propagated to serve as an offset in the next higher level, and DP on this higher level is restricted to a band around the offset. The process is repeated up to the highest (hence original) sample level. This hierarchical approach is useful not only for increased speed but also provides some smoothness across the rows. Standard DP, on the other hand, considers every pair of rows independently and may therefore yield unwanted inconsistencies across the rows.

We will explain an extension of this approach that attempts to match corresponding epipolar lines in a sequence of uncalibrated images, but first a method for estimating camera matrices from such a sequence is discussed.

III. Estimating unknown camera motion

Next we consider a sequence of uncalibrated images. By this we mean that the extrinsic parameters, i.e. translation and rotation, of the cameras are unknown. We assume at this stage that the intrinsic parameters (focal length, camera center, etc.) are known. A simple method for finding these parameters for a single camera is explained in [12].

The feature-based approach discussed in this section follows a standard technique to find camera matrices between consecutive pairs of images. We then discuss an effective means of combining these relative camera motions to find positions and orientations of all the cameras in a single fixed coordinate system. Déjà vu correction and bundle adjustment are also discussed briefly.

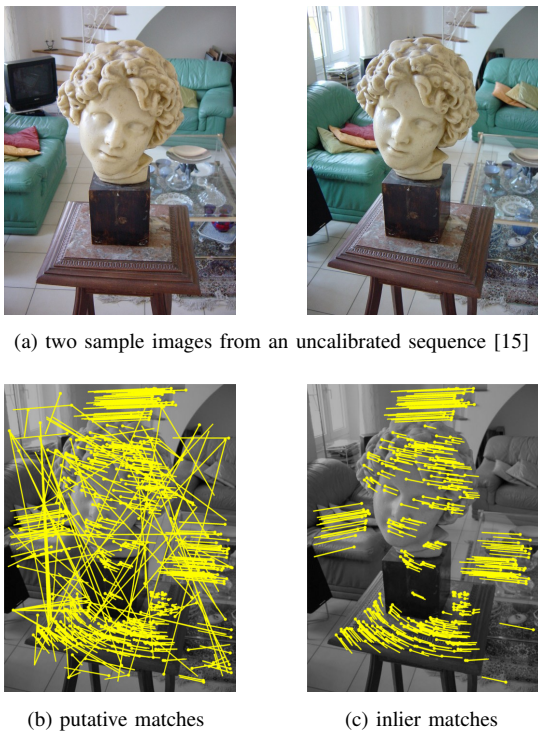
A. Feature detection and matching

As mentioned we follow a feature-based approach to find the relative rotation and translation between two consecutive cameras. The assumption is, of course, that the camera motion was small enough so that enough corresponding features are visible in the images.

The popular scale invariant feature transform (SIFT) [13] is an obvious choice for detecting and matching salient features in two images. Figure 3(b) shows example output, where every line segment indicates a match between image 1 and 2 (image 1 is shown in the background for reference). We observe that many incorrect matches are present. A stricter matching scheme may remove many of these but at the risk of losing some correct ones. Instead, an iterative RANSAC-based approach [14] can be followed in an effort to identify the largest set of matches that conforms to a physically possible camera motion. Since incorrect matches do not have a consistent structure they will not form such a set, and it should be possible to identify the correct set even when the number of incorrect matches far exceeds the number of correct ones.

B. Pairwise estimation of camera matrices

From a set of putative feature matches it is possible to find the relative camera motion from one image to the other. We observe that, for any pair of matching image coordinates $\mathbf{x}_1$



(a) two sample images from an uncalibrated sequence [15]

(b) putative matches (c) inlier matches

Fig. 3. Matching SIFT features determined for the two images shown on top, and the inliers obtained by RANSAC.

and $\mathbf{x}_2$, there exists a single 3×3 matrix $\mathbf{E}$ such that

$$\hat{\mathbf{x}}_2^T \mathbf{E} \hat{\mathbf{x}}_1 = 0, \tag{6}$$

with $\hat{\mathbf{x}}_1 = \mathbf{K}_1^{-1}\mathbf{x}_1$ and $\hat{\mathbf{x}}_2 = \mathbf{K}_2^{-1}\mathbf{x}_2$, where $\mathbf{K}_1$ and $\mathbf{K}_2$ are the internal calibration matrices of the two cameras. This matrix is called the essential matrix and, following from its definition, is specifiable up to scale. It is also known that $\mathbf{E}$ has a determinant of zero [8] hence there are 7 degrees of freedom and we need at least 7 matches to determine it.

The RANSAC-based approach we follow operates by choosing a set of 7 matches randomly from the available ones, calculating $\mathbf{E}$ from (6), and counting the number of other matches that agree with it. These inlier matches form a consensus set and the procedure is repeated until a large enough consensus set is obtained. An example of such a set is shown in Fig. 3(c).

The singular value decomposition (SVD) of the essential matrix is determined, and the resulting matrices are used to find the rotation matrix $\mathbf{R}$ and translation vector $\mathbf{c}$ that describes the motion of the camera from the first image to the second image (see [8] for details). It is important to be aware of a scale ambiguity that presents itself here. If no information about absolute scale is available it is not possible to determine the physical distance between the cameras. The translation vector is therefore usually normalized such that $||\mathbf{c}|| = 1$.

Camera motion estimation can be performed for consecutive pairs of images in a sequence and the motion of each new camera will then be determined relative to the previous one, such that the distances between cameras are normalized. The next section describes a technique that places all the cameras in a single coordinate system as well as a possible remedy for the unknown relative scale issue.

C. One coordinate system for all cameras

The motion estimation procedure described above finds a rotation matrix and translation vector for every camera, relative to the preceding camera. In order to use information from all the images for reconstruction, it is necessary to move these relative motions to one coordinate system.

Suppose $\mathbf{R}'_i$ and $\mathbf{c}'_i$ denote the rotation matrix and translation vector obtained from estimating the motion from camera $i-1$ to i, and that there are m images (hence cameras) in total. Since these parameters give the position and orientation of camera i relative to camera $i-1$, the following can be computed:

$$\mathbf{R}_1 = \mathbf{I}, \quad \mathbf{R}_2 = \mathbf{R}'_2, \quad \mathbf{R}_i = \mathbf{R}'_i \mathbf{R}_{i-1}, \tag{7}$$

$$\mathbf{c}_1 = \mathbf{0}, \quad \mathbf{c}_2 = \mathbf{c}'_2, \quad \mathbf{c}_i = \alpha_i \mathbf{R}_{i-1}^T \mathbf{c}'_i + \mathbf{c}_{i-1}, \tag{8}$$

for $i = 3, \ldots, m$. It gives the rotation matrix $\mathbf{R}_i$ and translation vector $\mathbf{c}_i$ associated with camera i. The values of α_i are not yet specified, and indicate the scale factors that need to be corrected for (recall that the pairwise motion estimation procedure fixes the distance between a pair of cameras arbitrarily to be 1). The overall scale cannot be established, for the same reasons as mentioned above, but it should be possible to fix distances between the cameras relative to, say, the distance between camera 1 and camera 2.

Suppose $\mathbf{x}_{i-2}$, $\mathbf{x}_{i-1}$ and $\mathbf{x}_i$ denote image coordinates in images $i-2$, $i-1$ and $i > 2$ respectively, such that $\mathbf{x}_{i-2}$ and $\mathbf{x}_{i-1}$ was identified as an inlier match as was $\mathbf{x}_{i-1}$ and $\mathbf{x}_i$. Clearly, these two pairs of image coordinates that form matches must triangulate to the same point in space, say $\mathbf{X}$. Our aim is to fix the scale between cameras $i-1$ and i, given that the scale is already fixed for cameras $i-2$ and $i-1$, so that the two triangulated points coincide.

To this end, let $\mathbf{X}$ be the point resulting from triangulating $\mathbf{x}_{i-2}$ and $\mathbf{x}_{i-1}$ and let $\mathbf{P}_i = \mathbf{K}_i \mathbf{R}_i \begin{bmatrix} \mathbf{I} & | & -\mathbf{c}_i \end{bmatrix}$. We force the point triangulated from $\mathbf{x}_{i-1}$ and $\mathbf{x}_i$ to coincide with $\mathbf{X}$ so that, by virtue of (2) and (8),

$$\mathbf{0} = \mathbf{x}_i \times \mathbf{P}_i \mathbf{X} = \hat{\mathbf{x}}_i \times \mathbf{R}_i (\tilde{\mathbf{X}} - \alpha_i \mathbf{R}_{i-1}^T \mathbf{c}'_i - \mathbf{c}_{i-1}), \tag{9}$$

where $\hat{\mathbf{x}}_i = \mathbf{K}_i^{-1}\mathbf{x}_i$ and $\tilde{\mathbf{X}}$ is the Euclidean version of the homogeneous vector $\mathbf{X}$, so that $\mathbf{X} = [\tilde{\mathbf{X}}^T, \ 1]^T$. Therefore

$$\left[\hat{\mathbf{x}}_i \times \mathbf{R}_i (\tilde{\mathbf{X}} - \mathbf{c}_{i-1})\right] - \left[\hat{\mathbf{x}}_i \times \alpha_i \mathbf{R}_i \mathbf{R}_{i-1}^T \mathbf{c}'_i\right] = \mathbf{0}, \tag{10}$$

yielding

$$\alpha_i \left(\hat{\mathbf{x}}_i \times \mathbf{R}'_i \mathbf{c}'_i\right) = \mathbf{x}_i \times \mathbf{R}_i (\tilde{\mathbf{X}} - \mathbf{c}_{i-1}), \tag{11}$$

from which α_i can be determined so that $\mathbf{c}_i$ can be found. We would typically compute a scale factor for every available set of matches that overlaps between the three images and choose a final value for α_i as some average of the results.

D. Déjà vu correction

Quite frequently, particularly in multi-view reconstruction scenarios, the camera makes a loop around the object of interest or, similarly, the object is placed on a turntable and undergoes a full rotation. The position of the last camera is then typically close to the first one, allowing pairwise motion estimation to be performed on camera m and camera 1. However, because the first camera is fixed at the origin by equations (7) and (8) and subsequent pairwise estimation is subject to drift, the position and orientation of camera 1 relative to that of camera m may yield an inconsistency.

Alleviating this problem is referred to as déjà vu correction in autonomous navigation (the robot realizes it has been at some place before, which clashes with its believed location, and updates its position estimation history accordingly).

We propose the following simple correction. Camera motion parameters $\mathbf{R}_{i,1}$ and $\mathbf{c}_{i,1}$ are determined for cameras $1, 2, \ldots, m$ in that order. We then also apply the method to find the motion from image m to image 1, obtain parameters, say, $\mathbf{R}_{m+1}$ and $\mathbf{c}_{m+1}$, and let

$$\mathbf{R}_{i,2} = \mathbf{R}_{m+1}^T \mathbf{R}_{i,1}, \quad \mathbf{c}_{i,2} = \mathbf{c}_{i,1} - \mathbf{c}_{m+1}. \tag{12}$$

Note that these new parameters are equivalent to ones that would have been obtained by estimating motion for cameras $1, m, m-1, \ldots, 2$ in that order.

It then remains to combine $\mathbf{c}_{i,1}$ and $\mathbf{c}_{i,2}$, as well as $\mathbf{R}_{i,1}$ and $\mathbf{R}_{i,2}$, in some sensible way. For that we have experimented with

$$\mathbf{c}_i = \left(\tfrac{m+1-i}{m}\right)\mathbf{c}_{i,1} + \left(\tfrac{i-1}{m}\right)\mathbf{c}_{i,2}. \quad (13)$$

$\mathbf{R}_i$ is obtained similarly as a weighted average of $\mathbf{R}_{i,1}$ and $\mathbf{R}_{i,2}$ (since the matrix must remain orthogonal, rotation parameters are extracted from $\mathbf{R}_{i,1}$ and $\mathbf{R}_{i,2}$, and their weighted averages are used to determine $\mathbf{R}_i$). The weights specified in (13) place more importance on parameters that were calculated sooner in the queue, which should then minimize drift.

E. Bundle adjustment

In a final attempt to increase the accuracy in the estimated positions and orientations of the cameras, a technique known as bundle adjustment (BA) [16] can be applied.

For n distinct points in space viewed by m cameras BA adjusts all the camera motion parameters and 3D coordinates of reconstructed points, in order to minimize the sum of squared differences between the actual locations of features in the images (as determined by SIFT, for example) and the locations obtained from re-projecting the 3D points onto the image planes. The Levenberg-Marquardt algorithm [17], [18] has proven to be extremely successful in solving this nonlinear optimization problem.

IV. Dense matching on uncalibrated images

We now discuss our technique of utilizing dense stereo matching algorithms in an uncalibrated scenario where a sequence of images is given, with no camera motion information available. The idea is quite straightforward. Camera motion is estimated for the sequence by the method described, images are rectified, and dense stereo matching is performed on the rectified images. There are, however, some slight adaptations that need to be made.

A. Adapting the stereo matching algorithm

Pairs of images are rectified by the transformations given in (5), rendering the epipolar lines horizontal. A stereo matching algorithm can now be performed on the two rectified images, but two issues should be taken note of.

Firstly, image coordinates may now be negative, to avoid unwanted cropping of the rectified images. Also, image data in a row no longer starts at some fixed column, and these offsets for the different rows should be accounted for.

Secondly, provision should be made for negative disparities. It is clear from Fig. 3(c) that some corresponding features may shift to the left from one image to the other while others shift to the right. We allow for negative disparities by extending the DSI to also contain negative disparities, and allow an optimal path through the DSI to cross the zero-disparity axis.

B. Match propagation

The fact that a sequence of images (not only two) is available should be exploited. We have used this fact to our advantage somewhat, in fixing the relative scale changes between pairs of cameras. Normally in stereo vision a matching pair of coordinates $\mathbf{x}_{i-1}$ and $\mathbf{x}_i$ is triangulated in order to obtain a point in space. However, now that a dense set of matches is at our disposal for all pairs of consecutive images, every match under consideration can be propagated forwards and backwards through the sequence until occlusions are reached.

This procedure results in a sequence of pairwise matches, all corresponding to the same feature in space (assuming that the stereo algorithm was successful). Each one of these matches provides a candidate 3D position. We combine them, for example by taking a median to decrease the occurrence of obviously incorrect outliers, and arrive at a single point.

C. Incorporating image segmentation

A further improvement in accuracy and quality of the reconstruction is attainable from the inclusion of segmentation information. A clear distinction between pixels belonging to the object and those belonging to the background would be extremely useful in constraining the stereo algorithm to match foreground segments only of every corresponding pair of rows. Moreover, accuracy of the output will be improved and computational cost will be lowered.

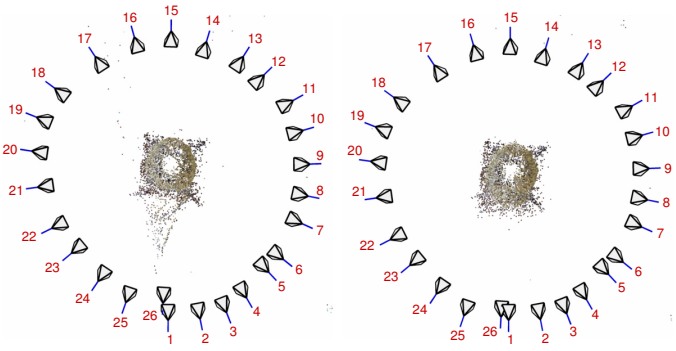
Of course, segmenting arbitrary images into foreground and background is by no means trivial, and this improvement in accuracy and speed comes at the cost of having to implement a sufficiently robust segmentation algorithm.

V. Results

We implemented the methods described above and present here some experimental results obtained.

The test data used was obtained from [15] and consists of 26 images of a marble sculpture, two of which are shown in Fig. 3(a). The exact movement was uncontrolled during capturing but the camera more-or-less followed a ring around the object of interest and ended up in a position close to where the first image was taken.

Figure 4(a) depicts the estimated camera positions we obtained from the pairwise approach described in section III. Images were processed counterclockwise from camera 1 as shown. Because of this pairwise estimation of motion we



(a) before DVC (b) after DVC

Fig. 4. Top-down view of the estimated positions and orientations of the cameras, and reconstructed features, for the test sequence (a) before déjà vu correction and (b) after. Note in particular the positions of camera 26.

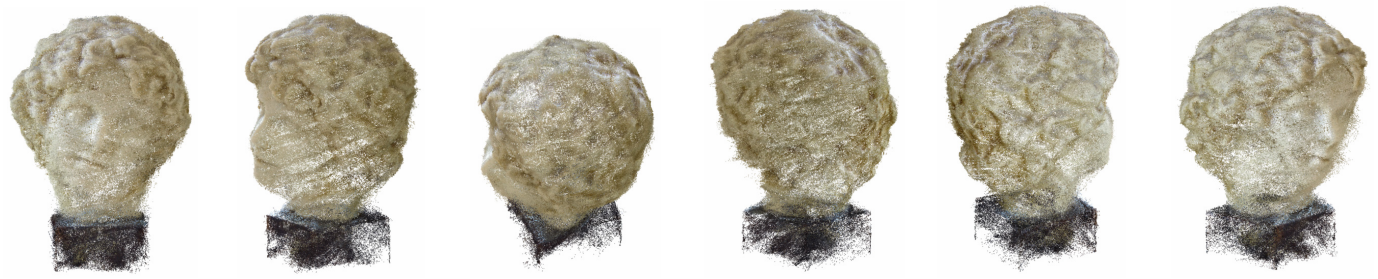


Fig. 5. Final reconstruction of the marble sculpture, as viewed from various angles. The 3D point cloud consists of 681,237 vertices and was obtained from 26 images taken all around the object with a hand-held camera (dataset available at [15]). Background points have been cropped for the sake of clarity.

should expect drift but, rather surprisingly, the last camera's estimated position is fairly close to where we expect that it should be. The slight drift does influence the reconstruction negatively, however, as can be seen in the reconstructed points.

The camera configuration in Fig. 4(b) is the result of our déjà vu correction procedure, whereby camera motions are estimated in the opposite direction (starting from 1 and moving clockwise), and combined with those obtained previously in a weighted manner. Note that the obvious errors in the reconstruction have been reduced significantly.

Once camera motions are known, points can be triangulated. A final reconstruction is shown in Fig. 5, from different viewpoints. We also applied the technique of match propagation described, both forwards and backwards through the image sequence, and it appears to remove most erroneous points that result from the triangulation of incorrect matches.

It should be realized that the reconstructed point cloud shown in the figure is raw output from the algorithm and could be perfectly suited for any number of different post-processing procedures (such as smoothing and surface fitting).

VI. Conclusion

We have presented a method for generating dense reconstructions from uncalibrated image sequences. The method estimates relative motion pair-wise, and transforms all the estimated camera poses to a single coordinate system. The scale ambiguity is resolved by forcing overlapping matches across different pairs of images to triangulate to equivalent points in space. We also introduced a procedure for correcting drift in the case of loop closure that seemed to work well. Stereo matching is performed on rectified pairs of images and we recommended a hierarchical version of dynamic programming to find an optimal set of matches for each pair of corresponding epipolar lines in two images. It is not without fault but our match propagation procedure seems to eliminate many incorrectly triangulated 3D points.

It is important to stress that we have not tested the accuracy of the proposed method, as no ground truth was available for the test set, and further investigation is needed. However we have demonstrated that the method can be successful in providing useful 3D information from an uncalibrated sequence of images.

In future we hope to move away from the requirement that the internal camera calibration parameters are known, and rather attempt to estimate them from the image data. The combination of the proposed method with image segmentation is another exciting prospect.

References

[1] R. Szeliski, *Shape from rotation*, IEEE Computer Vision and Pattern Recognition, 2:625–630, 1991.

[2] S. Seitz and C. Dyer, *Complete structure from four point correspondences*, IEEE International Conference on Computer Vision, 330–337, 1995.

[3] J. Heikkilä, *Geometric camera calibration using circular control points*, IEEE Transactions on Pattern Analysis and Machine Intelligence, 22(10):1066–1077, 2000.

[4] V. Fremont and R. Chellali, *Turntable-based 3D object reconstruction*, IEEE Conference on Cybernetics and Intelligent Systems, 1277–1282, 2004.

[5] S. Agarwal, N. Snavely, I. Simon, S. Seitz and R. Szeliski, *Building Rome in a day*, IEEE International Conference on Computer Vision, 72–79, 2009.

[6] M. Lhuillier and L. Quan, *A quasi-dense approach to surface reconstruction from uncalibrated images*, IEEE Transactions on Pattern Analysis and Machine Intelligence, 27(3):418–433, 2005.

[7] D. Scharstein and R. Szeliski, *A taxonomy and evaluation of dense two-frame stereo correspondence algorithms*, International Journal of Computer Vision, 47:7–42, 2002.

[8] R. Hartley, A. Zisserman, *Multiple View Geometry*, 2nd edition, Cambridge University Press, 2003.

[9] S. Birchfield and C. Tomasi, *A pixel dissimilarity measure that is insensitive to image sampling*, IEEE Transactions on Pattern Analysis and Machine Intelligence, 20(4):401–406, 1998.

[10] G. van Meerbergen, M. Vergauwen, M. Pollefeys and L. van Gool, *A hierarchical symmetric stereo algorithm using dynamic programming*, International Journal of Computer Vision, 47:275–285, 2002.

[11] F. Singels, *Real-time stereo reconstruction through hierarchical dynamic programming and LULU filtering*, Master's Thesis, University of Stellenbosch, South Africa, 2010.

[12] Z. Zhang, *A flexible new technique for camera calibration*, IEEE Transactions on Pattern Analysis and Machine Intelligence, 22(11):1330–1334, 2000.

[13] D. Lowe, *Object recognition from local scale invariant features*, International Conference on Computer Vision, 1150–1157, 1999.

[14] M. Fischler and R. Bolles, *Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography*, Communications of the ACM, 24(6):381–395, 1981.

[15] M. Lhuillier and L. Quan, `http://www.cs.ust.hk/~quan/WebPami/pami.html`

[16] B. Triggs, P. McLauchlan, R. Hartley and A. Fitzgibbon, *Bundle adjustment: a modern synthesis*, Vision Algorithms: Theory and Practice, Springer-Verlag, 2000.

[17] K. Levenberg, *A method for the solution of certain problems in least squares*, Quarterly of Applied Mathematics, 2:164–168, 1994.

[18] D. Marquardt, *An algorithm for least squares estimation on nonlinear parameters*, SIAM Journal of Applied Mathematics, 11:431–441, 1963.