Dense Multiple View Stereo with General Camera Placement using Tensor Voting

Philippos Mordohai and Gérard Medioni
Institute for Robotics and Intelligent Systems,
University of Southern California,
Los Angeles, CA 90089, USA
{mordohai, medioni}@iris.usc.edu

Abstract

We present a computational framework for the inference of dense descriptions from multiple view stereo with general camera placement. Thus far research on dense multiple view stereo has evolved along three axes: computation of scene approximations in the form of visual hulls; merging of depth maps derived from simple configurations, such as binocular or trinocular; and multiple view stereo with restricted camera placement. These approaches are either sub-optimal, since they do not maximize the use of available information, or cannot be applied to general camera configurations. Our approach does not involve binocular processing other than the detection of tentative pixel correspondences. We require calibration information for all cameras and that there exist camera pairs which enable automatic pixel matching. The inference of scene surfaces is based on the premise that correct pixel correspondences, reconstructed in 3-D, form salient, coherent surfaces, while wrong correspondences form less coherent structures. The tensor voting framework is suitable for this task since it can process the very large datasets we generate with reasonable computational complexity. We show results on real images that present numerous challenges.

1 Introduction

Reconstruction of three-dimensional scenes from sets of images is a fundamental problem in computer vision. The minimum number of images required for reconstruction, be it projective or metric, is two. The major challenge is the establishment of pixel correspondences, which is a core problem that has received extensive attention since the early days of computer vision. For a review of research on binocular stereo, the case of exactly two images, interested readers should refer to the work of Scharstein and Szeliski [1]. Even though one might justifiably argue that the core issues of stereo vision are addressed in the binocular case, the generalization to multiple images is not straightforward. Note that by "multiple images or views" we always refer to cases where the number of images is greater than two.

The additional difficulties in the multiple image case stem from insufficiencies of either the representation or the computational framework. View-centered representations are inadequate for general camera placement, while the computational complexity associated with processing a possibly very large number of pixels is prohibitive for certain methodologies. To bypass these difficulties, researchers have taken different paths. One such path is requiring all cameras to be "on the same side" of the scene. This class of approaches includes multi-baseline stereo and other approaches that have one or more privileged images, for which a depth map is computed and validated using the remaining images. The $2\frac{1}{2}$-D, view-centered representation is sufficient for such configurations. A different approach is to adopt a world-centered representation but restrict all processing to a limited number of features to keep computational complexity manageable. Once the structure of the feature points has been estimated, dense stereo is performed on image pairs. Conversely, other researchers compute depth maps from image pairs and merge the results. Both these approaches do not utilize all the images where a world point appears at the same time. Recently, a considerable amount of work has been devoted to computing scene approximations, instead of reconstructions, in the form of "visual hulls", inspired by Laurentini [2] and by Seitz and Dyer [3]. Section 2 contains a more extensive review of multiple-view reconstruction methods.

We present an approach (a preliminary implementation of which appeared in [4]) that allows truly general camera placement, employs a world-centered representation and is able to process large numbers of potential pixel matches, typically in the order of a few millions, efficiently. No images are privileged and features are not required to appear in more than two views. The restriction on camera placement is that cameras must be placed in pairs in such a way that for each camera there exists at least another one with a similar viewpoint that allows for automatic correlation-based dense pixel matching. One could place such cameras pairs arbitrarily in space with no other considerations for image overlap or relationships between the locations of camera centers and the scene. Moreover, unlike other leading multiple view reconstruction methods [5][6][7][8][9], we do not segment and discard the "background" but attempt to reconstruct it together with the foreground. Camera calibration information and a common world coordinate frame for all cameras have to be provided. We cast the problem as one of perceptual organization of potential pixel matches reconstructed in 3-D. The premise is that correct potential matches should form coherent surfaces, while erroneous potential matches should not align in ways that form surfaces that are as salient as those formed by the correct matches. The tensor voting framework [10] provides an effective, computationally efficient means for achieving this type of perceptual organization even under severe noise contamination.

We present results on challenging datasets captured for the Virtualized Reality project of the Robotics Institute of Carnegie Mellon University and distributed freely at http://www-2.cs.cmu.edu/virtualized-reality. Besides the people that appear in the images, we also reconstruct the floor and the visible parts of the dome. To our knowledge, no reconstructions of these scenes, using static images only, have been published, even thought the data have been available for a few years. Some of the input images can be seen in Fig. 1. The outputs presented in these paper are in the form of 3-D point clouds. Inference of surfaces from such data is out of the scope of this paper and can be addressed as in [11][12].

The paper is organized as follows. Related work is presented in the next section, an overview of the approach is presented in Section 3, the initial processing steps in Section 4, and the tensor voting framework in Section 5. Experimental results are shown in Section 6, while a summary and perspectives are given in Section 7.

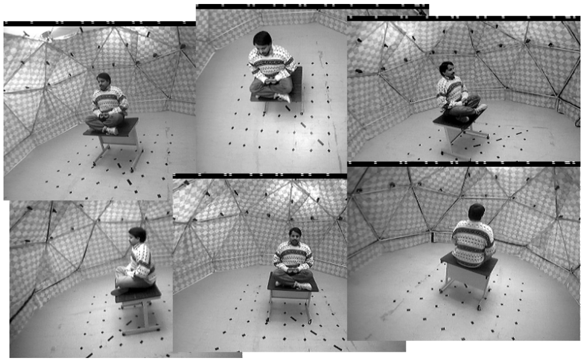


Figure 1: Some of the input images of the "meditation" set captured at the CMU dome. Some of the cameras are visible in each image

2 Related Work

Early research on multi-baseline stereo includes the work of Okutomi and Kanade [13] who use both short and long baselines, which offer different advantages, from a set of cameras perfectly placed on a straight line with parallel optical axes. A layer based approach that can handle occlusion and transparency was presented by Szeliski and Golland [14]. The assumption is that the cameras are sufficiently far away from the scene so that it can be represented as a set of planes in 3-D. The number of these layers, however, is hard to estimate a priori and initialization of the algorithm is difficult. Kang *et al.* [15] advance the state of the art in multi-baseline stereo by taking occlusion into account when selecting the subset of images in which each pixel can be matched. The representation in all these approaches is view-centered and dense disparity maps are produced for one or more privileged images.

Approaches that employ 3-D representations are also numerous. They fall into two categories: those that reconstruct sparse features initially and then use binocular stereo to produce dense depth estimates; and those that merge the view-centered depth maps that have been produced from binocular or multi-baseline stereo. A landmark approach of the former category is the work of Pollefeys *et al.* [16] that performs self-calibration of the cameras, even in the case of varying internal parameters, based on a set of sparse features and then uses binocular matching to complete the models. Lhuillier and Quan [17] increase the robustness of scene and camera geometry estimation by using "quasi-dense" instead of sparse features. Then, the results from image pairs and triplets are merged to produce the final reconstruction. Other approaches the fall in the category of merging depth maps, include the work of Fua [11], Narayanan *et al.* [18] and Taylor [19]. We argue that even though these schemes are effective, the fact that they do not utilize all the available information during binocular processing does not allow them to achieve the best possible quality in the produced depth maps, and thus the merged surfaces are also not as good as possible.

Recently, volumetric multiple view methods have attracted a lot of attention from the computer vision community. Kutulakos and Seitz [6] introduced the space carving framework which is based on the notion of "photoconsistency" to derive the visual hull of the scene, which is an approximation of shape, from a set of images. Voxels from an initial volume, which must contain the scene, are progressively carved away if their projections on the images are not photoconsistent, that is if they exhibit large color variations. This indicates that the voxel under consideration is empty and the projections come from voxels occluded by it, which are revealed once it is removed from the volume. The formulation of [6] alleviates the camera placement restrictions of [3] and the only assumption is that of "free space", which states that the photoconsistent scene must completely lie within an arbitrary volume, which must not contain the optical centers of the cameras. A limitation of the original space carving algorithm is that it cannot recover from the erroneous carving of a voxel which leads to further carving into the volume since photoconsistency is violated by voxels behind the actual surface. Techniques that have been proposed to address this problem can be found in recent surveys of volumetric methods such as [20] and [21].

Outstanding results on binocular stereo have been achieved using graph cuts. Kolmogorov and Zabih [8] extended the framework to multiple images. The problem is formulated as finding the approximate minimum of a global energy function via graph cuts. The framework treats all images equally, takes visibility into account and most importantly enforces *smoothness* to the solution while preserving boundaries. Smoothness in the form of spatial coherence of the reconstructed surfaces is a critical limitation of the volumetric methods of the previous paragraph, which operate at the pixel-voxel correspondence level without imposing any form of smoothness in the resulting surfaces. While the cameras can be placed according to the restrictions of voxel coloring [3], results are only presented for examples in which all the cameras lie on a plane looking at the same direction. Furthermore, computational complexity is quadratic with respect to the number of possible labels (depth layers), which has to be kept low (16 in the experiments presented in [8]).

Variational approaches have also been proposed for the problem of the multiple view stereo with excellent results. Faugeras and Keriven [5] pose multiple view stereo in a variational framework where an initial surface evolves according to image correspondence criteria. Cross-correlation on the tangent plane of the surface, instead of regular square windows which imply fronto-parallel surfaces, is used as the similarity measure. Yezzi and Soatto [7] address a different type of scenes in which surfaces have smooth or constant albedo, or fine homogeneous texture. To avoid the difficulties associated with the computation of image derivatives, they do not rely on local correspondence, but on region similarity measures which are very effective for the types of objects they handle. Jin *et al.* [9] extend the framework to handle any type of surface including non Lambertian ones by imposing a rank constraint on the radiance tensor. They achieve excellent results, but are limited, as are the previous two approaches, by the "blue sky" assumption [7] regarding the background which must be segmented and discarded.

Our work differs in that camera pairs can be arbitrarily placed, as long as calibration information is provided or computed as in [16], without any requirements for free space or blue sky type background. The proposed approach utilizes all available information simultaneously for reconstruction and does not approximate the scene surfaces. It is important to note that we do not make any decisions about pixel correspondence at the binocular level. We only make hypotheses for potential correspondences.

3 Overview of the Approach

The input to our algorithm is a set of images of a static scene and complete calibration information (both internal and external parameters). The output consists of a set of 3-D points that belong on the scene surfaces. Processing entails the following steps:

- selection of image pairs and rectification to align conjugate epipolar lines
- initial binocular matching to generate matching candidates
- reconstruction of matching candidates in 3-D
- tensor voting on the point cloud to infer surface inliers.

Tensor voting [10] is an effective framework to infer perceptual structures, such as surfaces, curves and junctions, from noisy data. For the problem at hand we can assume that each pixel is a projection of an elementary scene surface patch (surfel) and that the only structure types we are likely to encounter are surfaces and surface intersections, were abrupt changes of surface orientation occur. Each matching candidate is reconstructed as a 3-D point and encoded as a second order tensor that contains the point's orientation information. During the voting process, tensors cast to their neighbors votes that can be interpreted as support for an orientation at the receiver consistent with that of the voter. Therefore, points that are aligned to form coherent smooth surfaces (not necessary planar, or of any other parametric form) reinforce each other's orientation estimate and develop a preference for a certain surface orientation. We assume that these are the true scene surfaces that are formed by the correct matching candidates. On the other hand, points that were derived from wrong matches are, if not random, not aligned to form surfaces as smooth and coherent as the correct ones. Thus, these points receive contradicting votes and do not develop a strong preference for a particular surface orientation. We propose to use surface saliency, the amount of support points receive as being inliers of a smooth surface, as the measure to select the correct match among all the candidates on each line of sight.

4 Initial Processing Steps

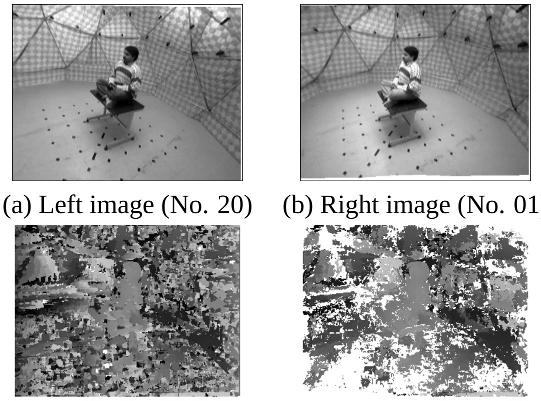
In this section we describe the processing stages that precede the main tensor voting stage.

Image pair selection and rectification As can be seen in Fig. 1, the input images suffer from severe radial distortion. The first step, therefore, is to correct the distortion using the provided κ_1 coefficient. The next step is the selection of image pairs taken from similar viewpoints. Only when the images are taken from a similar angle and a similar distance is automatic pixel matching possible. Otherwise scaling and perspective effects make the matching process really hard. This is why we need pairs of images from similar viewpoints to proceed. Once the image pairs have been selected (manually, even though automatic selection would not have been very complicated to implement), the images are rectified in pairs so that all epipolar lines in both images become horizontal and conjugate epipolar lines share the row coordinate. This is accomplished using the calibration information, but no correspondences, by our implementation of the work of Gluckman and Nayar [22]. An example pair can be seen in Fig. 2(a-b). Note that the effects of radial distortion have not been entirely removed close to the image boundaries, and that in these areas corresponding features do not lie on conjugate epipolar lines. Each image can be used in more than one pair, but rectification has to be performed separately for each pair.

Generation of matching candidates The input to this stage is rectified image pairs. The output is matching candidates reconstructed in 3-D world coordinates, labeled according to the line of sight (going through the optical center and the image pixel) they belong to. It can be viewed as binocular processing, even though no decisions are made and the objective is the inference of potential 3-D points that belong to the scene.

Matching candidates are generated for each image pixel of the "left" or reference image of each pair by a 7×7 normalized cross-correlation window. We apply a simple parabolic fit using the neighboring values of peaks in the correlation function to achieve subpixel precision. All significant peaks of cross-correlation are retained as matching candidates. As a result, more than one candidate can be generated for each pixel. Figure 2(c) shows a disparity map of the matching candidates, where candidates for the same pixel with larger disparity are written on top of ones with smaller disparity, for one pair of the "meditation" set.

Figure 2(d) shows an attempt to process the pair in the same way we process datasets generated by multiple pairs in the remainder of the paper. The results are not satisfactory since wrong matches caused by occlusion (around the head of the person for instance), misalignment due to radial distortion (close to the borders) and lack of texture (on the floor) cannot be discriminated from the correct ones. However, when processing is done using all matching candidates generated from all image pairs, consistent matches form surfaces that stand out among the clutter. We consider this a significant advantage of processing all data simultaneously over merging inaccurate depth maps produced binocularly. The next section discusses how these surfaces can be inferred using tensor voting.



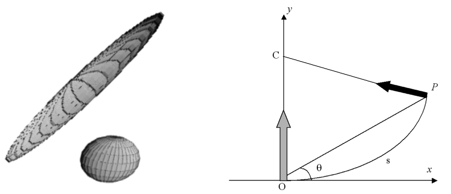
(a) Left image (No. 20) (b) Right image (No. 01)

(c) Matching candidates in disparity space (d) Disparity map after binocular processing

Figure 2: Two images of a rectified pair (a-b), the generated matching candidates in disparity space (c) and an attempt to produce a disparity map with binocular processing (d) (lighter intensity corresponds to larger disparity, white indicates no match)

5 Tensor Voting

The use of a voting process for structure inference from sparse and noisy data was presented in [10]. The methodology is non-iterative and robust to considerable amounts of outlier noise. It has one free parameter: the scale of voting σ, which essentially defines the size of the neighborhood of each point. The input data is encoded as second-order symmetric tensors, and constraints, such as proximity, co-linearity and co-curvilinearity are propagated by voting within the neighborhood. The tensors allow the representation of points on smooth surfaces, surface intersections, curves and junctions, without having to keep each type in separate spaces. In 3-D, a second-order tensor has the form of an ellipsoid, or equivalently of a 3×3 matrix, whose eigenvectors correspond to the axes of the ellipsoid and whose eigenvalues correspond to their lengths. Its shape encodes the type of feature that it represents, while its size the *saliency* or the confidence we have in this information



(a) Example tensors (b) Vote generation

Figure 3: Tensor Voting. (a) The shape of the tensor indicates if there is a preferred orientation, while its size corresponds to the confidence of this information. The top tensor has a strong preference of orientation and is more salient than the bottom tensor. (b) Vote generation as a function of the distance and curvature of the arc and the orientation of the voter

(Fig. 3(a)).

Tensors communicate information to their neighbors in the form of *votes* which are also second order tensors. Their shape corresponds to the orientation the receiver would have, if the voter and receiver were in the same structure. The saliency (strength) of a vote cast by a unitary stick tensor (an elementary surface) decays with respect to the length of the smooth circular path connecting the voter and receiver, according to the following equation:

$$S(s, \kappa, \sigma) = e^{-(\frac{s^2 + c\kappa^2}{\sigma^2})} \tag{1}$$

Where s is the length of the arc between the voter and receiver, and κ is its curvature (see Fig. 3(b)), σ is the scale of voting, and c is a constant.

As shown in Fig. 3(b), the vote from the voter O (which has a surface *normal* orientation indicated by the thick vector) to the receiver P has the orientation that P would have had, if O and P indeed were in the same smooth surface, which is not restricted to being planar. If more points were consistent with this surface, P would receive a number of votes with similar preference for orientation. Thus, it would develop high *surface saliency* and, thus, it would be inferred as a surface inlier, while an estimate of its orientation would also be available.

In practice, vote accumulation is performed by tensor addition, which is equivalent to the addition of 3×3 matrices. After voting is complete, the eigensystem of each tensor is analyzed and the tensor is decomposed as in:

$$\begin{aligned} T = \lambda_1 \hat{e}_1 \hat{e}_1^T + \lambda_2 \hat{e}_2 \hat{e}_2^T + \lambda_3 \hat{e}_3 \hat{e}_3^T = \\ = (\lambda_1 - \lambda_2) \hat{e}_1 \hat{e}_1^T + (\lambda_2 - \lambda_3)(\hat{e}_1 \hat{e}_1^T + \hat{e}_2 \hat{e}_2^T) \\ + \lambda_3 (\hat{e}_1 \hat{e}_1^T + \hat{e}_2 \hat{e}_2^T + \hat{e}_3 \hat{e}_3^T) \end{aligned} \tag{2}$$

where λ_i are the eigenvalues in decreasing order and $\hat{e}_i$ are the corresponding eigenvectors. Whether a point belongs to a smooth perceptual structure is determined as follows. The difference between the two largest eigenvalues encodes surface saliency, with a surface normal given by $\vec{e_1}$. The difference between the second and third eigenvalue encodes curve saliency, with $\vec{e_1}$ and $\vec{e_2}$ being normal to the curve and $\vec{e_3}$ tangent to it. Finally, the smallest eigenvalue encodes junction saliency. If surface saliency is high, the point most likely belongs on a surface and $\vec{e_1}$ is its normal. Outliers that receive no or inconsistent support from their neighborhood can be identified by their low saliency and the lack of a dominant orientation. In the case of stereo, we assume that that all inliers lie on surfaces that reflect light towards the cameras, and therefore we do not consider curves and junctions.

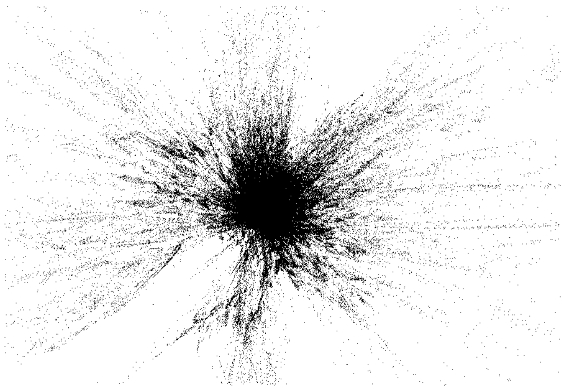


Figure 4: A view from above of all matching candidates (a little over one million) of the "meditation" set reconstructed in world coordinates. Notice the rays of matching candidates emanating from the cameras that are contained in the dome (close to the center of the point cloud)

The size of the voting neighborhood is a function of the scale σ of equation (1). It is the only free parameter of the framework and controls the amount of smoothness. The size of the voting neighborhood is set as the distance from the voter beyond which the magnitude of the vote is less than 1% of the magnitude of the voting tensor. In all cases, reasonable outputs can be produced over a large range of scales, with gradually varying degree of smoothness and robustness to noise.

Application to multiple view reconstruction The input to the tensor voting stage is a cloud of points (Fig. 4) and the desired output is a subset of those points that includes the inliers of the most salient surfaces. Since we need to process over one million points in the examples presented here, the fact that tensor voting operates in neighborhoods of fixed size is beneficial since the number of votes cast by each point is a function of the size of the neighborhood and thus complexity is not quadratic.

Taking into account visibility, we initialize the tensor at each point as an oriented surfel with a stick tensor pointing to the midpoint of the optical centers of the cameras that generated the matching candidate. This orientation estimate, that can be corrected after voting, assures that the point supports surfaces that could be visible from the appropriate cameras. All points cast tensor votes to their neighbors in 3-D, regardless of which images were used to generate each point, thus combining all information in one processing stage. Analysis of the accumulated tensors after voting produces corrected surface normal estimates and saliency values based on the support each point receives. A second pass of tensor voting using this information is performed before the final decisions are made.

Then, uniqueness with respect to the lines of sight is enforced, leaving one candidate for every image pixel. The criterion is maximum surface saliency. Therefore, for each pixel of all images, the match that receives the maximum support as an inlier of a salient surface, remains in the dataset. This step is not enough to guarantee the removal of all wrong matches, since we cannot assume that the matching candidates for some pixels, especially the ones in distorted or textureless areas, include the correct match. The remaining wrong matches can be rejected based on their low surface saliency, since, if no accidental alignment has occurred, they do not form coherent surfaces and, therefore, do not receive significant support from their neighbors as surface inliers. Simple thresholding with respect to average surface saliency is enough to produce datasets with mostly correct surface points, as shown in the next section.

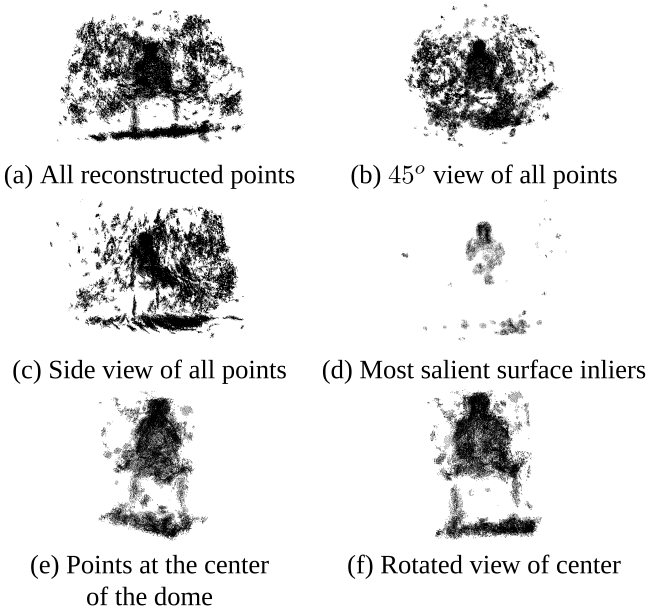

(a) All reconstructed points
(b) 45^o view of all points
(c) Side view of all points
(d) Most salient surface inliers
(e) Points at the center of the dome
(f) Rotated view of center

Figure 5: Results for the “meditation” set. Three views of all the reconstructed points with surface saliency at least twice its average value (a-c). A view of the points with surface saliency at least 12 times the average, which are mostly on the person and the floor (d); and two views of only the center of the entire dataset (e-f)

6 Experimental Results

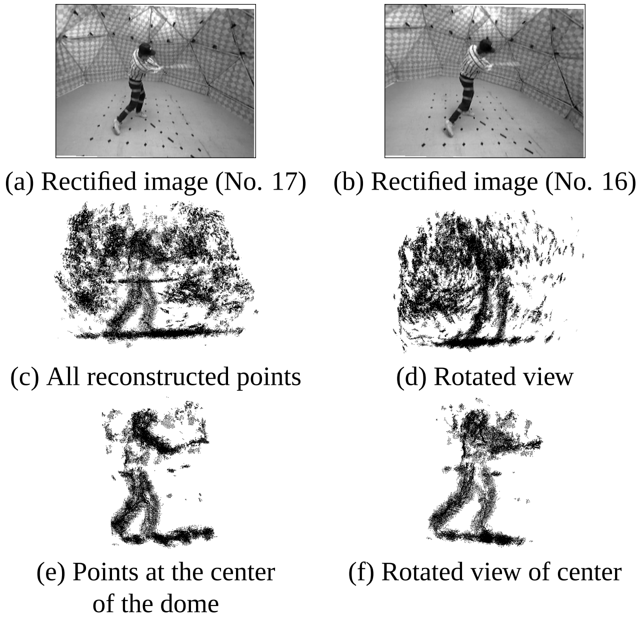
For the experiments presented in this section, we used two image sets captured at the CMU dome. Both the “meditation” and “baseball” sets were captured with identical settings and the results we present were produced with the same parameters. The input consists of the same 10 image pairs using a total of 17 images that were selected among the 51 total images based on viewpoint similarity to facilitate automatic matching. The images are 320×245 with 8 bits of grayscale values. Besides the distortion issues that were mentioned above, there is also motion blur in the “baseball” set (Fig. 6(a-b)).

Matching candidates are generated using a 7×7 correlation window as in Section 4. Nine pairs are processed with the same disparity range (-40 to 40 for “meditation” and -20 to 55 for “baseball”) and the last pair, which is elevated, is processed using 20 to 75 for the disparity range. The only negative effect of a large disparity range is an increase in processing time for the generation of matching candidates.

The matching candidates are reconstructed in 3-D by triangulating the rays that go through the two pixels and the optical centers of the cameras. Given the poor image resolution, quantization noise is significant and the localization of the reconstructed points is not perfect. They do, however, form surfaces that are close to the true scene surfaces. We are able to infer these surfaces from the point cloud after tensor voting since the points they comprise receive more support than the outliers. Unfortunately, we do not have ground truth data to compare our results to. We use $4mm$ as the unit of distance in the world coordinate system and perform tensor voting with $\sigma^2 = 500$, which corresponds to a voting radius of 48 units of distance.

The execution times of the un-optimized implementation of our code on a Pentium IV at 2.8GHz for the “meditation” set are the following:

- The generation of matching candidates using a 7×7 normalized cross-correlation window and searching 80 possible disparity levels takes 16 seconds.
- The reconstruction of all points in 3-D takes 1 minute and 21 seconds.
- Tensor voting and analysis of the results for 1,126,554 points with $\sigma^2 = 500$ takes 44 minutes and 30 seconds. (In the case of “baseball” tensor voting for 1,117,920



(a) Rectified image (No. 17) (b) Rectified image (No. 16)

(c) All reconstructed points (d) Rotated view

(e) Points at the center of the dome (f) Rotated view of center

Figure 6: Results for the “baseball” set. One of the ten pairs used (a-b). Two views of all the reconstructed points with surface saliency at least twice the average value (c-d); and two views of only the center of the dataset (e-f)

points takes 40 minutes and 2 seconds.)

Due to the nature of the scenes, visualization of the results is hard. Besides the entire output which includes the person, the visible parts of the dome and the floor, we have also included in Figs. 5 and 6 views of the center of the data only, where the person is, as well as of the topmost salient surfaces, which again tend to be those of the person where maximum point concentration occurs since they are visible from a larger number of cameras.

7 Concluding Remarks

We have presented an approach for multiple view reconstruction that can deal with scene and camera configurations that cannot be handled by current state of the art algorithms. The typical “blue sky” and “free space” assumptions are not required, with the only constraint on camera placement being that viewpoints must be similar enough to allow automatic pixel matching. The tensor voting computational framework allows the simultaneous processing of all matching candidates, utilizing all the available images to a far greater extend than approaches based on the fusion of binocular results. The computational cost is reasonable since all processing is local and distant points do not affect each other. Methods with quadratic complexity with respect to the number of pixels would incur severe computational requirements in examples with close to one million pixels, such as the ones presented here. We have strived to maintain a high level of generality. All images are treated equally and there are no privileged views for which depth maps are computed. The representation is fully three-dimensional and there are no view-dependent elements.

Satisfactory results can be produced even with low resolution and low quality images. This is due to the fact that features are not required to appear in more than two views. Given the poor resolution of the available images (320×245), matching a feature in multiple views is hard. In fact, very few matches where validated in additional images, besides the two used for the matching. However, since features are validated by the support the receive as surface inliers, the problem is transformed from validation at the point level to validation at the surface level. Therefore, as long as a matching candidate can be reconstructed in 3-D sufficiently close to the true surface, exact point correspondence in multiple images is not required. Thus, we are able to achieve reconstructions at a resolution higher than what would have been possible with methods such as [16] that require point correspondences in multiple frames or methods based on space carving [6] that require exact pixel-voxel correspondences.

Processing all data at the same time maximizes performance at occluded and textureless regions where binocular stereo typically encounters difficulties. The “fattening” of occluding surfaces, which is a problem in the binocular case, is diminished in the multiple view case because parts of both the occluding and occluded surfaces that are invisible in a particular image can be seen in different views and thus produce correct matches that outweigh the wrong ones. Errors in textureless regions that are caused by the ambiguity of matching are less severe when matching candidates from more than one pair are combined. Then, the correct surface is more likely to be formed, even in the midst of a large number of outliers. The same is not guaranteed to happen in the binocular case with the same error rate in the matching stage. Figure 2(d) shows a disparity map for one pair of the “meditation” set that was processed the same way as the entire dataset. We do not by any means claim that this is the best result that can be achieved for this pair, but it is worth noticing the number of errors that occur in occluded and textureless areas which are eliminated when all matching candidates are processed simultaneously. Fusion of binocular reconstructions cannot be more effective than simultaneous processing of all available information.

The proposed approach fails when the assumption that correct matches form more salient surfaces than incorrect ones does not hold. This is the case when systematic errors in the initial matching stage occur for some reason and the algorithm “hallucinates” non-existent surfaces. This is a weakness of all perceptual organization approaches, but does not occur often. The noise that appears in the results presented in this paper could have been eliminated by another pass of tensor voting or during the surface extraction process, but this is beyond the scope of this paper.

The contributions of this paper over the preliminary version of this work published in [4] improve the quality of the reconstruction and expand the range of configurations where our algorithm is applicable. Initial matching is now dense and no pixels are considered "unmatchable". It is also performed with subpixel accuracy leading to more accurate reconstruction of the matching candidates. The input set of images is not limited to equally spaced turntable sequences and there is no need for manual background segmentation. As discussed above, matches do not need to be validated in more than two views. In the examples presented here, very few matches are exactly confirmed by more than one pair of images due to low image resolution.

Acknowledgments

The authors would like to thank all the people associated with the Virtualized Reality project of the Robotics Institute of Carnegie Mellon University for making the data available to the community.

This research has been supported by the National Science Foundation grant IIS 03 29247.

References

[1] D. Scharstein and R. Szeliski, "A taxonomy and evaluation of dense two-frame stereo correspondence algorithms," *IJCV*, vol. 47, no. 1-3, pp. 7–42, April 2002.

[2] A. Laurentini, "How far 3d shapes can be understood from 2d silhouettes," *PAMI*, vol. 17, no. 2, pp. 188–195, February 1995.

[3] S.M. Seitz and C.R. Dyer, "Photorealistic scene reconstruction by voxel coloring," in *CVPR*, 1997, pp. 1067–1073.

[4] P. Mordohai and G. Medioni, "Perceptual grouping for multiple view stereo using tensor voting," in *ICPR02*, 2002, pp. III: 639–644.

[5] O.D. Faugeras and R. Keriven, "Variational principles, surface evolution, pdes, level set methods, and the stereo problem," *IEEE Trans Image Processing*, vol. 7, no. 3, pp. 336–344, March 1998.

[6] K.N. Kutulakos and S.M. Seitz, "A theory of shape by space carving," in *ICCV*, 1999, pp. 307–314.

[7] A.J. Yezzi and S. Soatto, "Stereoscopic segmentation," in *ICCV*, 2001, pp. I: 59–66.

[8] V. Kolmogorov and R. Zabih, "Multi-camera scene reconstruction via graph cuts," in *ECCV*, 2002, p. III: 82 ff.

[9] H. Jin, S. Soatto, and A.J. Yezzi, "Multi-view stereo beyond lambert," in *CVPR*, 2003, pp. I: 171–178.

[10] G. Medioni, M.S. Lee, and C.K. Tang, *A Computational Framework for Segmentation and Grouping*, Elsevier, 2000.

[11] P.V. Fua, "From multiple stereo views to multiple 3-d surfaces," *IJCV*, vol. 24, no. 1, pp. 19–35, August 1997.

[12] C.K. Tang and G. Medioni, "Inference of integrated surface, curve, and junction descriptions from sparse 3d data," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 20, no. 11, pp. 1206–1223, November 1998.

[13] M. Okutomi and T. Kanade, "A multiple-baseline stereo," *PAMI*, vol. 15, no. 4, pp. 353–363, April 1993.

[14] R. Szeliski and P. Golland, "Stereo matching with transparency and matting," *IJCV*, vol. 32, no. 1, pp. 45–61, August 1999.

[15] S.B. Kang, R. Szeliski, and J. Chai, "Handling occlusions in dense multi-view stereo," in *CVPR*, 2001, pp. I:103–110.

[16] M. Pollefeys, R. Koch, and L.J. Van Gool, "Self-calibration and metric reconstruction in spite of varying and unknown internal camera parameters," in *ICCV*, 1998, pp. 90–95.

[17] M. Lhuillier and L. Quan, "Quasi-dense reconstruction from image sequence," in *ECCV*, 2002, pp. II: 125–139.

[18] P.J. Narayanan, P.W. Rander, and T. Kanade, "Constructing virtual worlds using dense stereo," in *ICCV*, 1998, pp. 3–10.

[19] C.J. Taylor, "Surface reconstruction from feature based stereo," in *ICCV*, 2003, pp. 184–190.

[20] C.R. Dyer, "Volumetric scene reconstruction from multiple views," in *Foundations of Image Analysis*, 2001, pp. 469–489.

[21] G. Slabaugh, W. B. Culbertson, T. Malzbender, and R. Schafer, "A survey of volumetric scene reconstruction methods from photographs," in *Volume Graphics 2001, Proc. of Joint IEEE TCVG and Eurographics Workshop*. 2001, pp. 81–100, Springer Computer Science.

[22] J. Gluckman and S.K. Nayar, "Rectifying transformations that minimize resampling effects," in *CVPR*, 2001, pp. I:111–117.