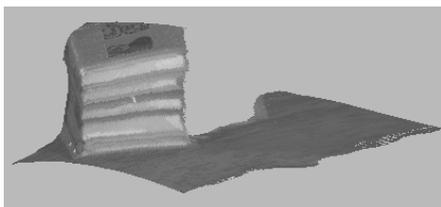
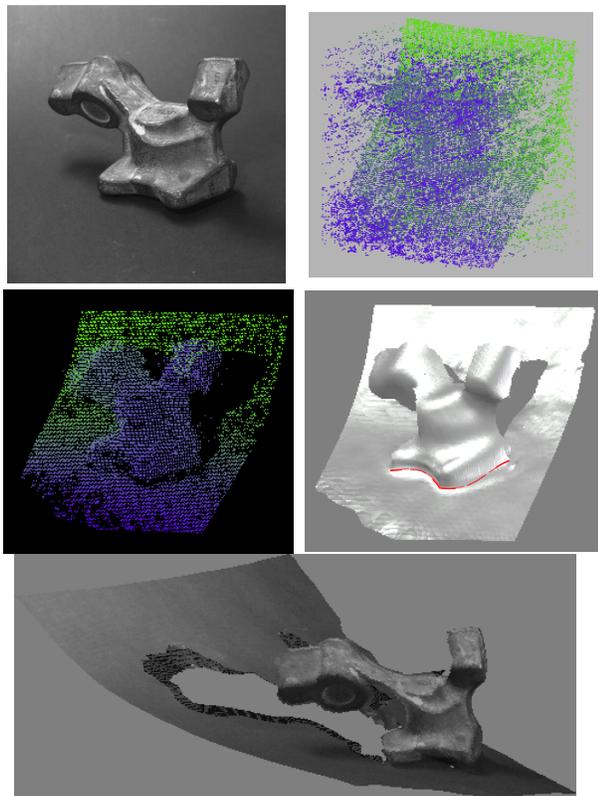


on the Renault part scene. Illustrated in the shaded view of the scene description is the inferred region for the half-occluded background (compare to the correspondence data). A rectified, texture mapped view of the scene is also presented. Notice that the left side of the Renault part, which is mostly occluded, is correctly inferred. In both texture mapped views, inferred regions with no texture information are given random texture.

We also applied our algorithm to a building scene captured by aerial image pair, depicted in figure 10. Using the knowledge that the target object is block-like building, we combine edge information with the inferred overlapping roof surfaces to derive vertical surfaces that are visible in both images. Inference of vertical surfaces is hard as they are often half occluded, or difficult to obtain by local correlation measurements. Also note that surfaces that are too small to provide correct correspondence are not detected.



(a) a rectified, texture mapped view of the book scene



(b) the Renault part scene

Figure 9 Experimental Results on real images

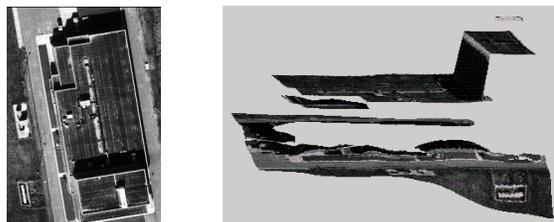


Figure 10 Result on a building scene

5 Conclusion

We have presented a surface from stereo method which addresses both the correspondence problem and the surface reconstruction problem simultaneously by directly extracting scene description from local measurements of point and line correspondences. Unlike most previous approaches, it explicitly addresses the occlusion process, leading to overlapping surface descriptions, that is multiple depth values for a given pixel. The method is not iterative, the only free parameter is scale, which was kept constant for all experiments shown here. We hope to demonstrate that this approach is also applicable to transparent surfaces with no change.

References

- [1] S.T. Barnard and M.A. Fischler, "Computational Stereo", *Computing Survey*, vol. 14, 1982, pp. 553-572.
- [2] P. Belhumeur and D. Mumford, "A Bayesian treatment of the stereo correspondence problem using half occluded regions", *Proc. CVPR*, 1992, pp. 506-512.
- [3] U.R. Dhond and J.K. Aggarwal, "Structure from Stereo-A Review", *IEEE SMC*, vol. 19, 1989, pp. 1489-1510.
- [4] P. Fua, "From Multiple Stereo Views to Multiple 3-D Surfaces", *IJCV*, vol. 24, no. 1, 1997, pp. 19-35.
- [5] G. Guy and G. Medioni, "Inference of Surfaces, Curves and Junctions from Sparse, Noisy 3-D data", *IEEE PAMI*, Nov 1997.
- [6] W. Hoff and N. Ahuja, "Surfaces from Stereo: Integrating Feature Matching, Disparity Estimation, and Contour Detection", *IEEE PAMI*, vol. 11, no. 2, Feb 1989, pp. 121-136.
- [7] H. Knutsson, "Representing local structure using tensors", *Proc. 6th Scandinavian Conf. Image Analysis*, Oulu, Finland, June 1989, pp. 224-251.
- [8] W.E. Lorensen and H.E. Cline, "Marching Cubes: A High Resolution 3-D Surface Reconstruction Algorithm", *Computer Graphics*, vol. 21, no. 4, 1987.
- [9] D. Marr and T. Poggio, "A Theory of Human Stereo Vision", *Proc. Roy. Soc. London*, vol. B204, 1979, pp. 301-328.
- [10] D. Marr, *Vision*, W.H. Freeman and Company, 1982.
- [11] L. Robert and R. Deriche, "Dense Depth map Reconstruction: A Minimization and Regularization Approach which Preserve Discontinuities", *Proc. 4th ECCV*, 1996.
- [12] R. Sara and R. Bajcsy, "On Occluding Contour Artifacts in Stereo Vision", *Proc. CVPR 97*, Puerto Rico, June 1997, pp. 852-857.
- [13] C.K. Tang and G. Medioni, "Integrated Surface, Curve and Junction Inference from Sparse 3-D Data Sets", *Proc. ICCV 98*, India, Jan. 1998.
- [14] C.F. Westin, *A Tensor Framework for Multidimensional Signal Processing*, PhD thesis, Linkoping University, Sweden, 1994. ISBN 91-7871-421-4.

junction curves and junction points can be extracted by a non-maximal suppression process [13] modified from the marching process [8]. Figure 6 depicts a slice of the inferred surface saliency for the book example. Note that although we use a specific surface model in our estimation, the estimation errors due to model misfit are incorporated as orientation uncertainties at all locations and are absorbed in the non-maximal suppression process.



Figure 6 A cut of the inferred surface saliency

While we only present our framework in the context of inferring surfaces, this tensor voting technique can also be applied to inferring curves, simply by changing the interpretation of the tensor components and the voting saliency tensor fields.

3.3 Region Inference

In this section, we address the problem of locating the boundaries of the surfaces obtained by the tensor voting process described above. Since monocular information of the boundaries is given in this case, we only need to identify the correct boundaries among the detected edges. Note that locating surface boundaries is essentially a 2-D problem embedded in 3-D. To simplify our description, we illustrate our region inference in 2-D.

In the spirit of our methodology, we again seek to compute for every line segments in the overlapping region a measure we call boundary saliency which relates the possibility of having the segment being the boundary of a region. Each point on a boundary has the property that most of its neighbors are on one side, which is illustrated in figure 7. We therefore can identify boundary points by computing the directional distribution of neighbors. This local discontinuity estimation is similar to the orientation estimation for salient surface inference, except that accurate orientation estimate is irrelevant to discontinuity estimation and thus only require the use of vectors. Therefore the voting function for boundary inference can be characterized as a radiant pattern with strength decays with distance from the center.

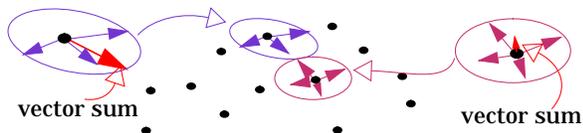


Figure 7 Boundary inference

Since a point on the boundary will only receive votes from one side while a point inside the region will receive votes from all directions, the size of the vector sum of the vector votes should indicate the “boundariness” of a point,

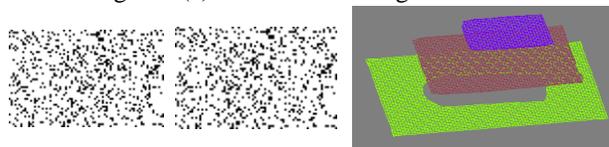
that is, the boundary saliency. On the other hand, the direction of the resulting vector relates the polarity information at the site. Boundary saliency for a line segment is defined to be the average boundary saliency of its points. To extract the salient boundaries, we use the boundary saliencies as the initial curve saliency and apply tensor voting to infer curves and junctions. Once the boundary curves are located, we can use the polarity information obtained in boundary inference to identify spurious regions. Embedding this region inference process in 3-D is straightforward as surface orientation have already been established. The results of applying this region inference process on the book example are shown in figure 3(b) and figure 2(e).

4 Experimental Results

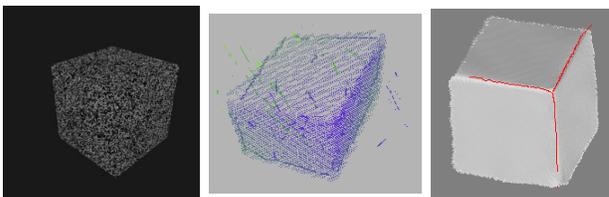
We implemented our algorithm and tested it on a wide range of images. For better visualization, we urge the reader to look at the associated video clips, movie 1-4, which show an animation, under different perspectives, of the resulting surface descriptions. Static results are also presented below.

Shown in figure 8 are two examples of shape inference from random dot stereogram. Figure 8(a) illustrates the description we obtained for the classical scene of 3 overlapping planes. Unlike the “wedding cake” description obtained by most approaches, our inferred description is that of 3 overlapping planes with holes where the continuity constraint is not strong enough to provide evidence. Applying the region inference process on the correspondence data, we accurately located the boundaries and the corners of the overlapping planes. Depicted in figure 8(b) is the cube scene in [6], where junction curves are correctly inferred.

Presented in figure 2 are the intermediate results during the inference of scene description for the book scene in the disparity scene. We estimated the camera calibration and obtained a rectified, texture mapped view of the scene, shown in figure 9(a). Also shown in figure 9 are the results



(a) a random dot stereogram of 3 planes



(b) a random dot stereogram of a cube

Figure 8 Experimental results on synthetic data

3.2 Tensor Voting

Our computational goal hence is to estimate the distribution of surface orientation and the associated saliency at every location in the domain space. According to the “matter is cohesive” principle, this task can be accomplished by analyzing the distribution of data in the neighborhood. Moreover, a discrete representation of the domain space is sufficient. Unlike traditional surface reconstruction methods which seek to obtain the estimation by computing a scalar value that measure the fitness of the data set against some specific surface model, we establish both the certainty of the surface orientation and the amount of support (the saliency) for the estimation by combining the surface orientation estimations obtained from every data item in a large neighborhood. Individual estimation is generated according to a simple, arbitrary surface model that takes into account the orientation information of the data item and the relative position of the target location with respect to the data item. Thanks to the redundancy and the linearity of the saliency tensor, this seemingly expensive process can be implemented by an efficient convolution-like operation which we call tensor voting. A similar voting technique that uses vector as data representation has been developed by Guy and Medioni [5] and shown to be effective in the inference of surface and discontinuities from sparse and noisy, but mostly synthetic, data. Once the orientation certainty and the saliency are established, we can label each location in the domain space by identifying the most salient estimations in the local neighborhood.

Since individual surface orientation estimations can be represented by saliency tensors and be combined by tensor addition to obtain a saliency tensor that describes the orientation certainty and the saliency of the estimation, it is the generation of individual estimation which we call vote that determines the efficiency of our method. By using simple surface model for orientation estimation, we are able to develop a convolution-like operation for vote generation.

Note that given the relative position of two locations and a surface orientation at one location, we can determine the surface orientation of the second location by fitting a simple surface. Figure 5(a) illustrates the situation by using sphere as the surface model. Since all input data contains orientation information, it is possible for each data item to generate votes for all locations in the domain space. For data with multiple surface orientation estimations, we only need to consider each of the orientations and then combine their effect into one vote. Since the reliability of the orientation estimation obtained by using such a simple model decreases with distance and curvature, each vote is associated with a saliency value according to the curvature of the fitting surface and the relative distance of the data and the target location. Thus, each input data can only vote

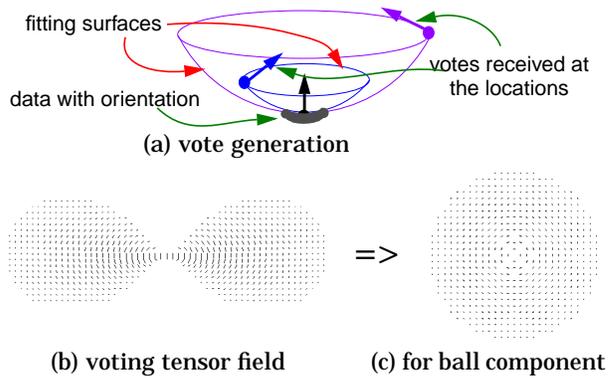


Figure 5 vote generation

in a finite neighborhood, whose size is a free parameter. In our case, we only consider surface models which provide consistent orientation information in a local neighborhood. In particular, any reasonable surface model should only provide one orientation estimation for a location and give similar votes for neighboring locations. Since the relative orientation of the vote with respect to the voter is only determined by the relative position of the target location with respect to the voting location, we can use a finite discrete saliency tensor field to capture the votes in voter’s coordinates, which we call the voting tensor field. Votes thus are generated by aligning the voting tensor field with the orientation estimation of the input data and centering the field at the data location. This voting process is similar to a convolution with a mask, except that the output is a tensor instead of a scalar.

In our implementation, we use the sphere as our surface model and the Gaussian function to model the saliency decay with respect to distance and curvature. Figure 5(b) illustrates a cut of the symmetrical voting tensor field.

Since our voting tensor field only take care of one orientation estimation of the input data, it would seem to imply that we need to perform multiple voting to generate the votes for a data item with multiple orientation estimations. Thanks to our tensorial framework, we can obtain the votes by considering each of the 3 basis components of the saliency tensor capturing the multiple orientation estimations and combining their effect linearly. Hence, we only need to handle 3 different cases of multiple orientation estimations, namely the stick, the plate, and the ball shapes. For each case, the votes can be precomputed in voter’s coordinates and stored in a finite discrete saliency tensor field. The voting process thus is modified as a convolution with three voting tensor fields, where each tensor contribution is weighted as described in equation (2). Figure 5(c) illustrates the projection of a cut of the symmetrical voting tensor field for the ball component.

A dense saliency tensor field is obtained after all input data have cast their votes, from which salient surfaces,

force the continuity constraint in a efficient and robust manner.

3.1 Saliency Tensor

Traditionally, the most important element of any surface reconstruction algorithm is the surface model. Numerous surface models have been proposed to characterize the smoothness aspect of 3-D surfaces. In most cases, surface discontinuities are only dealt with when model misfit occur. Moreover, the presence of outliers is not always addressed. We argue that these surface models are inadequate to capture the essence of the ‘‘matter is cohesive’’ principle proposed by Marr [10], as they do not explicitly represent surface discontinuities and outliers. This often results in iterations, and initialization and parameter-dependency.

We instead attempt to model surfaces, discontinuities and outliers simultaneously. Since it is hard to derive a global model for surfaces and discontinuities without any *a priori* knowledge, we prefer to use local representation. For surfaces, the basic information is the surface normal at the each location. A vector hence is adequate to locally model the surface and the saliency of the location as an inlier. Guy and Medioni [5], among others, have used this data representation in their surface reconstruction method. Modeling discontinuities is not as obvious though. First of all, there are two types of discontinuities, namely curve junctions and point junctions. We observe that discontinuities are located where multiple surfaces meet. At curve junctions, the surface orientation of the intersecting surfaces span a plane perpendicular to the tangent of the junction curve. At point junctions, the surface orientation of the intersecting surfaces span the 3-D space. An advantage of using the distribution of surface orientation to model discontinuities is its applicability to modeling surfaces as well. The desirable data representation hence should be able to encode the three different types of surface orientation distributions, which are in stick shape for surfaces, in plate shape for curve junctions, and in ball shape for point junctions. The inlier saliencies for each case should also be included. It turns out that a second order symmetric tensor possesses precisely these characteristics.

The tensor representation in fact can be derived directly from the covariance matrix \mathbf{T} that describes the distribution of surface orientations $\{\mathbf{u}_1, \mathbf{u}_2, \dots, \mathbf{u}_n\}$, where $\mathbf{u}_i = (x_i, y_i, z_i)$ is a vector representing a surface orientation and

$$\mathbf{T} = \begin{bmatrix} \sum x_i^2 & \sum x_i y_i & \sum x_i z_i \\ \sum x_i y_i & \sum y_i^2 & \sum y_i z_i \\ \sum x_i z_i & \sum y_i z_i & \sum z_i^2 \end{bmatrix}$$

By decomposing \mathbf{T} into its eigenvalues $\lambda_1, \lambda_2, \lambda_3$ and eigenvectors $\hat{e}_1, \hat{e}_2, \hat{e}_3$, we can rewrite \mathbf{T} as:

$$\mathbf{T} = \begin{bmatrix} \hat{e}_1 & \hat{e}_2 & \hat{e}_3 \end{bmatrix} \begin{bmatrix} \lambda_1 & 0 & 0 \\ 0 & \lambda_2 & 0 \\ 0 & 0 & \lambda_3 \end{bmatrix} \begin{bmatrix} \hat{e}_1^T \\ \hat{e}_2^T \\ \hat{e}_3^T \end{bmatrix} \quad (1)$$

Thus, $\mathbf{T} = \lambda_1 \hat{e}_1 \hat{e}_1^T + \lambda_2 \hat{e}_2 \hat{e}_2^T + \lambda_3 \hat{e}_3 \hat{e}_3^T$ where $\lambda_1 \geq \lambda_2 \geq \lambda_3$ and $\hat{e}_1, \hat{e}_2, \hat{e}_3$ are the eigenvectors correspond to $\lambda_1, \lambda_2, \lambda_3$ respectively. \mathbf{T} is a linear combination of outer product tensors and, therefore a tensor. As outer product tensors are symmetrical 2nd order tensors, the spectrum theorem[14] allow us to express \mathbf{T} as a linear combination of 3 basis tensors as:

$$\mathbf{T} = (\lambda_1 - \lambda_2) \hat{e}_1 \hat{e}_1^T + (\lambda_2 - \lambda_3) (\hat{e}_1 \hat{e}_1^T + \hat{e}_2 \hat{e}_2^T) + \lambda_3 (\hat{e}_1 \hat{e}_1^T + \hat{e}_2 \hat{e}_2^T + \hat{e}_3 \hat{e}_3^T) \quad (2)$$

where $\hat{e}_1 \hat{e}_1^T$ describes a stick, $(\hat{e}_1 \hat{e}_1^T + \hat{e}_2 \hat{e}_2^T)$ describes a plate, and $(\hat{e}_1 \hat{e}_1^T + \hat{e}_2 \hat{e}_2^T + \hat{e}_3 \hat{e}_3^T)$ describes a ball, and $\lambda_1 - \lambda_2, \lambda_2 - \lambda_3$, and λ_3 describe the associated saliencies respectively. Figure 4 illustrates the geometric interpretation of such tensor, which we call the saliency tensor for surface inference. Notice that the addition of 2 saliency tensors simply combine the distribution of the orientations and the associated saliencies.

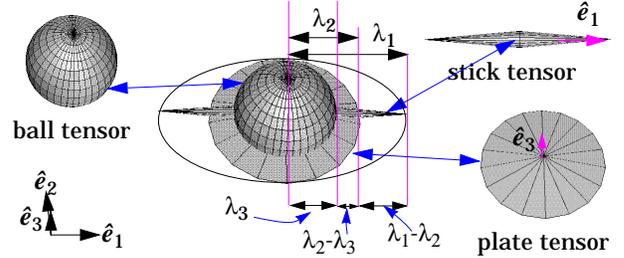


Figure 4 A general saliency tensor

Note that by changing the interpretation of the saliency tensor, we can also use this representation to encode the input data. While a point data signals the presence of activity, it contains no surface orientation information. Hence all surface orientations are equally probable at the location, which can be represented by a ball tensor as $(\hat{e}_1 \hat{e}_1^T + \hat{e}_2 \hat{e}_2^T + \hat{e}_3 \hat{e}_3^T)$, where $\hat{e}_1, \hat{e}_2, \hat{e}_3$ can be any set of orthogonal vectors. On the other hand, a line segment data with tangent \hat{e}_3 provide orientation information along the possible surface and hence can be described by a plate tensor as $(\hat{e}_1 \hat{e}_1^T + \hat{e}_2 \hat{e}_2^T)$. In the case where surface orientation is given as \hat{e}_1 , which does not occur in our surface from stereo algorithm, the stick tensor $\hat{e}_1 \hat{e}_1^T$ can be used. A general saliency tensor which capture multiple orientation estimations can also be used as input, such as in the third phase of our stereo algorithm. This uniformity of data representation allows us to combine various source of information efficiently and effectively.

data by removing non-salient correspondences in a non-iterative manner. Among the unexamined correspondences, the least salient one is identified. This disparity assignment, together with the neighborhood information associated with it, is removed unless it is the only one along the lines of sight in either image. We repeat this process until all the correspondence data are examined for uniqueness. Note that since local feature matching may fail to extract some of the correct correspondence, this uniqueness enforcement cannot remove all the wrong matches, as illustrated in figure 2(c) for the book scene.

2.3 Salient surface extraction

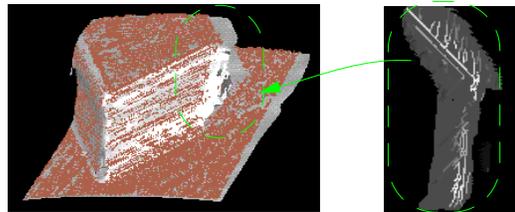
Once the more plausible correspondences are selected among the initial feature matches, we infer the underlying scene surfaces by spreading the computed saliency and continuity information throughout the entire disparity space: for each voxel in the disparity array, the density and distribution of the correspondence data in the neighborhood is collected and analyzed by applying tensor voting. Each location is then assigned a tensor which encodes its saliency and its orientation information as in the first phase. Locations with highest saliency in the local neighborhood are identified as being on a surface or on a surface discontinuity. The actual surfaces and discontinuity curves are extracted using a modified Marching process [13].

After this phase, the surfaces are segmented, but their boundaries are not accurately located. In fact, surfaces always overextend due to the absence of inhibitory process during voting. As an example, figure 2(d) (left) shows the inferred salient surface and junctions obtained from the disparity assignments depicted in figure 2(c). Also shown (right) are the correct correspondence so obtained.

2.4 Region trimming

In this last phase, we rectify the surface overexpansion problem by imposing the uniqueness constraint in the entire disparity space. To illustrate the symptom on the book scene, we superimpose the correspondence data on the inferred surface shown in figure 2(d). The problem area is highlighted in figure 3(a). Since overexpansion only happens at the occluding boundaries, and the associated occluded boundaries where scene surfaces stop abruptly, we can identify problem areas by finding surface patches which project onto the same pixel in one of the two images. Moreover, as each inferred overlapping surface region is partitioned by the corresponding occluding boundaries into two areas with correct matches all lie on one side, occluding boundaries and spurious regions can be inferred by analyzing the distribution of correspondence data along the inferred surfaces. However, as shown by Sara and Bajcsy in [12], intensity-based stereo matcher often shift the location of the occluding boundaries according to the contrast

of the corresponding edge in the images. It is therefore necessary to incorporate monocular information from the images when inferring occluding boundaries.



(a) inferred surface (shaded) and correspondence data (in red) (b) region boundary saliency

Figure 3 Region Trimming

We hence proceed to trim down the overexpanded surface by first locating overlapping surface regions in the inferred surfaces. Edge segments obtained in the preprocessing phase are then backprojected onto each of the overlapping surface regions. Using tensor voting, the distributions of the correspondence data along the surfaces are analyzed and are used to assign to each instance of edge segment a vector (also known as a first order tensor) which encodes both its saliency as a occluding boundary and its direction of occlusion. Figure 3(b) shows the inferred region boundary saliency for the book scene. Since edge detectors often fail to extract the entire occluding boundary, we need to fill in the gaps between the detected occluding boundaries. As this curve inference problem is essentially identical to the surface inference problem addressed in the previous phase, we apply a similar 2-D tensor voting process to extract the occluding boundaries. Spurious surface regions are then identified and removed from the inferred surfaces. Figure 2(e) presents the description we obtain for the stereo pair shown in figure 2(a). A texture mapped view of the inferred surfaces is also shown.

Note that only hangovers next to occluding boundaries are considered spurious. We argue that while the expanded regions of the occluded surface are not supported directly by any binocular evidence, their presence does not violate the uniqueness constraint (as they are occluded). In fact, according to the continuity constraint, their presence are strongly supported by the non-occluded regions of the corresponding surfaces. We hence retain these regions in our output for further analysis using higher level information.

3 Bounded Surface Inference

The computational core of our shape from stereo algorithm is tensor voting, which accomplishes the tasks of interpolation, discontinuity detection, and outlier identification simultaneously when inferring scene description from noisy, irregular data clusters. The strength of the technique stems from the use of tensor to represent the possible states of each location, and from non-linear voting to en-

thetic and real images in section 4 and conclude this paper with a discussion in section 5.

2 The Shape from Stereo Algorithm

In this section, we briefly describe our algorithm along with a running example. Figure 1 and figure 2 illustrate an outline of our algorithm.

After rectifying the images so that their corresponding epipolar lines lie along corresponding rasters, we initialize a 3-D disparity array in a traditional manner. We first extract from the two images the interest points that have measurable intensity variations in the neighborhood. We then compute the normalized cross-correlation between points in the corresponding rasters. For each interest point in the two images, we choose as potential correspondences the matches that have normalized cross-correlation values close to that of the match with the highest score (within certain percentage of the best score). Also, edges are extracted from the images. Potential edgel correspondences are generated by identifying edge segment pairs that share rasters across the images. Figure 2(b) depicts the point correspondences (left) and the line correspondences (right) so obtained for the book scene (from [6]) in figure 2(a). The

line correspondences are more ambiguous as they are obtained without considering the intensity information.

Given the initial multi-valued disparity assignments, we proceed to extract bounded surfaces by applying the continuity constraint and the uniqueness constraint to the correspondence data in the disparity space in four phases:

2.1 Correspondence saliency estimation

We assess the validity of each disparity assignment by establishing its role as either on a surface, or on a surface discontinuity, or an outlier in the disparity space through imposing the continuity constraint. Using tensor voting, which is detailed in section 3, neighborhood information is collected and analyzed. To avoid the “depth discontinuity problem”, we consider full 3-D neighborhood instead of the seemingly more convenient $2^{1/2}$ -D formulation. Based on the density and the distribution of the neighbors, each correspondence data is assigned a tensor which encodes both its saliency as an inlier and its orientation information as either a surface patch or a surface discontinuity.

2.2 Unique disparity assignment

Using the computed inlier saliency values, we then impose the uniqueness constraint on the correspondence

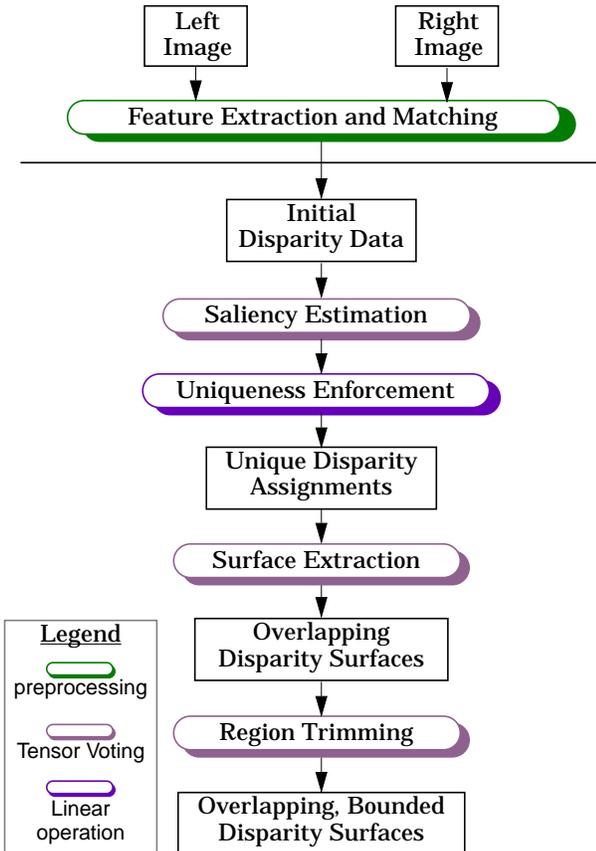


Figure 1 Overview of the algorithm

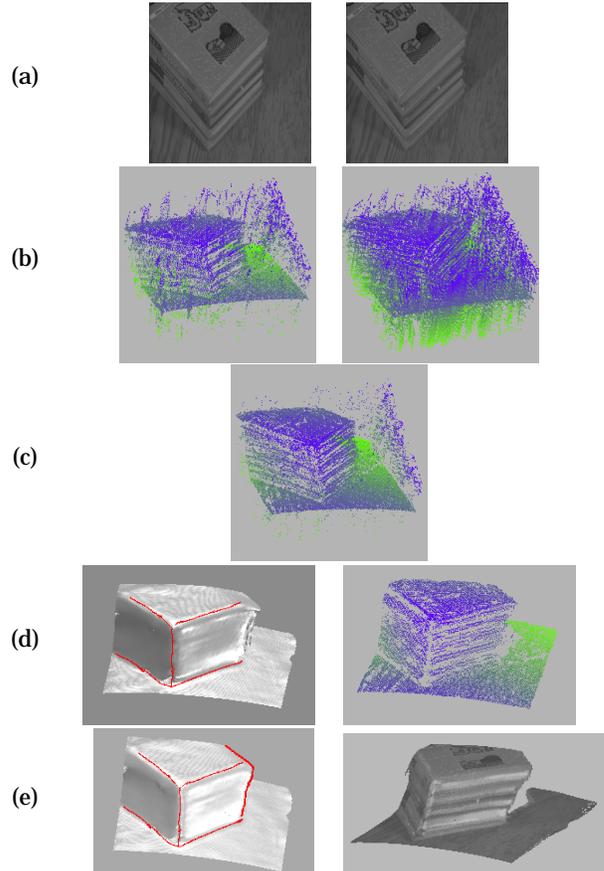


Figure 2 The book scene (see text)

Inferring Segmented Surface Description from Stereo Data

Abstract

We present an integrated approach to the derivation of scene description from binocular stereo images. Unlike popular stereo approaches, we address both the stereo correspondence problem and the surface reconstruction problem simultaneously by inferring the scene description directly from local measurements of both point and line correspondences. In order to handle the issues of noise, indistinct image features, surface discontinuities, and half occluded regions, we introduce a robust computational technique call tensor voting for the inference of scene description in terms of surfaces, junctions, and region boundaries. The methodology is grounded in two elements: tensor calculus for representation, and non-linear voting for data communication. By efficiently and effectively collecting and analyzing neighborhood information, we are able to handle the tasks of interpolation, discontinuity detection, and outlier identification simultaneously. The proposed method is non-iterative, robust to initialization and thresholding in the preprocessing stage, and the only critical free parameter is the size of the neighborhood. We illustrate the approach with results on a variety of images.

1 Introduction

The derivation of scene description from binocular stereo images involves two processes: establishing feature correspondences across images, and reconstructing scene surfaces from the depth measurements obtained from feature correspondences. The basic constraints used in the two processes are common, namely, the uniqueness and the continuity constraints (proposed by Marr [10]). The issues needed to be addressed in both cases are identical, namely, the presence of noise, indistinct image features, surface discontinuities, and half occlusions [2]. Despite the similarities, these two processes are traditionally implemented sequentially. Instead, Hoff and Ahuja [6] have argued that the steps of matching and surface reconstruction should be treated simultaneously. In this paper, we present an algorithm which approaches the shape from stereo problem from the same perspective. Given a calibrated stereo image pair, we derive a scene description in terms of surfaces, junctions, and region boundaries directly from local measurements of feature correspondence.

Numerous approaches following the usual match and reconstruct paradigm have been developed since Marr and Poggio's [9] work. We refer readers to the reviews by Barnard and Fischler [1] and Dhond and Aggarwal [3] for a comprehensive survey. The common goal of existing stereo algorithm is to assign a *single* disparity value to each point in the image, producing a $2^{1/2}$ -D sketch [10] of the scene. In this framework, stereo matching is usually cast as a constrained functional optimization problem [11]. Optimization techniques such as relaxation, dynamic programming, and stochastic methods are widely used in stereo algorithms. This formulation results in iterative, initializa-

tion and parameter dependent solutions, which often fail to handle surface discontinuities and half occlusion (the so-called depth discontinuity problem) properly. Also, even approaches which couple matching and surface inference [6][4] are formulated in an optimization framework.

The difficulty in modeling smoothness, discontinuities and outliers *simultaneously* in an optimization framework comes from the fact that each point in the 3-D world indeed can only assume one of the three roles: either on a surface, or on a surface discontinuity, or an outlier. Since their properties are very different, it is hard to capture the possible states by a single continuous function and to recover the role by making binary decision.

Inspired by Westin [14] and Knutsson's [7] tensorial framework to signal processing and Guy and Medioni's [5] work on non-linear voting, we propose to make use of the representational capability of tensors to encode the three possible states, and the robustness of non-linear voting to establish the role of each point in the scene.

As demonstrated in ample attempts to derive the optimal stereo matcher, local measurements such as cross-correlation indeed provide reasonable hypotheses for feature correspondences, among which correct matches cluster into bounded surfaces in disparity space. To extract these salient surfaces, we develop a technique call tensor voting to efficiently collect information in a large neighborhood containing both point and edge segment correspondences. By analyzing the composition of the neighborhood information, we are able to handle the tasks of interpolation, discontinuity detection, and outlier identification *simultaneously*. To deal with half occlusion, we sidestep the $2^{1/2}$ -D formulation, and handle scene inference directly in 3-D. Our multiple-phase method is non-iterative, robust to initialization and thresholding that happen only in the preprocessing stage, and the only critical free parameter is the size of the neighborhood.

The resulting description presents the following properties, as demonstrated on examples further on:

- the correspondence problem is resolved by comparing the saliency (likelihood) of local surface patches, derived from large area of support.
- Beside using the epipolar (intra-scanline) constraint, a large 2-D neighborhood (inter-scanline) is used to derive the local description.
- Areas visible by only one camera are automatically handled, leading to overlapping layers in description.

In the following, we first describe our shape from stereo algorithm in section 2. We then present in section 3 the details of applying the tensor voting technique to surface and region inference. We illustrate our results on both syn-