DEPTH INPAINTING WITH TENSOR VOTING USING LOCAL GEOMETRY

Mandar Kulkarni[1], A. N. Rajagopalan[1] and Gerhard Rigoll[2]

[1]*IPCV Lab, Electrical Engineering Department, Indian Institute of Technology Madras, Chennai, India*

[2]*Institute for Man-machine Communication, Technical University Munich, Munich, Germany*

Keywords: Tensor Voting, Range Maps, Range Inpainting.

Abstract: Range images captured from range scanning devices or reconstructed form optical cameras often suffer from missing regions due to occlusions, reflectivity, limited scanning area, sensor imperfections etc. In this paper, we propose a fast and simple algorithm for range map inpainting using Tensor Voting (TV) framework. From a single range image, we gather and analyze geometric information so as to estimate missing depth values. To deal with large missing regions, TV-based segmentation is initially employed as a cue for a region filling. Subsequently, we use 3D tensor voting for estimating different plane equations and pass depth estimates from all possible local planes that pass through a missing region. A final pass of tensor voting is performed to choose the best depth estimate for each point in the missing region. We demonstrate the effectiveness of our approach on synthetic as well as real data.

1 INTRODUCTION

Due to decreasing costs of range scanners, range images of 3D structures are becoming easily available. Range maps are widely used for applications such as 3D reconstruction, image based rendering (IBR) and matting. Also, depth estimation from optical images continues to be an exciting area of research. Many a time, range maps derived from these modalities often have missing regions. This could be due to various factors such as occlusion, low reflectivity, limited field of view, sensor imperfections etc. One needs to fill-in the missing regions for effective use of this 3D data.

Many approaches exist in the literature for estimating small or medium sized missing regions in range data. In (Stavrou et al., 2006), algorithms for 2D image inpainting are applied to 3D data assuming range maps as images. In (Sharf et al., 2004), an example-based approach is used for estimating missing values. The best match from an example dataset is found and the patch is fitted by aligning it with the surrounding surface. In (Xu et al., 2006), a single range image is used for estimating missing values. First, the normal directions at the missing values are estimated based on training data from image patches. Then a 3D surface is produced based on estimated normal directions. In (Bhavsar and Rajagopalan, 2010), an intensity image along with the range image is used for estimating missing values. The intensity image is registered with the range map. Using the segmented intensity image, missing values are estimated. Techniques based on stereo (Frueh et al., 2005)(Frueh et al., 2004) and structure-from motion (Abdelhafiz et al., 2005)(Brunton et al., 2007)(Dias et al., 2003) also exist and they use multiple intensity images to estimate 3D geometry.

In this work, we attempt inpainting given a single range observation derived directly from a range scanner or from optical images using the multi-view stereo principle. We achieve this objective within a tensor voting framework. Tensor Voting (TV) is a non-iterative framework originally developed by Medioni et al. (Guy and Medioni, 1997)(Medioni et al., 2000). This formalism which is based on tensor calculus for representing data and on a voting process for communication, identifies geometrical structures described by the layout of a sparse and potentially noisy N-D dataset. It also provides for each point a saliency measure about the possibility of its belonging to the type of structure being inferred. In our approach, we first detect edges in the range map using tensor voting. We extrapolate edge components inside the missing region. Segmentation based on edge interconnection provides a cue for region filling. We model depth variations based on local geometry and estimate individ-

Kulkarni M., N. Rajagopalan A. and Rigoll G..
DEPTH INPAINTING WITH TENSOR VOTING USING LOCAL GEOMETRY.
DOI: 10.5220/0003840100220030
In *Proceedings of the International Conference on Computer Vision Theory and Applications* (VISAPP-2012), pages 22-30
ISBN: 978-989-8565-03-7
Copyright © 2012 SCITEPRESS (Science and Technology Publications, Lda.)

ual plane equations using 3D tensor voting. Note that we perform local plane fitting as in (Bhavsar and Rajagopalan, 2010) but we do not use the intensity image for synthesizing missing values. This eliminates computations required for registering intensity image and range map. We pass depth estimates, obtained from different plane equations, inside the missing region. A final pass of 3D TV is performed to decide the best depth estimate. As lower depth regions occlude higher depth regions, we perform region filling starting with the lowest depth region and moving on to higher depth regions.

The paper is organized as follows. Section 2 describes our TV-based edge detection and Least Squares (LS) based edge interlinking process to enable robust segmentation. In section 3 we discuss region filling by local plane fitting using 3D TV. Experimental results are provided in Section 4 for purpose of validation.

2 TV FOR EDGE LINKING

Range maps usually have regions with varying depths. Some depth maps (such as piecewise planar scenes) exhibit sharp variations in depth from one plane to another while some display smooth variations. Without the missing region, we have continuous boundaries between these regions while a defect renders these boundaries discontinuous. It is important to reconnect these boundaries as they define the extent of each region in the range map.

Edge detection based on Canny, Sobel or Prewitt is sensitive to noise and tends to produce weak and spurious edges when applied to real depth maps. For robust edge detection, we resort to 2D tensor voting. The key idea is that true edge pixels form coherent edges in the range image. We first apply basic Sobel operator on the image. We set edge strength threshold to a high value so that only strong edge pixels get selected. These pixels are more likely to be true edge pixels. We refer to them as Reference edge pixels as they serve as reference for selecting weaker edge pixels subsequently. Next, we reduce the edge strength threshold by a small value so as to allow selection of weak edge pixels (some of them might even be noise). We refer to them as Candidate edge pixels. We use 2D tensor voting for selecting true edge pixels from Candidate edge pixels. Here, Reference and Candidate edge pixel locations act as tokens for tensor voting. We find the coherency of Candidate edge pixels with respect to Reference edge pixels. Here, Curve_saliency (obtained from the tensor voting) acts as a measure of coherency. For 2D tensor voting, a tensor T can be decomposed as

$$\mathrm{T} = (\lambda_1 - \lambda_2)e_1e_1^T + \lambda_2(e_1e_1^T + e_2e_2^T) \qquad (1)$$

Here λ_1 and λ_2 are eigenvalues of T corresponding to eigenvectors e_1 and e_2, respectively. The Curve_saliency is given by

$$\mathrm{Curve\_saliency} = \lambda_1 - \lambda_2 \qquad (2)$$

Thus, among all Candidate edge pixels, only those with high Curve_saliency value are retained. The threshold for saliency is chosen empirically. The retained Candidate edge pixels now become a part of Reference edge pixels to be used for the next stage. We iteratively reduce the edge strength threshold in small steps and select edge pixels using the above procedure. As only those edge pixels that form coherent connections are retained, our edge detection process is robust to noise.

We demonstrate the performance of our edge detection method on a real range map. Fig.1(a) is a sample real depth map. Fig.1(b) shows reference edge pixels for one of the stages and Fig.1(c) shows candidate edge pixels obtained with smaller threshold. It can be seen from Fig.1(d) that our approach selects only those candidate pixels which form coherent connections with reference edge pixels. The selected edge pixels (after the first iteration) are shown in red in Fig.1(d) along with reference edge pixels.

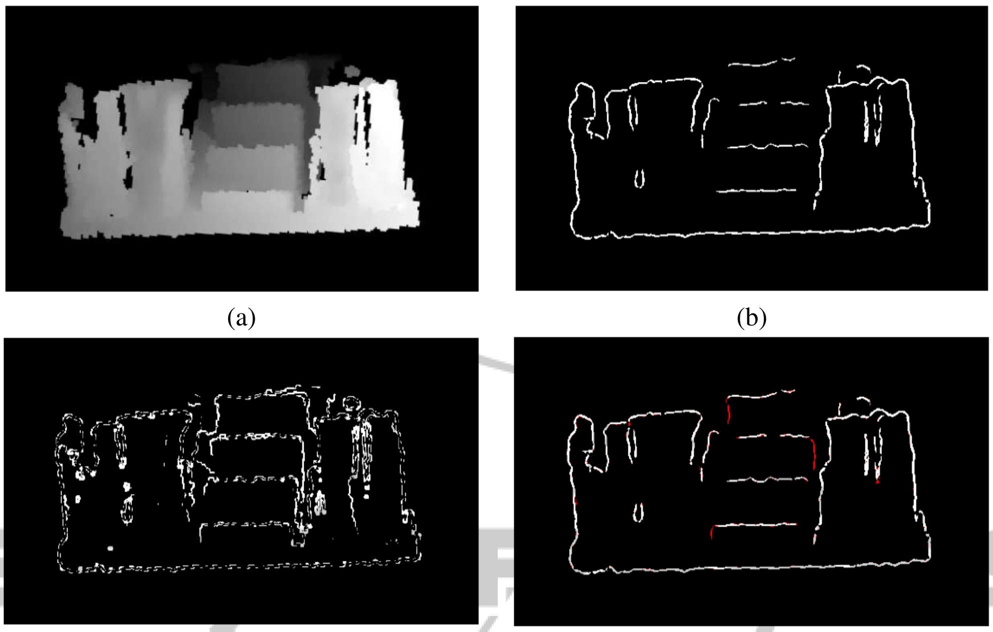
For interconnections, we only consider edges that are close to the missing region. The approach described in (Jia and Tang, 2003) uses 2D tensor voting for connecting edges. It assumes that there are no possible edge intersections within a missing region. In contrast, we use a least squares (LS) approach for modeling and connecting edges. The TV based approach, due to its limited voting range, imposes constraints on the size of the missing region. Also, it cannot handle possible edge intersections within a missing region. We attempt to connect two broken (discontinuous) edge components by a smooth curve. To enforce smoothness, we fit a second order polynomial to the edge data. A general equation of a second order polynomial is given by

$$Y = aX^2 + bX + c \qquad (3)$$

where (a,b,c) are the parameters of the polynomial fit. Here, X and Y are the spatial coordinates of a data point. We consider combinations of two separate (broken) edge components at a time. We define

$$\mathrm{Edge}(i) = [x_i, y_i] \qquad i = 1:N \qquad (4)$$

where $[x_i,\ y_i]$ is the co-ordinate of the i^{th} edge pixel and N is the total number of edge pixels from the two edge components. We use LS to fit a second order



SCIENCE AND TECHNOLOGY PUBLICATIONS

Figure 1: Robust edge detection. (a) A real depth map. (b) Reference edge pixels. (c) Candidate edge pixels. (d) Retained candidate edge pixels after the first iteration (shown in red).

polynomial to it and estimate the parameters (a,b,c). Least squares, if used directly for parameter estimation, can be erroneous as it is sensitive to outliers. We first perform 2D TV on the edge components and remove pixels with curve saliency less than a threshold.

We calculate the goodness of the fit with the data as

$$\text{Error}_j = \sum_{i=1}^{N} (y_i - ax_i^2 - bx_i - c)^2 \tag{5}$$

where Error_j is the polynomial fit error for the j^{th} edge combination.

For a given edge segment, we consider all potential combinations of edge components and find that edge combination which results in the least value for Error. We connect these edge segments using the parametric equation for that combination. The missing edge points [x,y] are generated as

$$y - ax^2 - bx - c \leq \text{ Edge\_threshold} \tag{6}$$

where (a,b,c) are parameters for the best match. The value of Edge_threshold should be close to zero. We set its value to 0.5 during experimentation.

For each individual edge segment, we carry out the same procedure and connect it to the best match derived from the remaining edge components. An edge segment which does not have a good match with any of the broken edge segments may result in an intersection within a missing region. We extrapolate such an edge segment across the missing region using its own parametric equation. This enables us to handle even edge intersections within the missing region. Fig.2(b) shows edge linking result on the broken edges of Fig.2(a).

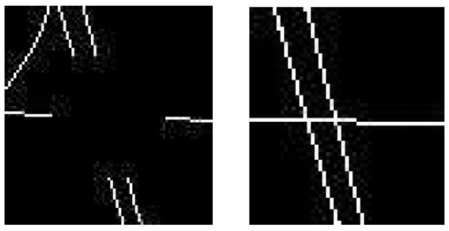


Figure 2: (a) Edge components near defect area. (b) Edge linking result using our LS approach.

3 REGION FILLING

In this section, we discuss two important steps namely, missing region segmentation and local plane fitting.

The edge interconnections, as described in Section 2, segment the missing regions and hence act as a cue for region filling. They define the correspondence i.e. part of available information that must be used for

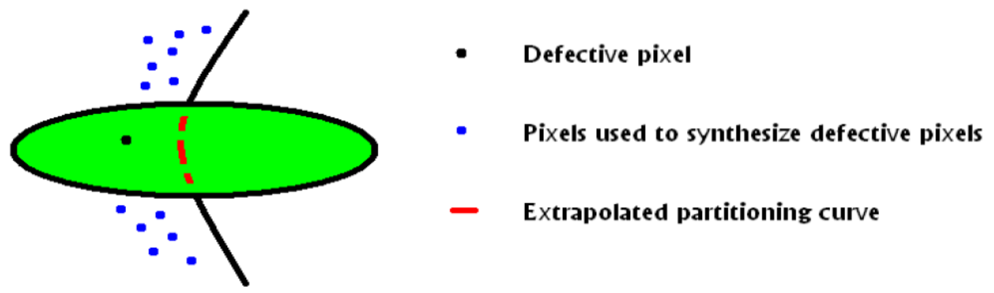

Defective pixel
Pixels used to synthesize defective pixels
Extrapolated partitioning curve

Figure 3: Missing region segmentation. Green color indicates the missing region.

synthesizing a given missing section. Each edge interconnection passing through a missing region divides it in two (left/right) sections. Each missing point coordinate $[X,Y]$ is locally assigned a region label L/R as follows.

$$\text{If } (Y - a_iX^2 - b_iX - c_i) \leq 0,\ \ M(X,Y) \rightarrow L \quad \text{else } \ M(X,Y) \rightarrow R \tag{7}$$

where (a_i,b_i,c_i) is the parametric equation of the i^{th} edge interconnection and M indicates missing region area. Based on the parametric equations of all the edge interconnections and the relative position of a missing point with respect to these interconnections, each point in the missing region is globally assigned a region label S_i (i=1:p) where p is the total number of segments. Using a bigger sub-image around the missing region, the available information segment with region label S_i is chosen to fill in the values for the missing segment with the same region label S_i. Fig.3 illustrates segmentation and labeling of a missing region based on the curve equations.

We next move onto assignment of depth values for each missing point.

3.1 Local Plane Fitting

We assume that points with similar depth values are part of a local linear plane in 3D. Assuming local planarity, a missing point is likely to belong to any one of the many planes that pass through the missing region. The task that remains is to identify the correct plane from a set of planes.

The general equation of a plane in 3D is given by

$$Ax + By + Cz + D = 0 \tag{8}$$

where (A,B,C) is the normal to the plane. We use 3D tensor voting to estimate the normal direction. For a missing segment with label S_i, we collect all the points with label S_i from known information. These points act as tokens for 3D TV. The tokens are formed as

$$\text{Data}(i) = [x_i, y_i, z_i] \qquad i = 1:N \tag{9}$$

where $[x_i,y_i]$ is the coordinate and z_i is the depth at known points. Here, N is the total number of tokens available to synthesize the missing segment information.

Each token has a 3x3 symmetric, positive semi-definite tensor matrix associated with it. At start, each tensor matrix is initialized to a 3x3 identity matrix. Following this, ball and stick voting are performed in the feature space. With the initialized shape and size, the tensors gradually deform due to the accumulation of votes cast from neighboring tokens within a certain range. The scale σ of the voting field controls the size of the voting neighborhood and the strength of the votes. A large value of σ corresponds to a higher voting range. The votes received contain both magnitude and orientation information. A generic tensor T with eigenvectors and eigenvalues can be decomposed into stick and ball components as

$$\text{T} = (\lambda_1 - \lambda_2)e_1e_1^T + (\lambda_2 - \lambda_3)(e_1e_1^T + e_2e_2^T) + \lambda_3(e_1e_1^T + e_2e_2^T + e_3e_3^T) \tag{10}$$

Here λ_1, λ_2 and λ_3 are eigenvalues of T corresponding to the eigenvectors e_1, e_2 and e_3, respectively.

We use the method proposed in (Lee and Medioni, 1997) to find the normal direction. We perform ball voting and stick voting on the variable Data. We obtain a 3x3 tensor matrix at each token. The eigenvector with the highest eigenvalue shows the normal direction at the token. We take normal directions at all token points and find the co-variance matrix. As each point on the same plane should have a unique normal direction, the eigenvector of the co-variance matrix corresponding to the highest eigenvalue is treated to be the normal direction to the plane. Thus, we can find D in Eq.(8) using one of the tokens from Data. We choose a token whose normal direction is most consistent with the estimated normal direction. Thus, we obtain the parameters (A,B,C,D) of the plane equation.

Note that all the tokens in the Data can be part of a single linear plane (e.g. for a planar scene) or there can exist multiple local planes (e.g for a curved surface) that pass through the missing region. To find equations for all possible planes, we first choose a set of tokens from Data which are clustered in space and similar in depth values as well (as these tokens are more likely to form a local plane). Using only those tokens, we estimate the equation of the plane as described earlier. We evaluate all other tokens against the estimated plane equation. A token $[x,y,z]$ is a part of the k^{th} local plane if

$$A_k.x + B_k.y + C_k.z + D_k \leq \text{Plane\_threshold} \qquad (11)$$

where (A_k,B_k,C_k,D_k) are the parameters of the k^{th} local plane. The value of Plane_threshold should be close to zero. We set its value to 0.3 during experimentation. We eliminate tokens from Data which satisfy Eq.11 as these are redundant. On the remaining tokens, we iteratively perform the above procedure until all the tokens are modeled by their local planes. Thus, we get a set of local plane equations which facilitate filling of missing information.

We compute K number of depth estimates(z_k), corresponding to K number of plane equations, at each missing point co-ordinate $[x,y]$ as

$$z_k = \frac{-A_k.x - B_k.y - D_k}{C_k} \qquad k = 1:K \qquad (12)$$

Thus, the number of depth estimates at a missing point is equal to the number of estimated local planes. The points from the available segment as well as missing point co-ordinates along with their depth estimates (from Eq.(12)) act as tokens for 3D TV. The surface saliency at each token is computed as

$$\text{Surface\_saliency} = \lambda_1 - \lambda_2 \qquad (13)$$

Out of the K depth estimates, the token $[x,y,z_k]$ with the highest surface saliency is selected and that z_k is considered to be the true depth value for the missing point $[x,y]$. It is intuitive that lower depth regions will occlude higher depth regions. Hence, we fill regions starting from lowest depth segments and proceed to higher values. For each available information segment, we find its average depth value. Using an available segment with lowest average depth value, the corresponding missing segment is synthesized first. Other segments are filled subsequently (in the order from lower to higher depth).

Note that σ is the only tuning factor in the TV framework and it controls the voting range. For large missing regions, we need a high value of σ. To automate the process, we adapt the value of σ to the size of the missing region.

4 RESULTS

We begin this section with demonstration of inpainting of a simple linear plane range image to depict the effectiveness of our plane equation estimation. Fig.4(a) shows a range image of a linear plane with missing values. We use points around the missing region for estimating the plane equation. In Fig.5, the red colored points indicate the tokens used for 3D tensor voting for plane normal estimation. The green arrow shows the estimated plane normal direction. Using the estimated normal vector, we estimate the complete plane equation from Eq.(8). As there exists only one plane, we have a single estimate for each missing value and we synthesize these values within the missing region using Eq.(12).

SCITEPRESS SCIENCE AND TECHNOLOGY PUBLICATIONS

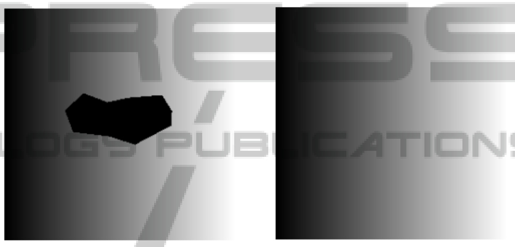

(a) (b)

Figure 4: Linear plane inpainting. (a) Defective range map, (b) inpainted range map.

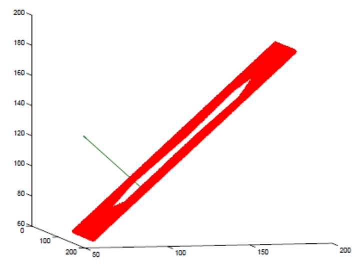


Figure 5: Plane normal direction: Red points indicate the tokens used for 3D TV. Estimated normal direction is indicated by green arrow.

Figs.6 and 7, we show the results of our approach on relatively large missing regions for scenes containing several planar segments. The range maps were taken from the Middlebury dataset (Scharstein and Szeliski., 2002). Once edge interlinking is performed, due to the planar nature of these scenes, there will be only one estimate for each segment as in the previous case. Thus, there are fewer number of tokens for final pass of 3D TV resulting in reduced execution time. Due to robust plane parameter estimation, missing values are indeed synthesized correctly (Fig.6(d) and Fig.7(b)(d)). In Fig.6(b,c,d), we also illustrate the order in which the depth estimates are filled-in by the proposed approach.

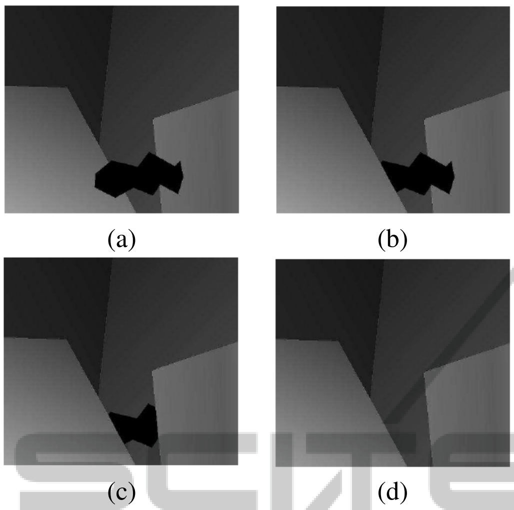


Figure 6: Planar range map inpainting. (a) Defective range maps. (b,c,d) order of filling-lower to higher depth.

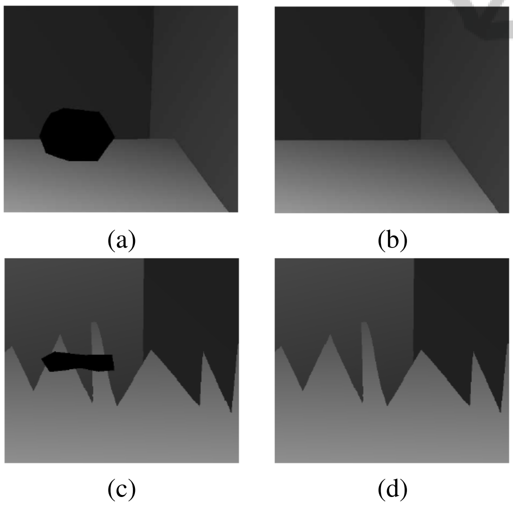


Figure 7: Planar range map inpainting. (a,c) Defective range maps. (b,d) inpainted range maps.

Fig.8 demonstrates the performance of our approach on depth maps from the Middlebury dataset (Scharstein and Pal., 2007)(Scharstein and Szeliski., 2002). One needs more number of local planes to model such a smooth surface. Using 3D tensor voting, we evaluate depth estimates obtained from local plane equations. We choose the estimate with the highest surface saliency value. As there are more number of tokens than in the planar case, the execution time goes up. It can be seen that our approach is quite effective in synthesizing even such smooth regions (Fig.8(b)(d)).

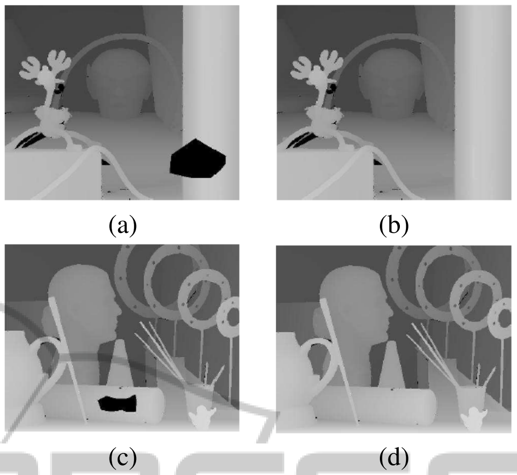


Figure 8: Smooth surface inpainting.(a,c) Defective range maps, (b,d) inpainted range maps.

Next, we take a scenario where there is a possible edge intersection within the missing region. Fig.9(a) shows the edge components existing near the missing region of Fig.9(c). It can be observed that the vertical edge segment in Fig.9(a) cannot be connected to other edge components with a smooth curve. There is a possibility of edge interlinking within the missing region. We extrapolate the non-matching segment in the missing region as shown in Fig.9(b). Then region filling is performed from lower to higher depths (Fig.9(d)(e)(f)) (in that order) as described earlier.

We also tested our method on range maps of human faces. Fig.10(a) shows a missing region in the smooth curved portion of a face. The proposed multiple local plane approach enables us to capture and fill this surface (Fig.10(b)). Fig.10(c) shows another case of missing region but now in the nose area. Our approach works yet again as is evident from Fig.10(d).

We also tested our approach on real range maps reconstructed using optical images. Interestingly, heritage sites provide examples of damaged structures due to weather, aging etc. Range maps of such structures show missing regions which we attempt to inpaint using the proposed method. We visited Mahabalipuram in Indai which is a UNESCO heritage site and captured images of monuments using an off-the-shelf digital camera. Multiple images from different view points were captured to reconstruct the shape.

Figs.11(a)(b) show optical images of a stone staircase captured from two different view points. This is an interesting case for inpainting as there are sharp edges and significant depth variations within the depth map. Using these images, we obtained the

SCITEPRESS
SCIENCE AND TECHNOLOGY PUBLICATIONS

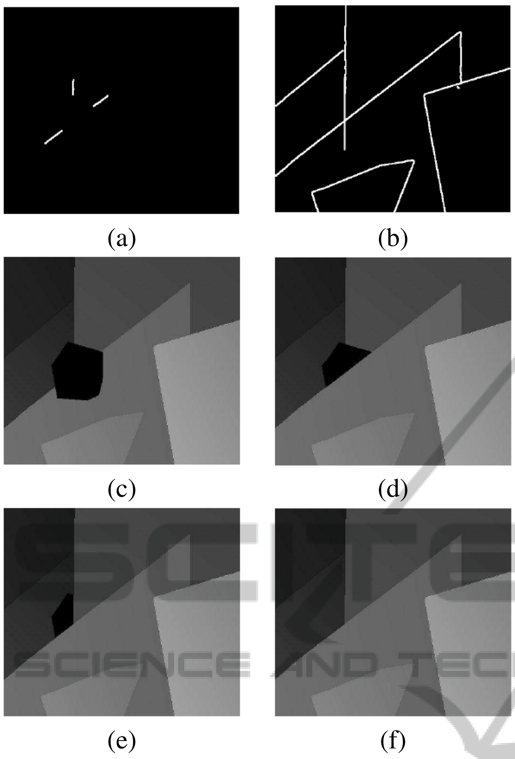


Figure 9: Possible edge interconnection in missing region: Edges are shown thick for demonstration. (a) Broken edge components around the missing region shown in (c). (b) Extrapolated edge within a missing region and (d,e,f) order of filling-lower to higher depth values.

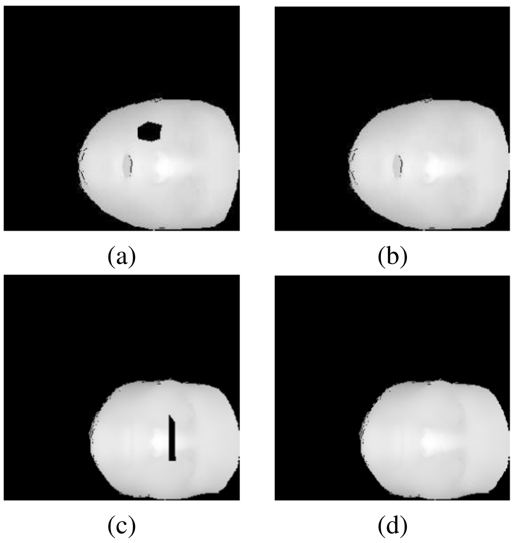


Figure 10: Results on face data. (a,c) Defective face range map (b,d) inpainted face range map.

disparity map. We manually marked a region for inpainting in this real depth map as shown in Fig.11(c). Since we perform TV based edge detection, we are able to locate true pixels accurately even for this noisy (real) depth map. We then interconnect broken edge components and synthesize missing values within the missing region. as shown in Fig.11(d).

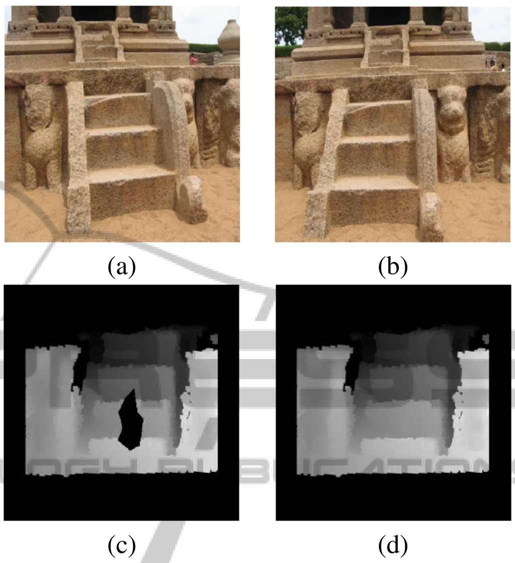


Figure 11: Results on real data. (a,b) Optical images of a staircase, (c) defective range map, and (d) inpainted range map.

Fig.12 depicts yet another example of a real depth map. A lion face sculpted on one of the pillars in the above mentioned heritage site was captured from different view points. Figs.12(a)(b) show the optical images of the structure taken from two different view points. Fig.12(c) shows the damaged depth map where the region to be inpainted has been marked. Note that there can be smooth variations of depth within the missing region along with possible edge interconnection. As TV based edge detection and region filling is robust to noise, our approach is able to synthesize the missing values accurately even in this situation as shown in Fig.12(d).

Fig.13(a) shows the image of a sculpted elephant. Fig.13(b) shows recovered depth map using multiple images of this object. Note that the depth map has missing regions at various places and these are indicated in red (Fig.13(c)). This is a commonly encountered scenario where the stereo algorithm fail to provide reliable depth values for all the scene points. Using our method, we synthesized the depth values in the missing regions one at a time. Fig.13(d) shows the final result of inpainting which is quite good.

We also performed a quantitative analysis of our results which is given in Table 1. To measure the error

SCITEPRESS SCIENCE AND TECHNOLOGY PUBLICATIONS

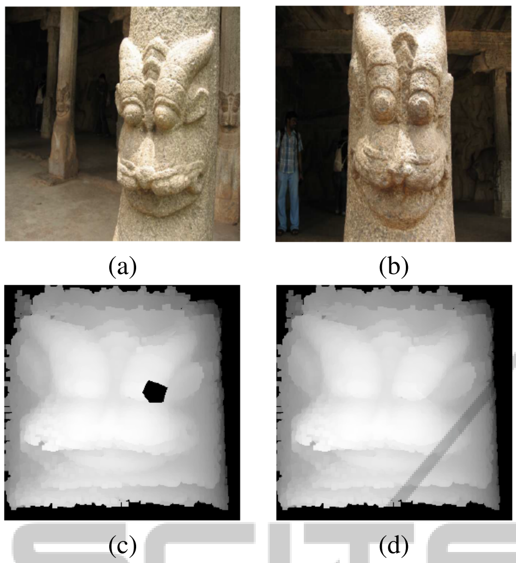


Figure 12: Results on real data. (a,b) Optical images of a lion face, (c) defective range map, and (d) inpainted range map.

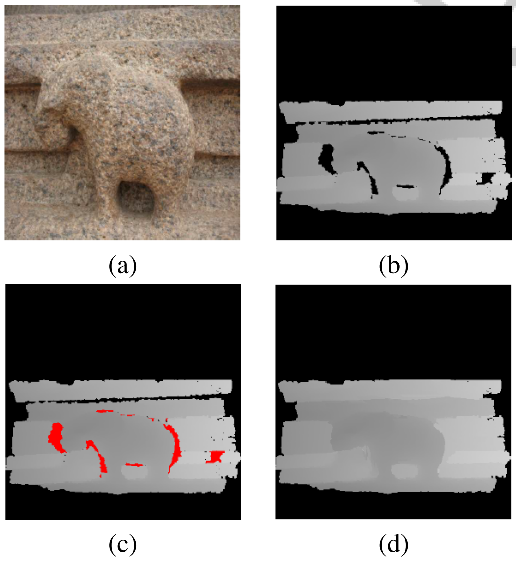


Figure 13: Results on real data. (a) Optical image of a sculpted elephant,(b) damaged range map,(c) missing regions to be inpainted shown in red, and (d) inpainted range map.

between the estimated depth values and the ground truth, we used the following error measure (Favaro et al., 2008).

$$\text{Error} = \sqrt{\text{Avg.}\left(\frac{\hat{z}}{z} - 1\right)^2} \qquad (14)$$

Here, z is the original and $\hat{z}$ is the estimated depth map. Averaging is performed over missing regions. From the table, it is clear that our results are close to the ground truth.

For a medium sized missing region in a planar scene, an unoptimized Matlab code takes around 5 seconds to execute while it takes around 15 seconds for a missing region of same size but within a smooth surface.

Table 1: Quantitative analysis.

Result	**Error**	**Result**	**Error**
Fig.6(d)	0.0173	Fig.7(b)	0.0124
Fig.7(d)	0.0189	Fig.8(b)	0.0049
Fig.8(d)	0.0063	Fig.9(f)	0.0224
Fig.10(b)	0.0075	Fig.10(d)	0.0077
Fig.11(d)	0.0127	Fig.12(d)	0.0062

SCITEPRESS
SCIENCE AND TECHNOLOGY PUBLICATIONS

5 CONCLUSIONS

We proposed a fast and reliable range inpainting approach for filling large regions given a single range/depth map. We used robust 2D tensor voting for edge detection. A least squares based approach was followed for modeling and interconnecting edge components around the missing region. 2D tensor voting was also used for refining and making edge interconnection robust. Edge interconnection enabled us to segment missing regions. This was followed by 3D TV which was employed to estimate plane equations using local geometry. Depth estimates obtained from different local planes were then passed inside the missing region and the best estimate was chosen based on surface saliency computed from a final pass of 3D TV. Results (both synthetic and real) reveal that our approach is quite effective.

ACKNOWLEDGEMENTS

The second author is grateful to the Humboldt Foundation, Germany for supporting this work.

REFERENCES

Abdelhafiz, A., Riedel, B., and Niemeier, W. (2005). Towards a 3d true colored space by the fusion of laser scanner point cloud and digital photos. *In Proc. of the ISPRS Working Group V/4 Workshop (3D-ARCH).*

Bhavsar, A. V. and Rajagopalan, A. N. (2010). Inpainting large missing regions in range images. *ICPR*, pages 3464–3467.

Brunton, A., Wuhrer, S., and Shu, C. (2007). Image-based model completion. *In Proc. of the 6th Int. Conf. on 3DIM*, pages 305–311.

Dias, P., Sequeira, V., Vaz, F., and Goncalves, J. (2003). Registration and fusion of intensity and range data for 3d modelling of real world scenes. *In Proc. 4th Int. Conf. on 3DIM*, pages 418–425.

Favaro, P., Soatto, S., Burger, M., and Osher, S. J. (2008). Shape from defocus via diffusion. *IEEE Trans. Pattern Anal. Mach. Intell*, 30(3):518–531.

Frueh, C., Jain, S., and Zakhor, A. (2005). Data processing algorithms for generating textured 3d building facade meshes from laser scans and camera images. *Int. J. Comp. Vis.*, 61(2):159–184.

Frueh, C., Sammon, R., and Zakhor, A. (2004). Automated texture mapping of 3d city models with oblique aerial imagery. *Proc. 2nd Int. Symp. on 3DPVT*, pages 396–403.

Guy, G. and Medioni, G. (1997). Inference of surfaces, 3d curves, and junctions from sparse, noisy, 3-d data. *IEEE Trans. on PAMI*, 19(11):1265–1277.

Jia, J. and Tang, C.-K. (2003). Image repairing: Robust image synthesis by adaptive nd tensor voting. *Conference on Computer Vision and Pattern Recognition(CVPR)*, pages 643–650.

Lee, S. H. and Medioni, G. (1997). Non-uniform skew estimation by tensor voting. *Proceedings of workshop on document Image Analysis*, pages 1–4.

Medioni, G., Lee, M. S., and Tang, C. K. (2000). A computational framework for segmentation and grouping. *Elsevier*.

Scharstein, D. and Pal., C. (2007). Learning conditional random fields for stereo. *In IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2007)*, pages 1–8.

Scharstein, D. and Szeliski., R. (2002). A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. *International Journal of Computer Vision*, 47(1):7–42.

Sharf, A., Alexa, M., and Cohen-Or, D. (2004). Context-based surface completion. *Proc. SIGGRAPH*, 23(3):878–887.

Stavrou, P., Mavridis, P., Papaioannou, G., Passalis, G., and Theoharis, T. (2006). 3d object repair using 2d algorithms. *Proc. International Conference on Computational Science*, pages 271–278.

Xu, S., Georghiades, A., Rushmeier, H., Dorsey, J., and McMillan, L. (2006). Image guided geometry inference. *Third International Symposium on 3D Data Processing, Visualization, and Transmission*, pages 310–317.

SCIENCE AND TECHNOLOGY PUBLICATIONS