www.cadanda.com
Computer-Aided Design AND APPLICATIONS™

Reconstruction of 3D Geometry of General Curved Surfaces from a Single Image

Alok Kumar Jha[1] and B. Gurumoorthy[2]

[1]IISc Bangalore, alok.mit@gmail.com
[2]IISc Bangalore, bgm@mecheng.iisc.ernet.in

ABSTRACT

Construction of 3D model from 2D images is important in reverse engineering and inspection applications. This work presents a new method to construct the 3D model of a free-form surface from a single image of the surface. The method combines the technique of recovering affine scene structure without requiring camera calibration, and the cubic corner method to extract 3D geometry. Recovering affine structure for images with free-form surfaces has been addressed for the first time. A novel modification of the cubic corner method is a construction scheme to determine a third point that forms an orthogonal triad. This approach is therefore able to handle cases where the three points being considered do not form orthogonal vectors. Results of implementation are promising. Future developments are discussed.

Keywords: 3D reconstruction, reverse engineering, freeform surface.

1. INTRODUCTION

Construction of 3D model from 2D images is important in reverse engineering and inspection applications. The problem of 3D reconstruction from a single 2D image is an ill posed problem, since information about the true depth of a scene point is lost in the 2D projection. Works on reconstruction from a single view have focused on either line drawings or photographs. Reconstruction of free-form entities from single line drawings has been studied extensively [18]. In this class of problems, it is assumed that the drawings are parallel projections of the object. Photographs are obtained by perspective projection. The popular approaches for handling sketches/line drawings such as labeling and optimization do not work well with images. Existing algorithms to solve such problems demand presence of reference entities (line, plane) and a variety of geometrical cues that can be easily recognized [5]. These entities or cues simplify the problem of camera calibration [19]. The reconstruction problem is straight-forward if the camera is calibrated. In single view approach camera calibration is the biggest hurdle.

There are reconstruction methods reported in the literature that are independent of the camera internal parameters such as focal length, aspect ratio principal point and skew [1,3,9]. Debvec et al. exploited the architectural scene characteristics [7]. Liu et al. addressed depth recovery by simulating human perception with user intervention and converting depth recovery to optimization problem with linear constraints [15]. Recovery of polyhedral model from a single projective image with complete geometric knowledge such as parallelism, perpendicularity, symmetry and reference distances all together is reported in the literature [11]. Colombo et al. have used the symmetry properties of the imaged surface of revolution [5]. Liebowitz et al. have suggested calibration based on scene constraints by exploiting orthogonality conditions, in order to reconstruct piecewise planar architectural scenes [13]. Vanishing points and plane at infinity (the set of all infinite points in projective 3-space defines a plane, known as the plane at infinity) have proven to be useful features for this task [1,3,9]. A survey by Company et al. [4] described the background and evolution of 3D reconstruction in last three decades mainly sketches and image based conceptual design. Criminisi et al. [6] proposed an approach for single view metrology, and showed that affine scene structure can be recovered from a single uncalibrated image. This approach requires three mutually orthogonal vanishing points to be available simultaneously in the image plane. Thereafter, in order to recover the metric measurements they require three reference distances. Once the affine structure is recovered, it is possible to use techniques used to inflate sketches. Cubic corner

Taylor & Francis
Taylor & Francis Group

technique has been used in the literature for obtaining the third dimension from polyhedral sketches drawn in parallel projection [12,17]. This technique is effective for regular polyhedra whose planes are all mutually perpendicular [16]. Liu and Lee [14] have extended this approach for the reconstruction of certain category of freeform shapes from a line drawing capturing the shape in parallel projection.

This paper proposes a combination of two approaches to address the problem of reconstructing free-form surface from given a single image. The first approach extracts a parallel projection of the surface from the perspective image available as input. This process, called rectification [13] involves removing the projective distortion using vanishing points in the image to establish affine geometry. Parallel projection is then obtained from the affine geometry using circular points.

The second part constructs a control polyhedron for the surface in the parallel projection followed by the application of cubic corner technique to inflate the control points to 3D. In the literature [14], cubic corner technique has been applied to points on the surface that may not always be the best suited for application of this technique. Control polyhedral is likely to offer points in the configuration (mutually orthogonal) where cubic corner works best.

The novelty in the first part lies in the relaxation of the requirement of three mutually orthogonal vanishing points required in the literature to the assumption that the surface boundary has parallel chords. This reduces the number of vanishing points required to two. In the second part, novelty is in the use of cubic corner with control points. A modification has been proposed to the implementation of the cubic corner technique where a different third point is determined that forms an orthogonal triad to apply the cubic corner method.

Presently, the input are restricted to untrimmed but multiply connected surfaces with parallel chords. The input image is first processed using standard edge detection schemes to identify the boundary of the surface. The boundary curve is then segmented to obtain the corners. This patch is then used to first obtain the parallel projection image followed by the control points. These are then inflated using cubic corner to obtain a 3D representation of the surface.

2. OUTLINE OF PROPOSED APPROACH

2.1. Edge Detection and Edge linking of the Image

Canny edge detection scheme [2] followed by edge linking is used to identify and segmenting the boundary of the surface of interest. Fig. 1 shows the output edge list.

2.2. Corner Detection of the Image

For corner detection Harris corner algorithm [10] is used. This algorithm identifies corners by determining change in intensity for every possible direction. Potential corners have large intensity change in all direction.

The corners detected in this may or may not belong to the pixels detected in edge list of section 2.1. These are now inserted in order in the edge list. The left frame in Fig. 2 shows the corners obtained from the Harris corner algorithm and the pixels in edge list. The right frame shows the corners after insertion into the edge list.

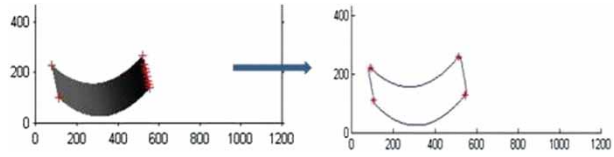

Fig. 2: Corners detected in input image.

2.3. Obtaining Parallel Projection from Perspective image

2.3.1. Mapping from Projective to Affine Space

Affine geometry is established by finding the plane at infinity in projective space for the input projective images. The usual method of finding the plane is by determining vanishing points in the image and then projecting them into space to obtain points at infinity. Vanishing points are the intersections of two or more imaged parallel lines. In the present case chords are assumed to be parallel. These are used to find the vanishing points and subsequently the vanishing line. In Fig. 3 the chords denoted by the projective line l_1 and l_2 intersect to yield vanishing point p_2 and chords

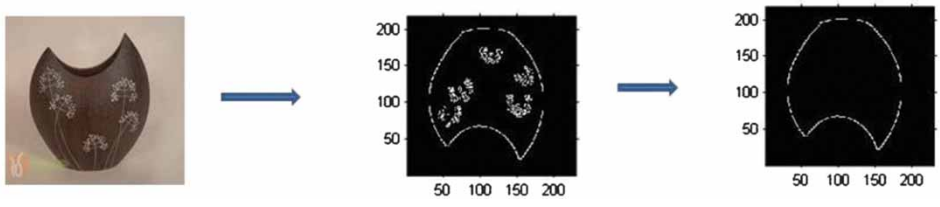

Fig. 1: Final segmented edge after smoothing and edge linking.

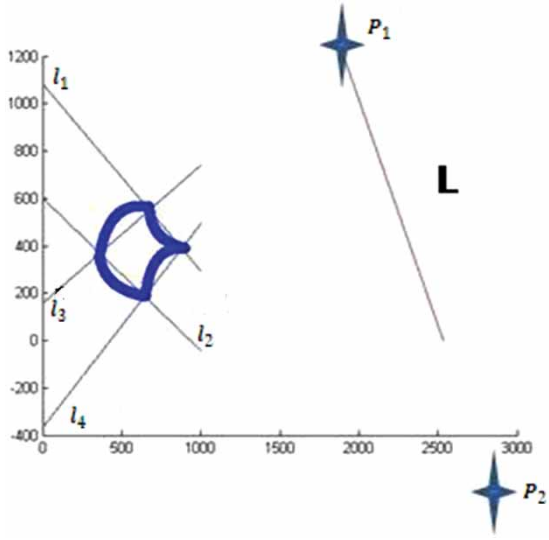

Fig. 3: Vanishing point and line.

denoted by projective line l_3 and l_4 intersect at vanishing point p_1. The line defined by these two vanishing points is the vanishing line (L).

The transformation corresponding to 'pure projection' is given by [13]

$$H = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ l_1 & l_2 & l_3 \end{bmatrix} \quad (1)$$

Here $L_\infty = [l_1, l_2, l_3]^T$ is the vanishing line of the plane. The input image is then 'affine rectified' using the above transformation to obtain the image in affine space.

2.3.2. *Mapping from Affine to Similarity Space*

Affine transformations preserve two points on the line at infinity called the absolute points or circular points. These points are also invariant to similarity transformations (rotations and translations). Therefore identifying circular points helps in recovering the similarity geometry. Therefore this mapping allows for a complete reconstruction up to an arbitrary scale. The scale can be fixed by knowing the dimensions of a reference object in the scene.

Affinely rectified image circular points are defined as

$$I = H^{-1}[1, i, 0]^T = [\alpha - i\beta, 1, -l_2 - \alpha l_1 + i l_1 \beta]^T \quad (2)$$

$$J = H^{-1}[1, -i, 0]^T = [\alpha + i\beta, 1, -l_2 - \alpha l_1 - i l_1 \beta]^T \quad (3)$$

Affine transformation (A) is defined as

$$A = \begin{bmatrix} \frac{1}{\beta} & \frac{-\alpha}{\beta} & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (4)$$

Here α and β define the image of the circular points. Liebowitz et al. have described the technique to calculate α and β by either known angle & length ratio or by Equal (unknown) Angles & Length Ratio [5]. Here using the known angle & chord length ratio image of circular points are found which in turn gives the α and β as shown in Fig. 4, therefore the matrix A to map corner and edge list from affine to similarity space.

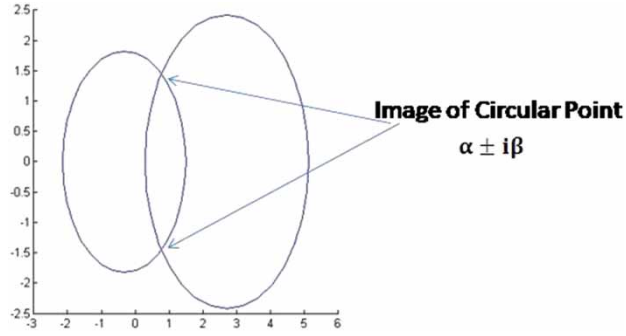

Fig. 4: Circular point.

The transformation from projective to similarity is shown in Fig. 5.

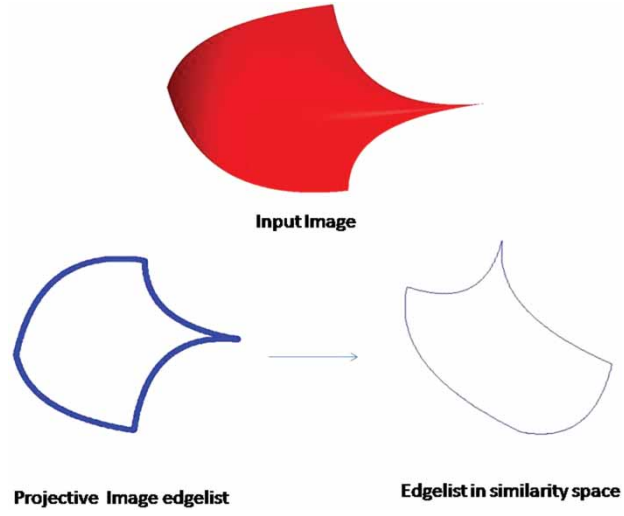

Fig. 5: Transformation from projective to similarity space.

2.4. 2D Control Point Grid Generation

2.4.1. *Fitting curve to the boundary edges*

From section 2.3 the edge list of the image in parallel projection is obtained. This boundary is segmented into boundary curves by the transformed corner points (Fig. 6). Points on each boundary segment are obtained from the transformed points corresponding to a pre-defined number of uniformly spaced pixels

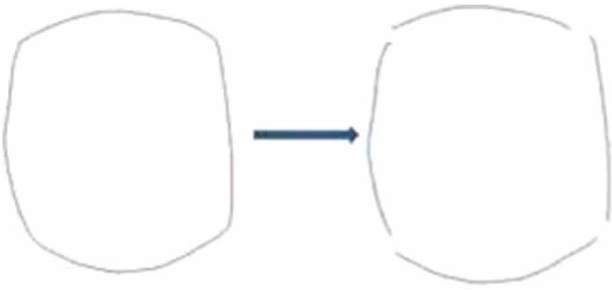

Fig. 6: Transformed patch and segregated edges.

in the image. Each boundary edge is then fitted with a B-Spline curve. The number of control points and the degree are maintained the same for each boundary curve. Presently trimmed patches are not addressed.

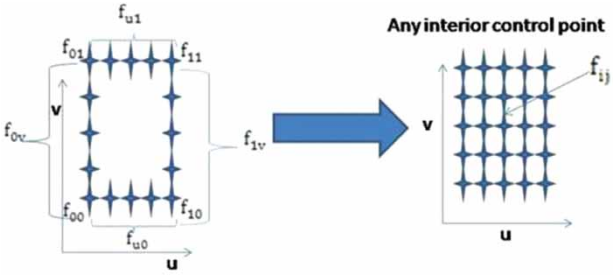

Fig. 7: 2D control point grid generation from boundary curves.

2.4.2. 2D Grid Generation

Let the boundary curves obtained be f_{u0}, f_{u1}, f_{0v}, f_{1v} and boundary points be f_{00}, f_{01}, f_{10} and f_{11} as shown in the left frame of Fig. 7. Here the domain of the parametric surface f_{uv} is the unit square $0 \le u \le 1$, $0 \le v \le 1$. A parametric surface f_{uv} with these as boundary curves can be computed from bilinearly blended Coons patch that interpolates the given boundary curves:

$$f_{uv} = [(1-u)f_{0v} + (u)f_{1v}] + [(1-v)f_{u0} + (v)f_{u1}] - [1-u\ u]\begin{bmatrix} f_{00} & f_{01} \\ f_{10} & f_{11} \end{bmatrix}\begin{bmatrix} 1-v \\ v \end{bmatrix} \quad (5)$$

In the present problem, the bounding polygon of the patch defined above can be treated as control polygons of the boundary curves. The interior control points of the surface can now be defined using the discrete Coons patch [8].

The interior control points f_{ij} (Fig. 7) are given by

$$f_{ij} = [(1-i/m)f_{0j} + (i/m)f_{mj}] + [(1-j/n)f_{i0} + (j/n)f_{in}] - [1-i/m\ i/m]\begin{bmatrix} f_{00} & f_{0m} \\ f_{n0} & f_{mn} \end{bmatrix}\begin{bmatrix} 1-j/n \\ j/n \end{bmatrix} \quad (6)$$

Here 'm' and 'n' are the number of control points along u and v directions respectively. In the present case, both are equal as the number of control points chosen to fit the curves were identical.

A grid of control points is now obtained. This will be used to construct 3D control points using the cubic corner method. Control points are inflated as opposed to points on the surface as they are more likely to satisfy the conditions required to apply cubic corner method.

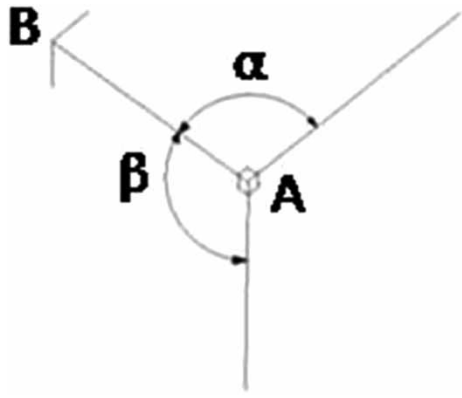

Fig. 8: Schematic of cubic corner.

2.5. 3D Grid Generation

2.5.1. Cubic Corner Method

For finding z-coordinates one of the well-known method is Perkins's Cubic Corners [16] which is the first formalized approach of perpendicularity hypotheses in interpreting line drawings as shown in Fig. 8 [15,16].

$$|Z_B - Z_A| = AB\sqrt{\frac{-\cos(\alpha)\cos(\beta)}{\cos(2\pi - (\alpha+\beta))}} \quad (7)$$

Here for every control point, angle α and angle β are found and then applying cubic corner theory ΔZ is found. All control points in the control polyhedron are traversed. The Z value for the first control point is initialized to be 0.

The method fails for junctions which do not meet the "Perkins criteria" (either a W-junction in which the two internal angles are acute but their sum is obtuse, or a Y-junction in which all three angles are obtuse), since the value under the square root in equation (7) becomes negative. In the literature, weighing functions are used to account for deviations from orthogonal edges.

2.5.2. New Approach to Apply Cubic Corner

A modification in the application of the cubic corner is proposed to overcome the limitations mentioned above.

For every three points say A_1, O_1, B_1 in 3D space a unique planar circle will exist. Projecting this unique 3D planar circle in 2D space will result in ellipse. Let the 2D projections corresponding to 3D points A_1, O_1, B_1 be A_1, O_1, B_1 respectively. Now the tangent to the circle at point O_1 in 3D space will be preserved as tangent at point O_1 of ellipse in 2D space.

Therefore a vector $\overrightarrow{O\_1\,Q\_1}$ that is tangent to the ellipse will be orthogonal to the radius vector through O_1. It follows that the vectors corresponding to $\overrightarrow{O\_1\,P\_1}$ and $\overrightarrow{O\_1\,Q\_1}$ in the 3D planar circle are also orthogonal. Now for translational and rotational sweep surface, vectors $\overrightarrow{O\_1\,P\_1}$, $\overrightarrow{O\_1\,Q\_1}$ and $\overrightarrow{O\_1\,R\_1}$ are mutually perpendicular. Hence cubic corner can be applied to O_1 without heuristics. Fig. 9 shows 2D image of the real 3D cubic corner in parallel projection. Fig. 10 illustrate the example in which the P_1 and Q_1 is computed to apply cubic corner at O_1.

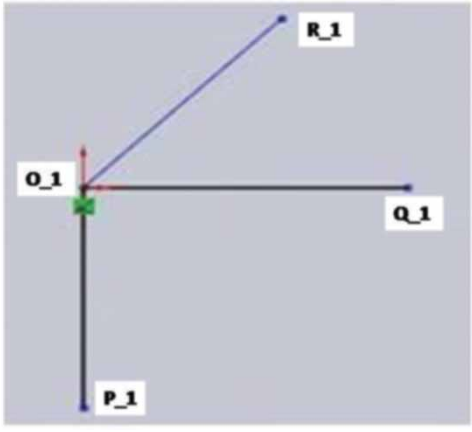

Fig. 9: Ideal situation for cubic corner.

Now applying cubic corner to the 2D point O_1 will give the Z value of P_1, Q_1 and R_1 with respect to O_1. Finding 3D plane equation from these points, one can find the Z value for A_1 and B_1, as all these points are lying in the same ellipse plane. An illustration with an example is described below.
A surface ABCD is shown in Fig. 11. The control polygon for this surface is A O_1 B C R_1 D. Tangent at O_1 is $\overrightarrow{O\_1\,Q\_1}$ of ellipse passing through AO_1B. The vector $\overrightarrow{O\_1\,P\_1}$ is passing through centre of ellipse. Applying cubic corner at point O_1 having neighbor points Q_1, P_1 and R_1, assuming Z = 0 at O_1, will give the Z value for the Q_1, P_1 and R_1. As A, B, O_1, Q_1 and P_1 are laying in a same plane, using the plane equation the Z value for A and B can be evaluated.

Assigning the third coordinate to 2D control points, we obtain the 3D control point and thereafter the B-spline surface.

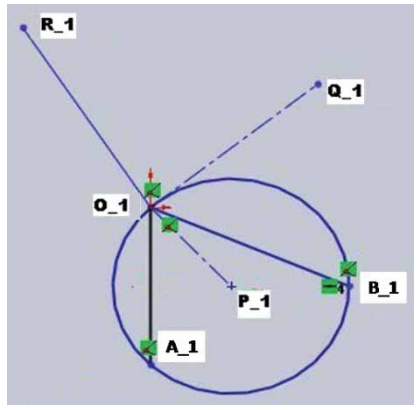

Fig. 10: Generation of P_1, Q_1 at O_1.

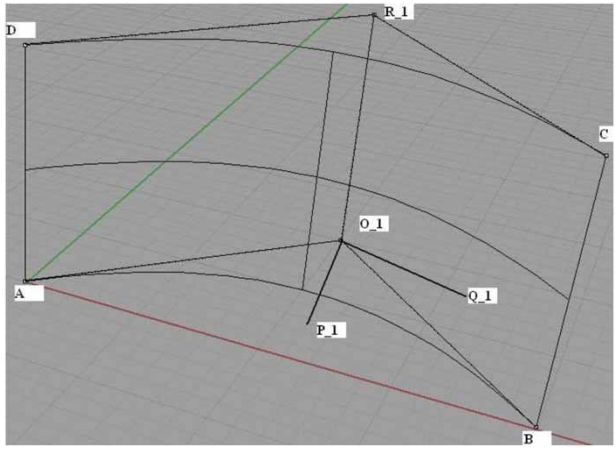

Fig. 11: Surface with its control polygon.

3. RESULTS

The algorithms described above have been implemented in a Windows platform (Intel(R) Core^{TM}2Duo CPU, 2.93GHz processor, and 2.93 GB of RAM) using MATLAB for edge detection and display of results. The first two images shown below (Fig. 12) are synthetic and the last three are from real images. As can be seen in the images below, some of the examples do not have parallel chords. The resulting surface looks correct visually. Comparision has not been possible for the real images due to lack of availability of the original 3D surface. In the following section, a synthetic image is used to check the performance of the procedure.

4. VALIDATION

The 3D model obtained is in similarity space and includes an arbitrary scale factor. Measurement of quantities for validation would need scale correction using known lengths in the image. In this case, a synthetic perspective image was constructed from a known 3D CAD model and used as input to the proposed approach. Fig. 13 shows the input synthetic

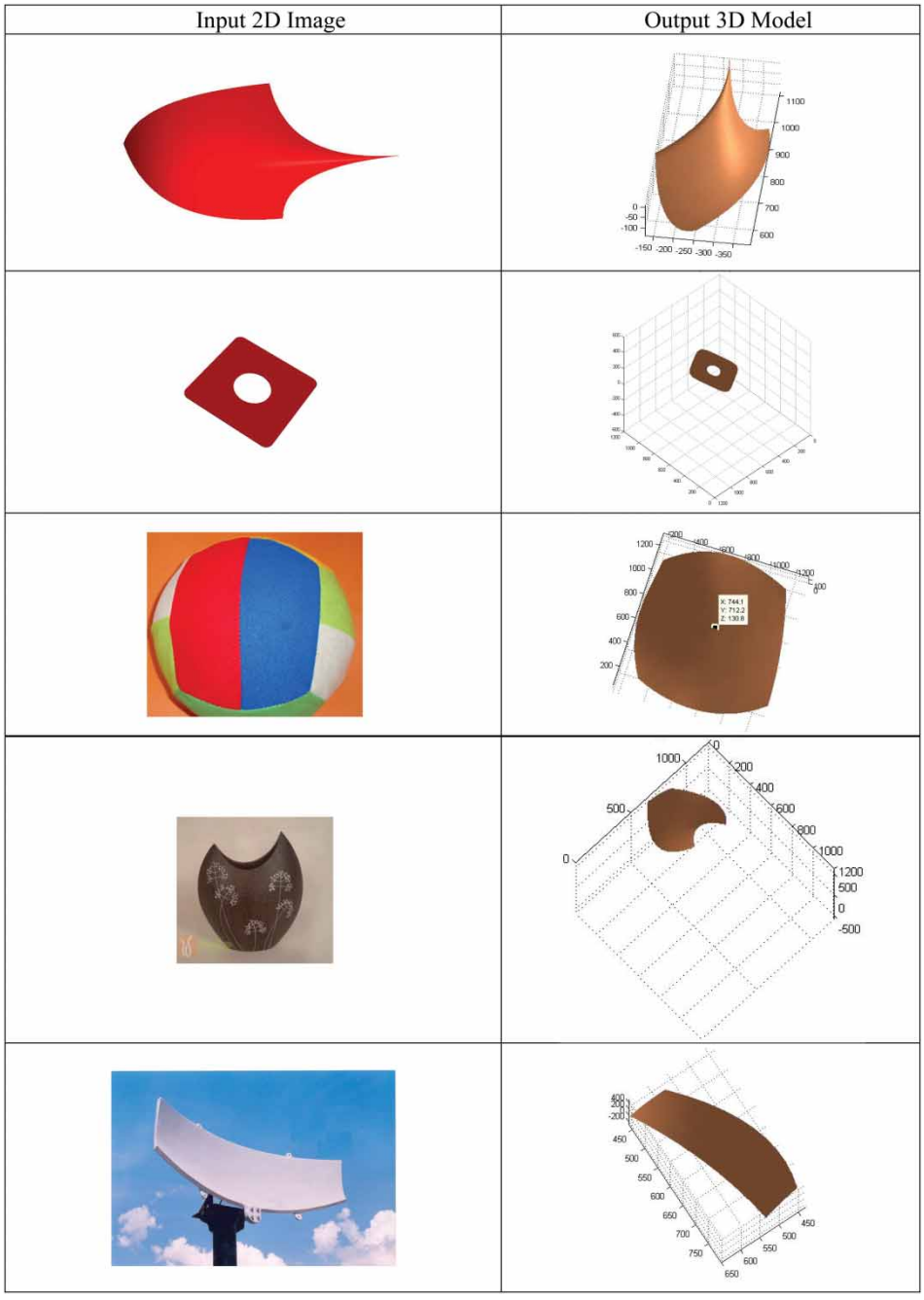

Fig. 12: Results.
First two input images are synthetic (made in solid-works and the last three images are captured from 12.1 mega-pixels Sony Cyber-shot camera).

image and Fig. 14 shows the surface obtained as output from the approach presented. One of the lengths is used as a reference dimension to obtain the scale factor. This scale factor then is used to obtain other dimensions in the surface that are then compared with the 3D model.

The height of a point on the surface is corrected using the scale ratio and compared with the height of the corresponding point on the CAD model along a parametric grid. The deviation or error is shown in the Tab. 1. Fig. 15 shows the error plot.

5. DISCUSSION & CONCLUSION

The proposed procedure has been implemented and tested on some surfaces. Chord lengths are taken as cues for conversion from perspective to similarity space and later in full metric three dimensional

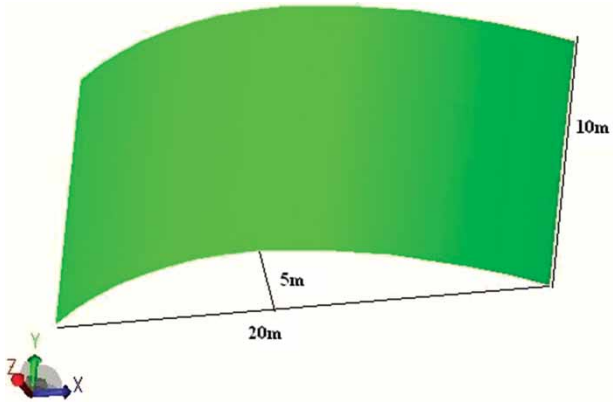

Fig. 13: Input image.

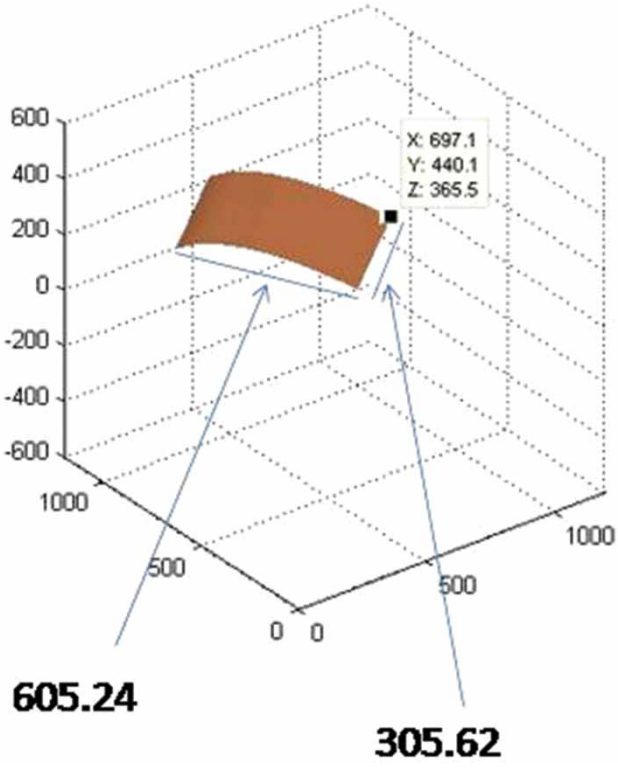

Fig. 14: 3D Output model.

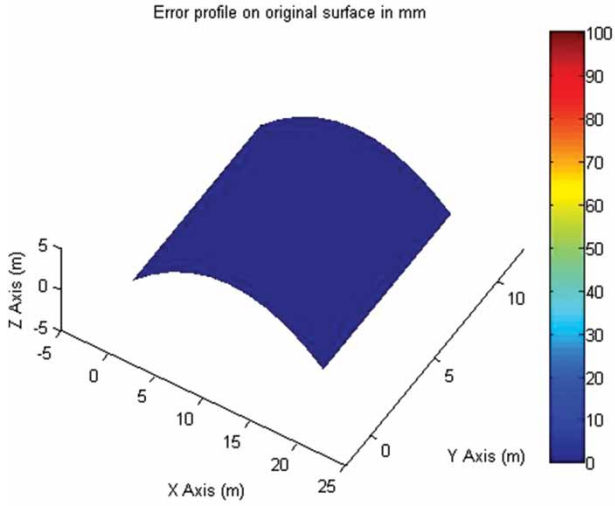

Fig. 15: Error profile on original surface in mm at respective parametric point.

reconstructions. The error in the third dimension (depending on the view) is within 5%. The error is worst at the ends because of the fact that the third point at the boundary edge is not available for the cubic corner method and is currently forced to a pre-defined value. The advantages of the proposed approach over the current art are that no feature extraction is required and the image needs to have only two vanishing points in orthogonal directions. This approach will be of use in situations where neither camera parameters nor features on the object are readily available. Only untrimmed surfaces have been addressed here. Ongoing work is exploring the use of intensity data in the interior (in addition to the boundary information presently used) to improve the accuracy of reconstruction and to remove the restriction of parallel chords.

REFERENCES

[1] Caprile, B.; Torre, V.: Using vanishing points for camera calibration. International Journal of Computer Vision, 4(2), 1990, 127–140.

−0.0467	−0.05487	−0.05707	−0.05331	−0.04359	−0.02789	−0.00623	0.0214	0.054989	0.094553	0.140086
−0.0467	−0.0528	−0.0534	−0.04849	−0.03808	−0.02215	−0.00072	0.026217	0.05866	0.096617	0.140086
−0.0467	−0.05074	−0.04974	−0.04369	−0.03259	−0.01644	0.004769	0.03102	0.062319	0.098676	0.140086
−0.0467	−0.04869	−0.04609	−0.0389	−0.02712	−0.01073	0.010241	0.035808	0.065967	0.100728	0.140086
−0.0467	−0.04665	−0.04246	−0.03413	−0.02166	−0.00505	0.015697	0.040583	0.069605	0.102774	0.140086
−0.0467	−0.04461	−0.03883	−0.02937	−0.01622	0.000615	0.021137	0.045342	0.073231	0.104814	0.140086
−0.0467	−0.04257	−0.03521	−0.02462	−0.0108	0.006265	0.02656	0.050088	0.076847	0.106848	0.140086
−0.0467	−0.04054	−0.03161	−0.01989	−0.00539	0.011897	0.031967	0.054819	0.080452	0.108875	0.140086
−0.0467	−0.03852	−0.02801	−0.01517	−2.41E−06	0.017512	0.037358	0.059535	0.084045	0.110897	0.140086
−0.0467	−0.03651	−0.02443	−0.01047	0.005371	0.023109	0.042731	0.064237	0.087628	0.112912	0.140086
−0.0467	−0.0345	−0.02086	−0.00579	0.010729	0.02869	0.048089	0.068925	0.091199	0.114921	0.140086

Tab. 1: Error in Height at respective parametric point in mm.

[2] Canny, J.: A computational approach to edge detection, IEEE Transaction on Pattern Analysis and Machine Intelligence, 8(6), 1986, 679–698.
[3] Cipolla, R.; Drummond, T.; Robertson, D.: Camera calibration from vanishing points in images of architectural scenes, In Proc. of the British Machine Vision Conference, 1999, 382–391.
[4] Company, P.; Piquer, A.; Contero, M.; Naya, F.: A survey on geometrical reconstruction as a core technology to sketch-based modeling, Computers and Graphics, 29, 2005, 892–904.
[5] Colombo, C.; Bimbo, A. D.; Pernici, F.: Metric 3D Reconstruction and Texture Acquisition of Surfaces of Revolution from a Single Uncalibrated View, IEEE Transactions On Pattern Analysis and Machine Intelligence, 27(1), January 2005.
[6] Criminisi, A.; Reid, I.: Zisserman, A.: Single view metrology, International Journal of Computer Vision, 40(2), 2000, 123–148.
[7] Debevec, P. E.; Taylor, C. J.; Malik, J.: Modeling and rendering architecture from photographs: A hybrid geometry and image based approach, ACM SIG-GRAPH, 1996, 11–20.
[8] Farin, G.: Dianne Hansford, Discrete Coons patches, Computer Aided Geometric Design, 16, 1999, 691–700.
[9] Hartley, R.; Zisserman, A.: Multiple view geometry in computer vision, Cambridge University Press,–, second edition, 2004.
[10] Harris, C.: Geometry from visual motion, In A. Blake and A.Yuille, editors, Active Vision, MIT Press, Cambridge MA, 1992, 263–284.
[11] Heuvel Van den F.A.: 3D reconstruction from a single image using geometric constraints, ISPRS Journal of Photogrammetry and Remote Sensing, 53(6), 1998, 354–368.
[12] Lee, Y. T.; Fang, F.: 3D reconstruction of polyhedral objects from single parallel projections using cubic corner, CAD 43, 2011, 1025–1034.
[13] Liebowitz, D.; Zisserman, A.: Metric Rectification for Perspective Images of Planes, Proc. IEEE Conf. Computer Vision and Pattern Recognition, 1998, 482–488.
[14] Liu, Y.; Lee, Y. T.: 3D Reconstruction of Freeform Shapes from 2D Line Drawings, 2010 ACM, Seoul, South Korea, December 12–13, 2010.
[15] Liu, Y. J; Tang, K.; Joneja, A.: Sketch-based free-form shape modeling with a fast and stable numerical engine, Computer & Graphics, 29, 2005, 778–793.
[16] Perkins, D.: Cubic corners: Quarterly Progress Report 89, MIT Research Laboratory of Electronics, 1968, 207–214.
[17] Suh, Y. S.: Reconstructing polyhedral swept volumes from a single-view sketch, IEEE International Conference on Information Reuse and Integration, 2006, 585–588.
[18] Wang, Y.Z.; Chen, Y.; Liu, J. Z.; Tang, X.O.: 3D reconstruction of curved objects from single 2D line drawings. In: Proceedings of the IEEE conference on computer vision and pattern recognition, 2009, 1834–49.
[19] Zhang, Z.: Determining the Epipolar Geometry and its Uncertainty: A Review, The International Journal of Computer Vision, 27(2), March 1998, 161–195.