INTERNATIONAL JOURNAL OF ADVANCE RESEARCH, IDEAS AND INNOVATIONS IN TECHNOLOGY

ISSN: 2454-132X
Impact factor: 4.295
(Volume3, Issue6)
Available online at www.ijariit.com

Algorithmic Study of 3D Reconstruction of Objects

Trupti Lokhande
Department of Computer Engineering, Pune University, India
Dr. D.Y. Patil College of Engineering, Pune, Maharashtra
tlokhande30@gmail.com

Abstract: 3D reconstruction is the process of capturing the shape and appearance of real objects. The task of recovering three-dimensional (3-D) geometry from two-dimensional views of a scene is called 3-D reconstruction. The third dimension plays an important role in the analysis of dynamic or static environments. 3D model of a scene has applications in various computer vision fields, including robotics, virtual reality, and entertainment. There is a large body of 3-D reconstruction algorithms available. Some algorithms only produce a sparse 3-D reconstruction while others are able to output a dense reconstruction. The selection of the appropriate 3-D reconstruction algorithm relies heavily on the intended application as well as the available resources.

We divided the algorithm into two large categories depending on whether a prior calibration of the camera is required i.e. pre-calibrated and online calibrated algorithms.

Keywords: Virtual Reality, Sparse Reconstruction, Camera Calibration, Pre-calibrated, Online Calibrated.

I. INTRODUCTION

In computer vision and computer graphics, 3D reconstruction is the process of capturing the shape and appearance of real objects. The task of recovering three-dimensional (3-D) geometry from two-dimensional views of a scene is called 3-D reconstruction. It is an extremely active research area in computer vision. There is a large body of 3-D reconstruction algorithms available. Some algorithms only produce a sparse 3-D reconstruction while others are able to output a dense reconstruction. The selection of the appropriate 3-D reconstruction algorithm relies heavily on the intended application as well as the available resources. Even though these algorithms may potentially differ in their initial assumptions and sparsity of input and output, the ultimate goal they have in common is to produce a 3-D description of the underlying scene. Because of the vast amount of algorithms available, it is helpful to group them into classes and study each class individually.

II.CLASSIFICATION OF RECONSTRUCTION ALGORITHM

An important step towards emulating the human visual system is the ability to compute 3-D properties of the world from two or more two-dimensional (2-D) images. Algorithms that hope to fulfill this purpose need to operate on the measurements of light in free space. The complete set of all such measurements is known as the plenoptic function $P = \{ Vx, Vy, Vz, \Theta, \phi, \lambda, t\}$ which represents the radiance in space as a function of viewing position Vx, Vy and Vz the angles Θ and in which the light rays pass through the pupil, the wave length λ and time t[2]. In our classification, we differentiate between algorithms that assume a prior calibration of the camera (pre-calibrated) and those that can obtain the calibration at run time (online calibrated). This distinction is significant in practical settings as well. For example, when the goal is to obtain a 3-D model as accurately as possible and we have access to the camera such that a full calibration of it can be performed, it is much more beneficial to select one of the pre-calibrated reconstruction algorithms. However, if we wish to reconstruct a scene from some previously recorded video sequence, we would then have to use an online calibrated reconstruction method. Algorithms within both the pre-calibrated and online calibrated categories are further divided according to the common principles in which the algorithms are designed shown in fig.1

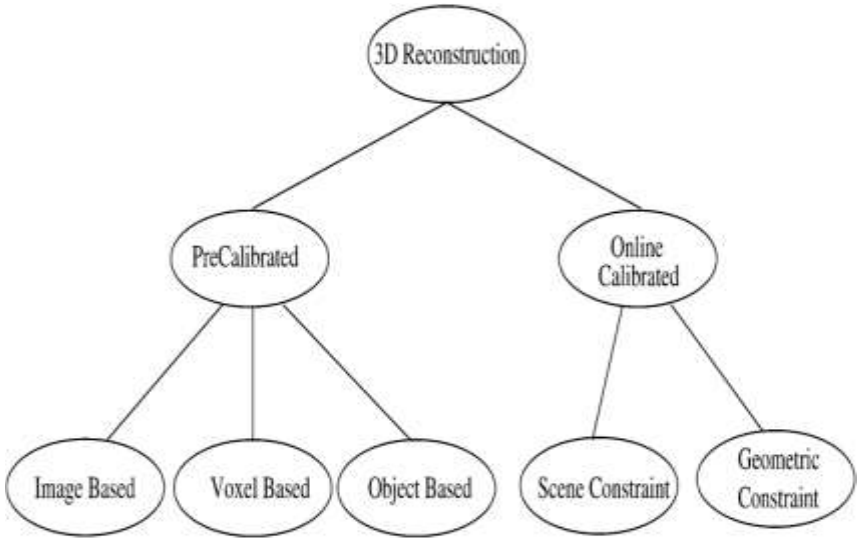

Fig. 1 Classification of reconstruction algorithms

III. PRE-CALIBRATED RECONSTRUCTION

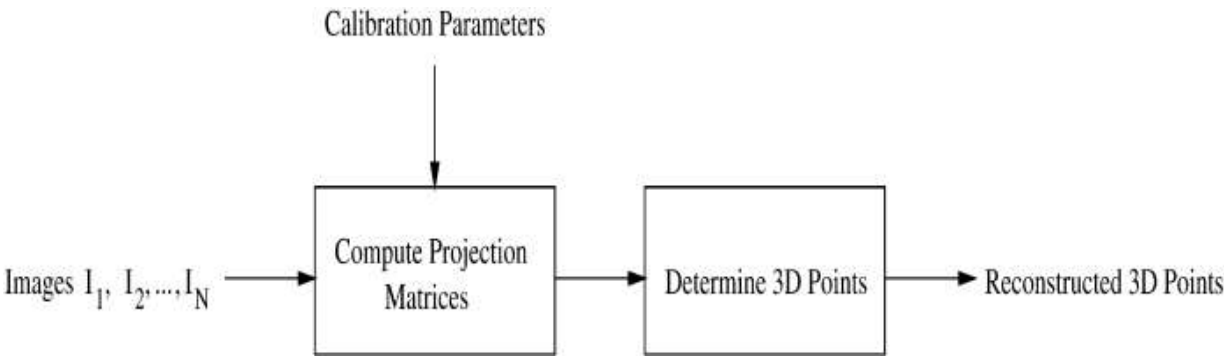

Fig.2 Block Diagram of Pre-Calibrated Reconstruction References

Pre-calibrated reconstruction algorithms are those that require an accurate prior calibration of the cameras. In other words, both the camera's intrinsic and extrinsic parameters need to be computed. One of the most popular camera calibration technique was developed by Tsai [11]. This method relies on the availability of a 3-D calibration object with special markers on it. In addition, it is required that the markers are not all coplanar. This calibration object provides a correspondence between points on the image and 3-D points in space. A more practical method was recently proposed by Zhang [13]. This method only requires the camera to observe a planar pattern in at least two different orientations. The camera calibration toolbox [7] is an efficient implementation of calibration algorithms.

It can be used both as a standalone application in Matlab or as part of the Intel Computer Vision Library [5]. Using the calibration parameters, the projection matrices can be trivially computed. The block diagram for calibrated reconstruction algorithms is shown in Fig.4.

A. Image-Based Reconstruction

Using sparse image features or dense pixel matchings are both considered image-based methods. Reconstruction of selected image features or dense matchings between images follows a very similar path. The only difference is the amount of data processed. The method used to determine the 3-D point from pairs of matched image pixels (the last step) is usually triangulation. In the absence of noise, triangulation is trivial. However, with the presence of noise, the triangulation problem is much more complicated as the back projected rays from the two images will not generally meet in 3-D space. We, therefore, need to find a suitable point of "intersection".

Because of the known camera calibration, our reconstruction is metric. Hence the concept of distance and perpendicularity is clearly defined. Therefore, the simplest method is to estimate the required 3-D location as the midpoint of the common perpendicular to the two back projected rays. In addition, with an assumed Gaussian noise model, a provably optimal triangulation method is available. From a pair of point correspondences and, this algorithm seeks an alternate pair and such that the sum of squared distances to the original pair of points is minimized subject to the epipolar constraint. Thus, the optimal points to select are those that lie on a pair of corresponding epipolar lines closest to the original point correspondence. This pair of epipolar lines, and, can be found the minimizing the distance between them and the original point correspondence.

1. Feature Detection and Correspondence

For feature- based algorithms, the crucial steps to an accurate reconstruction are feature detection and feature correspondence. The pioneer work on feature detection is done by Moravec in [4]. The features, or "points of interest", are defined as locations where large intensity variations in every direction occur. The un-normalized local autocorrelation in four directions are computed and the lowest result is taken as the measure of interest. These measurement values are then threshold with non maximum suppression applied. Harris and Stephens [1] built on the idea of the interest operator but used the first order image derivative to estimate local autocorrelation.

Now given the features extracted from the images, they need to be matched over the sequence of the images. This step involves finding correspondences across images. To find the corresponding features we need feature descriptors. The feature descriptor will produce descriptive information for each feature which is then used to match the features across several images. Any desired descriptor should be invariant to rotation, scaling and affine transformation. Thus the same feature needs to be identified on different images for a possible match. Point descriptors can be classified in the following categories Distribution based, spatial-frequency based, Differential descriptors and moment based descriptors. Distribution based descriptors use histograms of the region.Spatial descriptors are based on texture classification and description; uses a differential descriptor, where a set of local central moments of a region with different order and degree forms to describe the region.

2. Dense Stereo Matching

Similar to feature-based algorithms, the success of dense reconstruction algorithms relies heavily on the accuracy of the densely matched pixels. When the sampling along the time axis is also dense, the pixel displacements between consecutive frames can be approximated by optical flow [6]. Assuming that the camera is a rigid body, the motion field of the projected points on the optical plane satisfies the differential equation

$$\dot{\mathbf{X}}_{cam} = \boldsymbol{\omega} \times \mathbf{X}_{cam} + \mathbf{v}$$

Where w and v are angular and linear velocities of the camera respectively. It can be shown that the relationship between the image plane motion field u(x) = [ux, uy]T and the motion of the camera can be expressed as

$$\mathbf{u}(\mathbf{x}) = \frac{1}{Z} A(x)\mathbf{v} + B(x)\boldsymbol{\omega}$$

Where Z is the depth A(x) and B(x) are as defined in [3].

Thus, we see that the optical flow field has two components generated by the angular velocity and the linear translation respectively. Also, it is clear that no structure information is contained within the angular component of optical flow. This confirms our intuition that scene structure cannot be computed from images taken by a rotating camera.

B. Voxel-Based Reconstruction

Because of the rapid growth of computational storage and processing power, volumetric representation of scene structure becomes practical. Various approaches to recover volumetric scene structure from sequences of images have been proposed. Earlier attempts to solve this problem approximate the visual hull of the image objects. The visual hull of an object is defined as the maximal shape that gives the same silhouette as the actual object for all views outside the convex hull of the object. Methods that approximate the visual hull are referred to as volume intersection or silhouette intersection. Similar to the convex hull, the visual hull is an approximation of the actual shape of the object. However, the size of the visual hull decreases monotonically with the number of 2-D images.A necessary pre-processing step to compute the visual hull is the segmentation of each 2-D image into object foreground and background. The segmented 2-D silhouettes from each image are then back projected and intersected to yield a volume segment representation which can be further processed into a surface description.

In voxel occupancy, the scene is represented by a set of voxels and the algorithm labels each voxel as being either filled or empty.

An alternative volumetric reconstruction method utilizes color consistency to identify surface points in the scene. Color consistency requires that the projection of all surface point in different views to have consistent color. Specifically, if a non occluded point belongs to the surface of the object, then its projection onto different views should have approximately the same color. Conversely, the projections of a point not on the surface usually do not have a consistent color. Color consistency can be mathematically defined as the standard deviation, the L1, L2, L∞, or norm of the colors of the pixels that a particular voxel projects to in each view. Algorithms that exploit color consistency are all variants of the voxel coloring approach. In practice, voxel coloring starts by initializing the scene with only opaque voxels. As the algorithm iterates, voxels are tested for color consistency. Only consistent voxels are kept while inconsistent voxels are carved out. The algorithm stops when all voxels are consistent. As voxels often occlude each other, it is vitally important for voxel coloring algorithms to determine the visibility of a particular voxel before performing the consistency test.

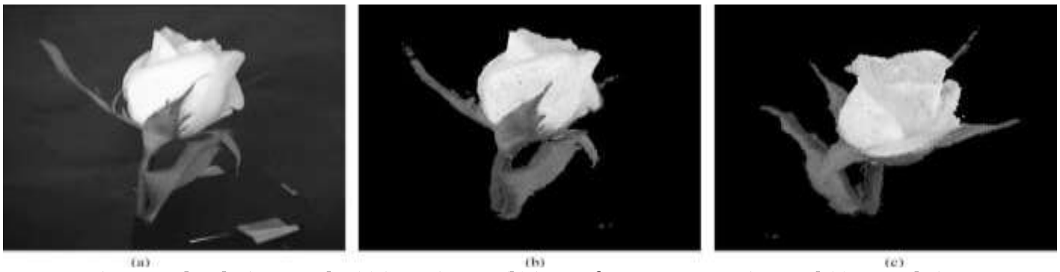

Fig.3 Voxel Coloring Results (a) input image, (b) image from reconstruction, and (c) a novel view.

C. Object-Based Reconstruction

While voxel-based reconstruction algorithms fill the scene with voxels and the reconstruction algorithm determines the visibility of each voxel, object-based reconstruction algorithms aim at recovering a surface description of the objects in the scene. The level-set reconstruction method proposed by Faugeras *et al.* [9] is the first object centered 3-D reconstruction technique from image sequences. Applying variational principles used in their previous work for dense depth recovery reformulated the reconstruction problem into a surface evolution problem that can be solved using the level-set technique.

The level-set approach starts by choosing a surface that minimizes some energy functional

$$E(S) = \int_S \Phi(\mathbf{X})dA.$$

Therefore, we need to choose a $\Phi(X)$ such that it is small at good matching locations and large otherwise.

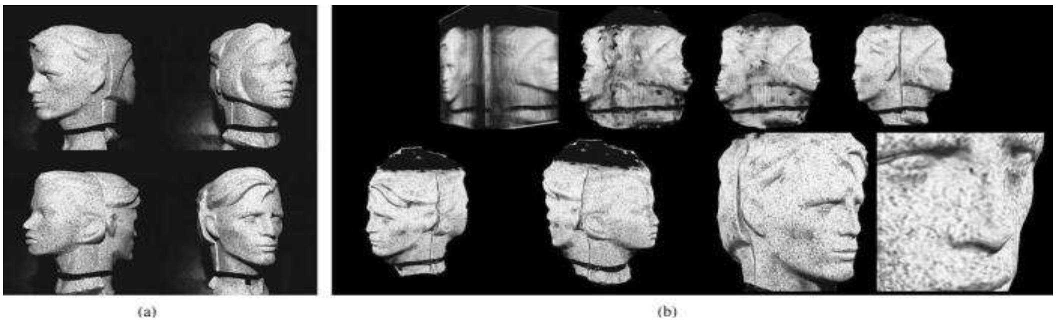

Fig.4 Reconstruction using the level-set method (a) input images and (b) evolving reconstruction results

Once $\Phi(X)$ is chosen, a surface which minimizes equation can be found using gradient descent. A simple choice of $\Phi(X)$ can be the summed square error of the matching pixels

$$\Phi(\mathbf{X}) = \frac{1}{n}\sum_{i \neq j} \Phi_{ij}(\mathbf{X})$$

$$\Phi_{ij}(\mathbf{X}) = (I_i(\mathbf{x}) - I_j(\mathbf{x}'))^2.$$

However, this choice of Φ is very sensitive to noise and local texture. Essentially, the level-set reconstruction method evolves a time varying surface toward the objects in the scene.

IV. ONLINE CALIBRATED RECONSTRUCTION

Under various circumstances, we may not have the luxury of performing the calibration task using a pre-made calibration object. It is a very realistic scenario for a number of applications. For example, in a video indexing application, we are only given the final video data without even knowing what type of video camera these videos were taken from, not to mention getting the same camera to take a video sequence of the calibration object.

Furthermore, the intrinsic parameters of the camera may be changing during the acquisition of the video sequence because of focusing and zooming. Therefore, 3-D reconstruction tasks under these scenarios would have to be performed using online calibrated reconstruction methods.

The key difference between online calibrated reconstruction methods is the way the camera's parameters are estimated. This on-the-fly estimation of camera parameters is often referred to as camera self calibration or auto calibration. Auto calibration methods can be divided into two classes: those that exploit scene constraints and those that exploit geometric constraints. Hence, we divide the online calibrated reconstruction methods using these two classes. A block diagram for typical online calibrated reconstruction algorithms is shown in Fig. 5

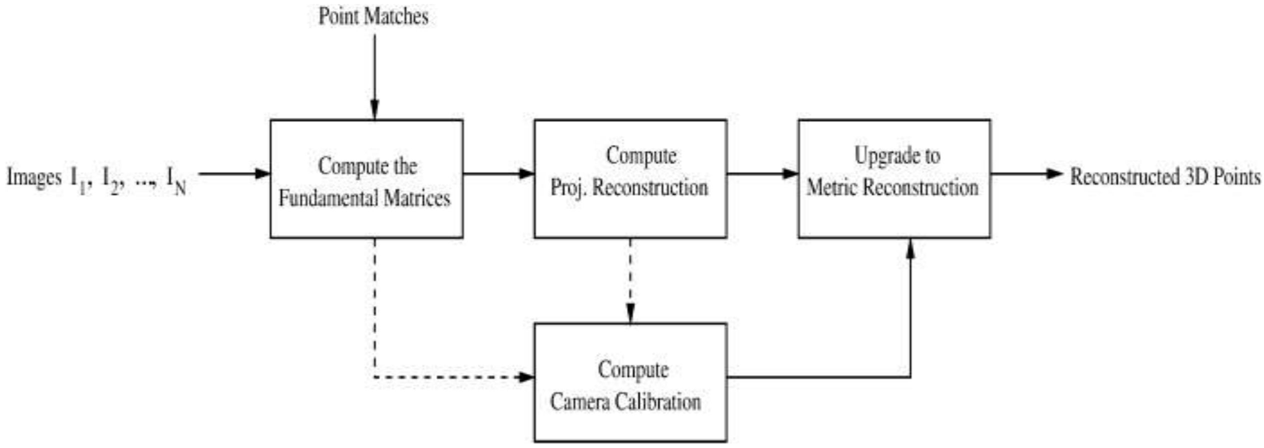

Fig.5 Block Diagram of Online Calibrated Reconstruction

A. Projective Reconstruction

Using the estimated fundamental matrix, we can compute a projective reconstruction of the 3-D scene. A projective reconstruction consists of a set of 3-D points {Xi} and a set of camera projection matrices {Pi}. However, being a projective reconstruction means that the reconstruction is determined only up to a projective transform. Thus, for any projective transformation H, {PiH-1}, and {HXi} and yield an equally valid reconstruction.

From the estimated fundamental matrix between two views, a pair of projection matrices can always be retrieved. The two camera matrices are chosen to be P = [I|0] and P' = [[e']xF|e']. Using these projection matrices, we can compute a projective reconstruction of the detected features with a simple linear triangulation method. Since x = PX and x' = P'X, we can eliminate the scale factor by taking the cross product of the left hand side with the right-hand side. Doing so produces three linear equations for each point, out of which two are linearly independent. Stacking the equations for the corresponding points and writing them in matrix form, we have

$$AX = \begin{bmatrix} xp^{3T} - p^{1T} \\ yp^{3T} - p^{2T} \\ x'p'^{3T} - p'^{1T} \\ y'p'^{3T} - p'^{2T} \end{bmatrix} \mathbf{X} = \mathbf{0}$$

Where piT are the rows of the projection matrix P. The solution of above matrix is the singular vector corresponding to the smallest singular value of A.

B. Calibration Using Scene Constraints

If images are taken within constrained environments, the camera self calibration step can usually be simplified considerably. One example of such constrained environment is the architectural or man-made scene. The most noticeable characteristic of this environment is that it contains a large number of parallel lines. Parallel lines in each direction intersect at a point on the plane at infinity. The projection of these intersection points are the vanishing points. The knowledge of the position of the vanishing points in three dominant directions in an image greatly simplifies the determination of the intrinsic parameters of the camera used to produce this image. In fact, we can obtain closed form solutions for these parameters as a function of the vanishing points.

Before we can take advantage of the parallel structures within the scene for self calibration, the vanishing points have to be computed first.

1. Computing the Vanishing Points

The estimation of vanishing points from detected line segments can be divided into two steps: the accumulation step and the search step. The goal of the accumulation step is to use the detected line segments to vote for some location in the accumulation space which could potentially share the same vanishing point. The search step searches the accumulation space for cells that possess a large number of votes. After the accumulation process, the search step is performed to find the vanishing points. It repeatedly searches for the most dominant cell and removes all line segments corresponding to it until the maximum vote drops below a certain threshold.

C. Calibration Using Geometric Constraints

The more flexible class of algorithms can perform self calibration using only geometric constraints that are inherently available in the image sequences themselves. In this section, we shall examine the underlying principle that allows us to perform a self-calibration. Several different methods will be presented and analyzed.

CONCLUSION

This paper has surveyed a number of 3-D reconstruction algorithms. We divided the algorithm into two large categories depending on whether a prior calibration of the camera is required. Under the pre-calibrated reconstruction category, we discussed image-based algorithms that rely on feature correspondence or dense stereo matching to compute the reconstruction of the scene, voxel based algorithms that project the scene filling voxels back onto each view to determine its visibility and object-based algorithms that formulates the 3-D reconstruction algorithm as a level-set evolution problem in which a system of partial differential equations gradually converges to the object to be reconstructed.

On the online calibrated side, we discussed algorithms that take advantage of parallel structures in the scene to compute the vanishing points which aid greatly in computing closed form solutions of the camera's calibration parameters. In addition, we also reviewed more flexible self-calibration methods that do not rely on specific prior scene structure but only on inherent geometric constraints such as the position of the absolute conic.

REFERENCES

[1] C. G. Harris and M. Stephens, "A combined corner and edge detector," in *Proc. 4th Alvey Vision Conf.*, 1988, pp. 147– 151.

[2] D. D. Morris, K. Kanatani, and T. Kanade, "Gauge fixing for accurate 3D estimation," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 2001, pp. 343–350.

[3] D. J. Heeger and A. D. Jepson, "Subspace methods for recovering rigid motion i: Algorithm and implementation," *Int. J. Comput. Vis.*, vol. 7, no. 2, pp. 95–117, 1992. [

4] H. P. Moravec, "Toward automatic visual obstacle avoidance," in *Proc. Int. Joint Conf. Artificial Intelligence*, 1977, pp. 584–584.

[5] Intel Corporation. (2004) Intel Open Source Computer Vision Library. [Online]. Available: http://www.intel.com/research/mrl/research/opencv/

[6] J. L. Barron, D. J. Fleet, and S. S. Beuchemein, "Performance of optical flow techniques," *Int. J. Comput. Vis.*, vol. 12, no. 1, pp. 43–77, 1994.

[7] J.-Y. Bouguet. (2004) Camera Calibration Toolbox for Matlab.[Online].Available:http://www.vision.caltech.edu/bouguet/calib_doc/

[8] M. Pollefeys and L. vas Gool, "Stratified self-calibration with the modulus constraint," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 21, pp. 707–724, Aug. 1999.

[9] O. Faugeras and R. Keriven, "Variation principles, surface evolution, PDE's, level set methods, and the stereo problem," *IEEE Trans. Image Processing*, vol. 7, pp. 336–344, 1998.

[10] R. Hartley and P. Sturm, "Triangulation," *Comput. Vis. Image Understanding*, vol. 68, no. 2, pp. 146–157, 1997.

[11] R. Tsai, "A versatile camera calibration technique for high accuracy 3D machine vision metrology using off the shelf TV cameras and lenses," *IEEE J. Robot. Automat.* Vol. RA-3, no. 4, pp. 323–344, 1987.

[12] "Visual mapping by a robot rover," in *Proc. 6th Int. Joint Conf. Artificial Intelligence*, 1977, pp. 598–600.

[13] Z. Zhang, "A flexible new technique for camera calibration," *IEEE Trans. Pattern Anal. Machine Intel.* vol. 22, pp. 1330–1334, Nov. 2000. [

14] C. H. Chien and J. K. Aggarwal, "A volume/surface representation," in *Proc. Int. Conf. Pattern Recognition*, 1984

[15] Jae Hyun Won, Man Hee Lee and In Kyu Park," Active 3D Shape Acquisition Using Smart phones" in *IEEE 2012.*

