Generating 3D Model Using 2D Images of an Object

*K SusheelKumar, **Vijay Bhaskar Semwal, ***Shitala Prasad, ***R.C Tripathi

*Department of computer science & engineering, Ideal Institute of Technology,
Ghaziabad-201003, India

**Product Engineering Group (PES)-KM Group, Newgen Software technology
A-125, sector 63 Noida

***Department of Human Computer Interaction, Indian Institute of Information Technology,
Allahabad-211012, India

Abstract:

In this paper we implemented 3D reconstruction is a process of regenerating 3D information of an object using its 2D images. It's been an important part of computer vision studies. Computer vision deals with automatic extraction of various kinds of information from images. The main of aim of machine vision is to let machine visualize the world as we humans do and let them interact it. Various algorithms have been discussed for reconstruction purpose. Present research paper gives an overview of some of these algorithms and then it discusses the implementation issues of one of that algorithm. Volumetric reconstruction algorithms are getting popular due to their less complexity and the increasing storing capability of the computers. This research paper focuses on how the voxel coloring algorithm behaves when operating on different now of images to reconstruct and when it has to operate on different resolution.

Keywords: 3D reconstruction; voxel coloring algorithm; extraction of images.

1. INTRODUCATION

3D reconstruction is a process of regenerating 3D information of an object using its 2D images. It has been an important part of computer vision studies. Computer vision deals with automatic extraction of various kind of information from images. The main aim of machine vision is to let machine visualize the world as we humans do and let humans interact with it.

One of the purposes of computer vision is the creation of computer models of 3D objects from its digital images. In general, image processing works on 2D images of an object so it misses the depth parameter of that object. Depth information of an object is very useful and plays a significant role in many decisions. The main aim of 3D reconstruction is to regenerate that depth information from simple 2D images of an object. 3D reconstruction produces a 3D model as an output, which could be stored and modified in future.

Reconstruction of the shape of 3D objects from a series of digital images is a challenging problem. When we use CAD kind of applications for 3D reconstruction then a lot of description has to be given by the modeler with the help of some GUI. We have to provide some textual description and have to define polygon patches according to that description. This kind of reconstruction is low in degree of realism and takes a lot of human effort. So the automated modeling becomes important which uses the images taken by the CCD camera.

A number of setups have been suggested for 3D reconstruction some of which are based on single camera and others are based on multiple cameras. In the case of single camera we can also divide the methods on basis of either the object is moving or camera is moving. In cases where is movable the object is generally placed on a turntable and multiple views are taken by changing the turntable rotational position. For reconstruction, calibration information about the camera and the camera position should be known. Similarly multiple cameras can be used for gathering multiple views of an object. In that case calibration and the extrinsic information are needed for each camera. Active methods use some source to highlight the shape of the object like structured light or some array of laser beams and one capturing device likely to be a CCD camera while passive methods works on the normal light with some camera to capture the views.

This research paper contains a brief overview of various popular 3D reconstruction techniques and then discusses one of the 3D reconstruction techniques, Voxel Coloring which uses photoconsistency to derive the shape of object.

II PROBLEM STATEMENT

This research works on reconstruction method for textured objects. The object is sampled and reconstructed in several steps and every step has different kind of challenges. The first step of the image based 3D modeling is image acquisition. We used one camera for the image acquisition and a rotating table where the object is placed.

So at first the camera must be calibrated for finding its intrinsic and distortion parameters. This calibration information is used later for determining the position of camera, in form of extrinsic parameters. The main approaches the volume reconstruction process with concept of voxels. When working with voxels, memory requirement for reconstruction plays a significant role because voxels take a huge amount of storage. So, we should define the voxel cube such that it can approximate the area of attention in the best way. We are using color images for the purpose of reconstruction so it is beneficial that we could use the color information of the object for shape reconstruction. Traditional volume intersection methods use techniques which add some extra voxels to the reconstruction. So it is a significant challenge to remove those extra voxels to find a better reconstruction. While working with voxels computing visibility of voxel from a particular viewpoint is significant because then only we can account image from that viewpoint as additional information for reconstruction. Only voxels from the surface are visible from all views so computing visibility also significant for further processing on voxel like consistency check. One problem usually faced during visibility computation is determining where a voxel of concern is heading towards.

III OVERVIEW OF 3D RECONSTRUCTION TECHNIQUE

A. Volumetric Reconstruction

As the computational storage and the secondary storage capacity increased the volumetric based approaches would become feasible to implement because they need huge amount of storage at the time of reconstruction. Various algorithms been given for reconstructing the 3D volume from sequence of images. Previously most methods solve this task by approximating a visual hull which is a maximal shape that gives the same silhouette as the original object. Visual hull is approximated shape of an object. As many 2D inputs are there for the reconstruction the size of visual hull got reduced that much. Laurentini [4] discussed the visual hull in detail. In visual hull based methods first the foreground is separated from the background then the segmented silhouette from each image is back projected and intersected to generate the final volume. Further processing can be done to restore the color information. Visual hull based algorithms are fast in terms of computation and re-re-rendering the final reconstruction.

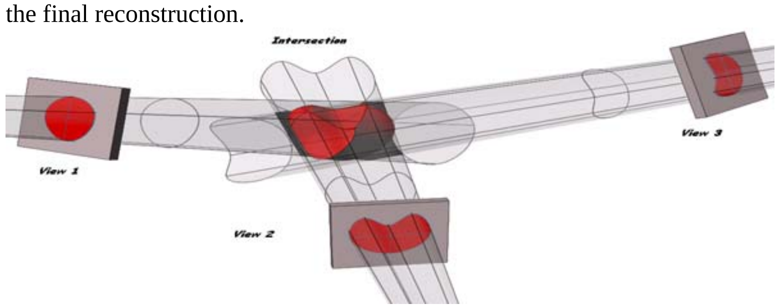


Figure 1. An example of visual hull reconstruction [8]

Visual hull based algorithms can generate surprisingly good results with help of relatively less number of views (4-8) of the object. But some excessive volume is created in this approximation of volume through visual hull algorithm due to existing concavity in object and insufficient view of that perspective. Though visual hull algorithms are fast in nature in comparison to other stereo based methods but these are deterministic in nature and don’t consider that some uncertainty might be present in views. Visual hull based algorithms are quite sensitive to segmentation errors. Segmentation error in a single view of the object might bring huge change in the 3D model generated [9].

a. Voxel Occupancy

Voxel occupancy is one of the well known methods for defining a 3D shape. In voxel occupancy the volume defined with help of some voxels which together define the shape of object [6]. First step in this method is to initialize the bounding box for the reconstruction and then that bounding box is divided into small voxels. Each voxel is then tested with help of multiple views which one is inside the object, which one is on the boundary and which one is outside. This process is called voxelization. Mostly voxel occupancy based methods use silhouettes because of this it also called shape from silhouette. In this method first the object's silhouette computed from each camera by deducting the background image from the image consisting the object. Each silhouette is back-projected and if the voxel comes inside the intersection of these back projections then it is considered as a part of the volume to be reconstructed. Algorithms for computing this intersection been proposed in [6].These voxels might be represented with help of octrees**.**

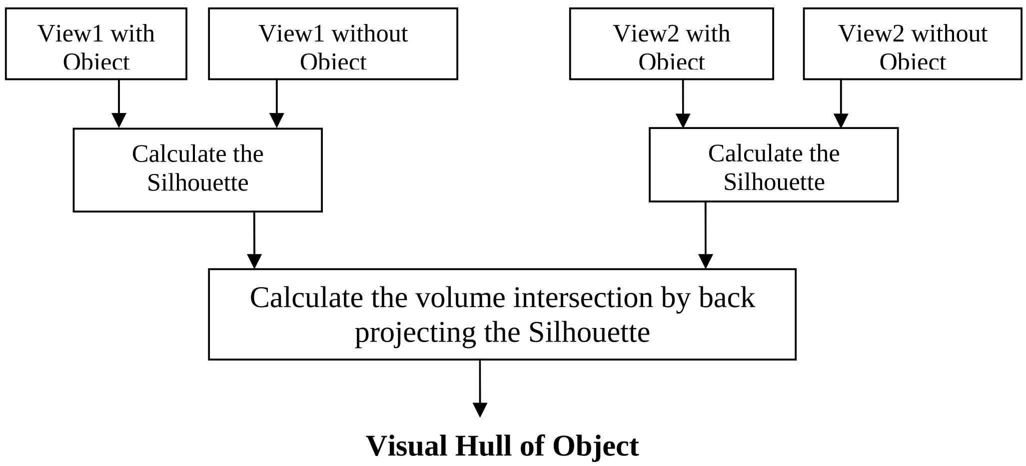


Figure 2. Overview of Shape from Silhouette

The sampling setup for acquiring the images also differs from the multi camera model to single camera model. Setup with single camera generally mounted on fixed location and the object mounted on a turntable [7]. This enables the setup to take multiple views of an object with help of one camera without disturbing the background. Silhouettes act as hard constraints while calculating the volume with silhouette intersection. There might be some error in computation of silhouette for example the pixel value of the object and the background might be same. A single pixel can cause a drastic change in the reconstructed object and can lead to a hole. That's why some morphological operator used to clean up either the 2D images or the 3D volumes. Snow et al. [6] proposed one method which doesn't compute the silhouette from each image and allows better spatial smoothness in the reconstruction. Input of voxel occupancy algorithms are some images from calibrated cameras. And each image use to be coupled with a background image which was taken without the object in the scene. Output would be binary labeled volume where the voxel labeled with 1 if it is occupied and labeled with 0 if it is empty [6]. If voxel v belongs to some volume V we will write the label f assigned to voxel v as fv. An energy minimized function proposed by Snow et al. [6]

$$E(f)=\sum_{v\in V} D_v(f_v) + \lambda \sum_{v,v'\in N(v)} \left(1-\delta(f_v - f_{v'})\right)$$

f_v : Label on voxel v if it belongs to final volume V

$D_v(f_v)$: A penalty for assigning f_v label to voxel v

λ : a penalty for assigning different labels to pair of adjacent

δ : Unit impulse function (1 at the origin and 0 elsewhere)

This method based on obtaining a labeling f^* that minimizes E (f0. There is a global local minima for E(f) at f^*

b. Voxel Coloring

In voxel coloring every voxel is labeled with its color and transparency. Voxel coloring method is based on the color consistency. If a voxel v has same color in all images from where it is visible then it is labeled with that color. If a voxel has different color in various views of the object then it would be declared as transparent.

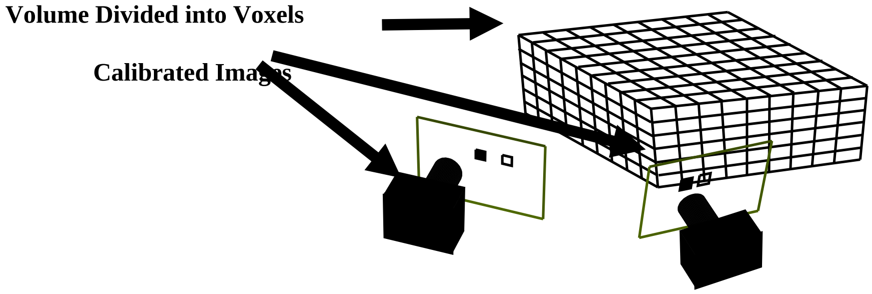
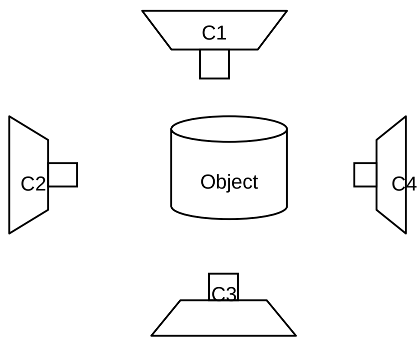
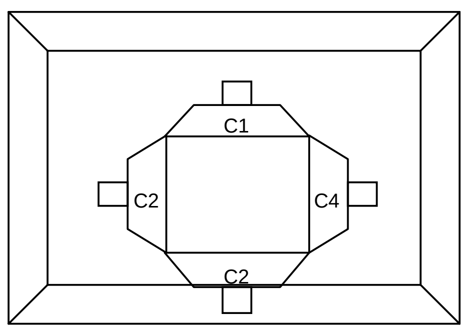


Figure 3. Overview of Voxel Coloring [10]

Voxel coloring algorithm puts constraint on the location of the cameras. It prefers that all cameras should be in such condition such that voxels could be visited in a single scan so that transparency of all voxels that might occlude the present voxel should be calculated before [14]. Two kinds of setups are there for scanning the surface



(a) Image acquisition when cameras surrounding the object



(b) Image acquisition when camera is inside the scene

Figure 6. Camera setup for the image acquisition [10]

The voxel coloring algorithm emphasizes on detecting whether a point in image corresponds to a surface pixel of the object or not. For this calculation it accounts the photoconsistency of that point by comparing the light coming from that point in various images. A point on the surface should have same color in various views if it's not an occluded one. This phenomenon is photoconsistency and used in voxel coloring to find the pixels on the surface. If light coming from any unconcluded part is not photoconsistent then that point should be marked as empty. With the help of lambertian surface the reflectance value could be calculated for each angle and the expected color could be matched accordingly [10]. For each view the bounding box or the approximated volume divided into layers. The layers could be based on the distance from the camera so that voxels can be processed layer by layer. For each layer firstly the visibility of voxel is calculated and the voxels are labeled. After finding the visible voxels a set of colors use to be find to know the photoconsistency of the voxel. For each voxel the photoconsistency checked using the views from which it is visible if it is photoconsistent then the voxel is marked otherwise it is deleted. Due to external occlusion and self occlusion all voxels are not visible in all images. Voxels which are not on the surface would not be visible in any image. If the non-surface voxels can be removed then the computational complexity of voxel coloring can be reduced significantly. We can calculate the surface voxels by identifying those voxels whose at least one out of six neighboring voxels is not in the object's volume [16]. More than one voxel can project the same region in an image of the object but the closest voxel to the camera attribute to the color of the voxel. A depth map is used to indicate which voxel is near to the camera. We can denote this depth of voxel v from j image as Depthj (v).

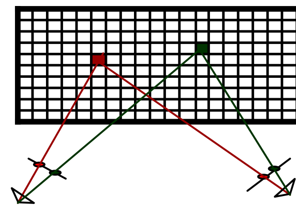


Figure 7. Voxel coloring with help of two views of a scene [10]

Result of the algorithm is a scene model that is consistent will be consistent with all of the pixels in all input views. This process removes all opaque voxels until every border voxel is photo-consistent. The closest photo-consistent voxel along each visual ray is assured to be on the surface of the final volume. The final volume could be thing as union of all photo-consistent scene reconstructions. For this reason it is called the photo hull [17]. Voxel coloring is one pass algorithm. For general camera configuration one pass algorithm is not feasible because voxels cannot be topologically sorted with respect to all view points.

Voxel carving algorithm can be summarized as follows [17]

Step 1: Initialize V (the volume which contains the final reconstruction)

Step 2: While consistent (v) =true; where $v \in$ Surface (V)

a. Calculate all photographs from where v is visible. Say the set of such photographs P_v .
b. Project v to all photographs in P_v .
c. Store the pixel colors C[1]; : : : ; C[j] to which v projects and calculate the optical rays R[1],……R[j] which connects v to the corresponding optical centers.
d. Determine the photo-consistency of v using calculate_consist(C, R).

Step 3: If consistent (v) =true; $\forall$ v, where $v \in$ Surface

set V* = V and terminate.

Else

set V = V – {v} and repeat Step 2.

The main problem with voxel carving is it's difficult to know on which iteration the computation should be stopped.

IV. METHODOLOGY

A. Voxel Coloring

Basic steps in reconstruction with help of voxel coloring are:

1. Calibration of the camera for intrinsic parameters
2. Acquisition of images of the object
3. Extrinsic parameters calibration for different images
4. Background removal from the images
5. Initialize the initial volume to be carved
6. Divide the volume into various voxels on the basis of the resolution of the reconstruction which would be specified by the user
7. Divide the array of voxels into layers *L*
8. For each voxel in layer *L* repeat 9 to 10
9. Back project voxels in all images one by one and if it's visible there then calculate the consistency of voxel using various consistency checks
    a. Original consistency check
    b. Adaptive consistency check
    c. Minkowsky distance based check
    d. Histogram based consistency check
10. If the voxel is consistent then involve this into final volume otherwise carve that voxel from the volume

B. Consistency Check

Basic Consistency Check

Seitz at el. [11] check the consistency of the voxel using the standard deviation of color in that image portion which corresponds to the voxel.

$$consistent(v) = \begin{cases} ture & \sigma \le \tau \\ false & otherwise \end{cases} \tag{1}$$

Where σ is standard deviation of a set of pixels π. Here π is set of pixels in the image which corresponds to the voxel currently considered for consistency.

Adaptive Consistency Check

Textured regions usually have a higher color variance than regular regions so no single threshold is prefect for an entire scene. When a voxel is visible in an image, it is usually visible from many pixels in the image so pixels have a span of region of different color. So there is a requirement of adaptive threshold that is would change according to the color variation seen from a single image [22]. If π_i^V is the set of pixels in image *i* from which the voxel *V* is visible and $\sigma_{\pi_i^V}$ be the standard deviation of pixel values. If $\bar{\sigma}$ is the average of $\sigma_{\pi_i^V}$ for all images *i* from where the *V* is visible then we can define adaptive consistency as follows

$$\sigma_{\pi_i^V} \le t_1 + t_2\bar{\sigma} \tag{2}$$

$$\bar{\sigma} = \frac{1}{p-1}\sum_{i=0}^{p-1} \sigma_{\pi_i^V} \tag{3}$$

This test is called adaptive standard deviation test (ASDT). Where t_1 and t_2 are thresholds for whom the values are to be set manually. ASDT model has fewer holes than the other models.

Minkowsky Distance Based Check

We can also define photoconsistency of a set using Minkowsky distances, L_1, L_2 and L_∞. Minkowsky distance between two point u and v is given by following equation [15]

$$L_p(x, y) = \left(\sum_{i=0}^{k} |u_i - v_i|^p\right)^{1/p} \tag{4}$$

If a voxel *v* is visible from *p* views, I_0, Ii; …. I_{p-1}, and the color sets extracted from these images are $\pi_0, \pi_1, \ldots\ldots, \pi_{p-1}$. The distance between color entities of any two color sets should be certain. With this concept of Minkowsky distance we can define photoconsistency using the following relation

$$consistent(v) = \begin{cases} true, \forall_{i,j}\, consistent_{i,j}(v) \\ false \quad otherwise \end{cases} \tag{5}$$

$$consistent_{i,j}(v) = \begin{cases} true, & \forall_{C_l \in \pi_i, C_m \in \pi_j} L_p(C_l, C_m) < t \\ false, & otherwise \end{cases} \tag{6}$$

One benefit of the Minkowsky distance is if during the photoconsistency check, any voxel is found to be inconsistent then we can stop the further computation. That means, if we found a pair of pixels whose distance is more than or equal to the threshold then we can say that voxel is not consistent.

Histogram Consistency Check

Standard deviation and Minkowsky distance L1 are not very good consistency measures as they can't differentiate between clustered color distributions and purely random distributions but histogram consistency check can deal with this [22, 23].

V. IMPLEMENTATION DETAILS

Details of the implementation can be summarized as follows.

Main_reconstruction

It defines the location of the dataset that will be used for the reconstruction and the path for the viewer to be used to show the final reconstruction. It defines the voxel coloring space settings and then call the voxel coloring module vxlclr_reconstruct.

vxlclr_reconstruct

Load images from the disk with help of vxlclr_load_images. Then it calls the reconstruction module ‘vxlclr’ with image details, camera position details and the voxel coloring space settings. It finally calls a routine vxlclr_save_volume for saving the final reconstruction.

vxl_load_images

It loads the images and then caches them in one file for further process. It also loads the camera calibration parameters and then makes a list for camera parameters for further use.

vxlclr

It takes image; calculate the step voxel size using the user specified voxel space settings. It projects the voxel in the image and then calculate how many pixels are visible belongs to the voxel. Then calls the consistency_check for detecting whether the voxel is consistent or not. If the voxel is consistent then its considered in the final

consistency_check

It calculates the consistency of voxel using various consistency checks which can be chosen by the user for reconstruction.

a. Original consistency check
b. Adaptive consistency check
c. Minkowsky distance based check
d. Histogram based consistency check

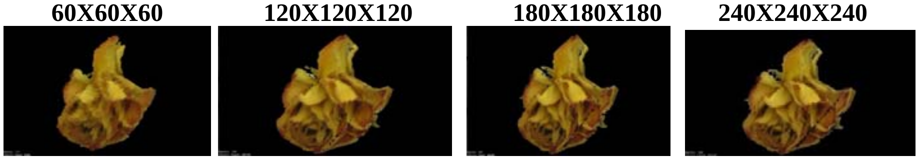
vxlclr_save_volume

It takes the array of voxels, then squeezes it to final volume and then saves it too finally**.**

VI. RESULTS AND DISCUSSION

A. Results Obtained

60X60X60 **120X120X120** **180X180X180** **240X240X240**

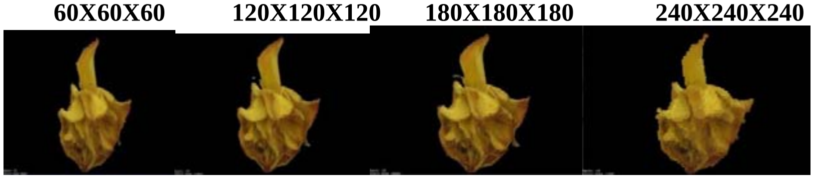


Resolution	Time(sec.)	Size	No. of voxels
60X60X60	35.969	844	5201
120X120X120	81.204	6751	20736
180X180X180	162.735	22782	44645
240X240X240	295.703	54001	75248

Table 1. Results for output from two images at various resolutions

B. Different Resolution Results for Set of 23 Images

60X60X60 **120X120X120** **180X180X180** **240X240X240**



Resolution	Time(seconds)	Size(KB)	No. of voxels
60X60X60	345	844	11374
120X120X120	864	6,751	53267
180X180X180	1572	22,782	113512
240X240X240	2262	54,001	186024

Table 2 Results for output from 23 images at various resolutions

C. Same Resolution with Different Set of Images (60X60X60)

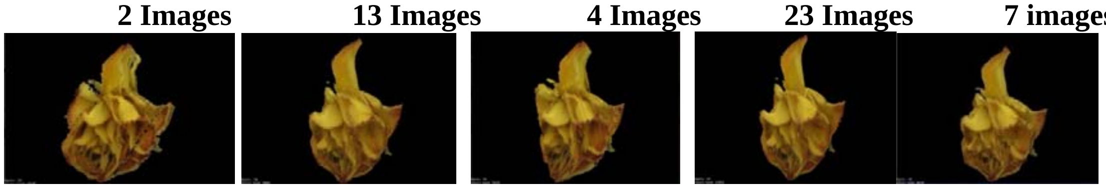
2 Images **13 Images** **4 Images** **23 Images** **7 images**

No of Images	Time(seconds)	Size	No of voxels
2	35.969	844	5201
4	66.547	844	8737
7	140.296	844	8037
13	201	844	9737
23	345	844	11374

Table 3 Output at 60X60X60 for various set of Images

D. Same Resolution with Different Set of Images (120X120X120)

2 Images **13 Images** **4 Images** **23 Images** **7 images**



No of Images	Time	Size	No of voxels
2	81.204	6751	20736
4	148.032	6751	36056
7	275.922	6751	37526
13	444	6751	46670
23	864	6751	53267

Table 4 Output at 120X120X120 for various set of images

E. Same Resolution with Different Set of Images (180X180X180)

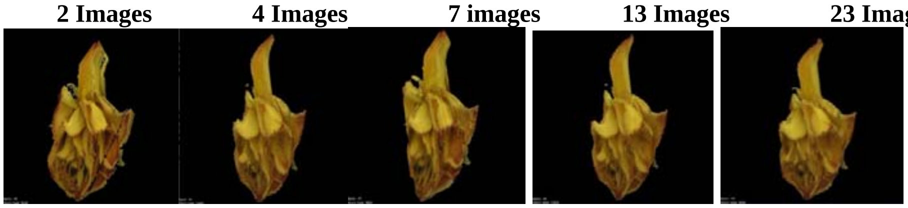
2 Images **13 Images** **4 Images** **23 Images** **7 images**

No of Images	Time	Size	No of voxels
2	162.735	22782	44645
4	286.359	22782	75275
7	532.094	22782	80148
13	866	22782	98551
23	1572	22782	113512

Table 5 Output at 180X180X180 for various set of images

F. Same Resolution with Different Set of Images (240X240X240)

2 Images **4 Images** **7 images** **13 Images** **23 Images**



No of Images	Time(sec)	Size(KB)	No of voxels
2	295.703	54001	75248
4	546.156	54001	123682
7	826.39	54001	133306
13	1404	54,001	161667
23	2262	54,001	186024

Table 6 Output at 240X240X240 for various set of images

VII. CONCLUSION AND FUTURE WORK

This research paper discussed Voxel Coloring. It derives how the voxel coloring algorithm behaves on different resolutions and number of images. In reconstruction using photoconsistency, the consistency measure plays an important role. While we are using threshold based methods like original consistency check and adaptive consistency check, the value of threshold affects the final reconstruction a lot. For different values should be tested while working with threshold based consistency test. One more method histogram consistency check is used for detecting consistency of voxels which doesn’t depend on any threshold value.

Voxel coloring algorithm needs a lot of memory for the computation and the array of voxel, which is used in general voxel coloring algorithm sometimes not cover the sharpness of object. As the resolution of the voxels got increased the refinement increased but with a significant increase in the time of computation.

REFERENCE

[1] Y. Lu, J.Z. Zhang; Q.M.J. Wu, and Li. Ze-Nian, “A Survey of Motion-parallax-Based 3-D Reconstruction Algorithms,” IEEE Trans. SMC, Part C, Volume 34, Issue 4, pp. 532-548, Nov. 2004.

[2] D.D. Morris and T. Kanade, “Image-consistent Surface Triangulation,” In IEEE Conf. on Computer Vision and Pattern Recognition, 2000. Volume 1, pp 332-338, June 2000.

[3] W. Matusik, C. Buehler, S. J. Gortler, R. Raskar, and L. McMillan, “Image Based Visual Hulls,” In Proc. of ACM SIGGRAPH, 2000, pp. 369-374, 2000.

[4] A Laurentini, “The Visual Hull for Understanding Shapes from Contours: A Survey,” In Proceedings of Seventh International Symposium on Signal Processing and Its Applications, 2003. Volume 1, pp. 25-28, July 2003.

[5] L. Zhang, D.Wang, and A.Vincent, “An Adaptive Object-based Reconstruction of Intermediate Views from Stereoscopic Images,” In Proceedings of International Conference on Image Processing, 2001, Volume 3, pp. 923-926, 2001.

[6] Snow, P. Viola, and R. Zabih, “Exact Voxel Occupancy with Graph Cuts,” In Proceedings of IEEE Conf. Computer Vision and Pattern Recognition, pp. 345-353, June 2000.

[7] A.Y. Mulayim, U. Yilmaz, and V. Atalay, “Silhouette-based 3-D Model Reconstruction from Multiple Images,” IEEE Trans. SMC, Part B, Volume 33, Issue 4, pp. 582-591, Aug. 2003.

[8] T. Fromherz and M. Bichsel, “Shape from Contours as Initial Step in Shape form Multiple Cues,” In ISPRS Commission III Symposium on Spatial Information Form Digital Photogrammetry and Computer Vision, pp. 240–256, 1994.

[9] K. L. Grauman, “A Statistical Image Based Shape Model for Visual Hull Reconstruction and 3D Structure Inference,” M.S. Thesis, Massachusetts Institute of Technology, June 2003.

[10] S. M. Seitz and C. R. Dyer, “Photorealistic Scene Reconstruction by Voxel Coloring,” In Proceedings of the Computer Vision and Pattern Recognition Conference, pp. 1067-1073, 1997.

[11] S. M. Seitz and C. R. Dyer, “Photorealistic Scene Reconstruction by Voxel Coloring,” International Journal of Computer Vision, Volume 35, Issue 2, pp 151–173, June 1999.

[12] M. Pilu, “A Direct Method for Stereo Correspondence Based on Singular Value Decomposition,” In Proceedings of the Computer Vision and Pattern Recognition Conference, pp. 261–266, 1997.

[13] O. Faugeras and R. Keriven, “Variational Principles, Surface Evolution, PDE’s, Level Set Methods, and The Stereo Problem,” IEEE Trans. Image Processing, vol. 7, pp. 336–344, 1998.

[14] Culbertson, T. Malzbender, and G. Slabaugh, “Generalized Voxel Coloring,” In Proceedings of the ICCV Workshop, Vision Algorithms Theory and Practice, Springer-Verlag Lecture Notes in Computer Science 1883, pp 100-115, 1999.

[15] O. Ozun, U. Yilmaz, and V. Atalay, “Comparison of Photoconsistency Measures Used in Voxel Coloring,” Msc. Thesis, Computer Vision & Remote Sensing, Berlin University of Technology, Germany, 2002.

[16] O. Balan, “Voxel Carving and Coloring-constructing a 3D Model of An Object from 2D Images,” Computer Science Department Brown University, URL: www.cs.brown.edu/~alb/papers/balan03voxel.pdf.
[17] K. N. Kutulakos and S. M. Seitz, “A Theory of Shape by Space Carving,” International Journal of Computer Vision, Volume 38, Issue 3, pp. 199–218, 2000.
[18] M. Sonka, V. Hlavac, and R. Boyle, *Image Processing, Analysis, and Machine Vision*. Brooks and Cole Publishing, 1998.
[19] R. Mohr and B. Triggs, “Projective Geometry for Image Analysis,” A Tutorial given at ISPRS, Vienna, July 1996,
[20] Camera Calibration Toolbox for Matlab,
[21] J. Heikkilä and O. Silvén, “A Four-step Camera Calibration Procedure with Implicit Image Correction,” Infotech Oulu and Department of Electrical Engineering, University of Oulu, Finland
[22] G. Slabaugh, W. B. Culbertson, T. Malzbender, M. R. Stevens, and R. W. Schafer, “Methods for Volumetric Reconstruction of Visual Scenes,” International Journal of Computer Vision, Volume 57, Issue 3, pp. 179–199, 2003.
[23] M. R. Stevens, B. Culbertson, and T. Malzbender, “A Histogram-based Color Consistency Test for Voxel Coloring,” In Proceedings of International Conference on Pattern Recognition, Volume 4, pp. 118- 121, 2002.