Photorealistic 2D to 3D Reconstruction with Improved GVC using Histogram-based Consistency measures

Dr. Kalpana Yadav*, Ms Ankita, Niharika Sachdeva***, Megha Pahwa****, Shruti Choudhary*****

* (Asst. Prof, Dept. Information Technology and Engineering, Indira Gandhi Institute of Technology, GGSIPU,)
**(Asst. Prof, Dept. Information Technology and Engineering, Indira Gandhi Institute of Technology, GGSIPU,)
***(Department of Computer Science and Engineering, Indira Gandhi Institute of Technology, GGSIPU, Delhi,)
****(Department of Computer Science and Engineering, Indira Gandhi Institute of Technology, GGSIPU, Delhi,)
*****(Department of Computer Science and Engineering, Indira Gandhi Institute of Technology, GGSIPU, Delhi, India)

ABSTRACT:

The improved GVC Algorithm proposed aims at high quality of 2-D to 3-D reconstruction by incorporating additional check of edge detection and usage of Histogram- based consistency as a photo consistency measure. The improved GVC algorithm comprises of two-stage consistency check, where at the first stage the multi modal histogram consistency check is applied on each uncarved voxel to determine its color consistency. At the next stage, all the voxels that were declared inconsistent after the first stage are tested for edges and features. If matches are found between the sets of pixels of images on which voxels project to (also called as footprints), the voxel is declared consistent, and else it is carved. The final output is reconstructed 3-D object/scene that resembles the original image. The histogram consistency measure is suited for lambertian surfaces, i.e. the surfaces that reflect light equally are all directions. Histograms are well suited for characterising multi modal distributions of color and hence are applicable to footprints of voxel with multiple clusters of color. Due to this reason, this consistency test yields better reconstruction results than obtained from tests based on means and variances.

***Keywords*-consistency check, GVC, reconstruction, histogram intersection, voxel, 2-D images, 3-D object/scene**

I. INTRODUCTION

Reconstructing a 3D model from a set of images is a popular field of work in computer vision. Scene reconstruction involves building 3-D models of scenes, given several 2-D photographs of the scenes. It has its applications in the field of virtual reality, tele-presence, and visual navigation. 3-D scene reconstruction has been carried out for many years from now and many algorithms have been proposed so far.

The improved GVC algorithm [13] that has been proposed takes into the account histogram based consistency checks and edge detection. The Generalized Voxel Coloring of Culbertson and Malzbender [3], (GVC) generalises Space Carving so that all views can be considered simultaneously, rather than with separate passes. It removes the camera positioning restrictions that existed in the original voxel coloring algorithm.

Any reconstruction algorithm is based on certain rules that specify whether the surfaces or regions shown in the images should be represented in 3-D reconstruction or not. One of the rules is based on color consistency. This paper talks about histogram consistency check that is well suited for characterising multi modal distributions of color sets. This measure has its own advantages over other consistency checks such as the ones based on standard deviation, means and variances. This paper highlights those advantages and discusses this photo consistency check in detail. It proposes a two stage consistency check where at first stage various sets of pixels belonging to the different footprints of a particular voxel are compared for similarity and consistency. In case the consistency holds, voxel is kept in the 3-D model, else it is declared inconsistent. At the second stage, all the voxels that were declared inconsistent at stage one, are re-evaluated for edges and other features to test whether the regions that voxel projects onto contain edges or not. In case it does, voxel is declared consistent, else it is carved out. Finally, all the voxels that are consistent, are kept in the 3-D reconstructed object, others form a part of the background. This paper discusses the improved GVC 2- stage consistency check algorithm in detail and the role of histogram measure in detail.

The outline of this paper is organized as follows- Section 2 highlights the previous related work in the field of 2-D to 3-D reconstruction. It discusses some previous methods of reconstruction that used different consistency checks. Section 3 discusses the histogram method in detail. In section 4, the histogram consistency method is applied to the improved GVC algorithm which comprises of two stage consistency check. In Section 5, conclusion regarding the histogram consistency check in Improved GVC is drawn.

II. PREVIOUS RELATED WORK

Various consistency measures have been using for image based rendering algorithms in the past. When reconstructing a scene using a reconstruction algorithm, there are two key factors that affect the quality of the reconstructed model. The first is the visibility that is computed for the voxels. The second factor is the test that is used to judge the photo consistency of voxels. The consistency test just can be thought of as being too strict for declaring voxels that belong in the model to be inconsistent. Tests can also be too lenient, declaring voxels to be consistent when they do not belong in the model; this can lead to voxels that appear to float over a reconstructed scene. A single consistency test can simultaneously be both too strict and too lenient, creating holes in one part of a scene and floating voxels elsewhere.

In most space carving implementations there has been an implicit assumption that the pixel resolution is greater than the voxel resolution—that is, a voxel projects to a number of pixels in at least some of the images. We believe this is reasonable and expect the trend to continue because runtime grows faster with increasing voxel resolution than it does with increasing pixel resolution. We make the assumption in this section that the scenes being reconstructed are approximately Lambertian, and we use the RGB color space, except where noted.

Most of the color consistency checks require a threshold value. One of the earlier methods used was the method of standard deviation. Using standard deviation as a photo consistency measure is proposed by Seitz and Dyer [2] Given a number n, of pixels with values x^i, their mean color, μ, can be computed as,

$$\mu = \left(\frac{\sum_{i=0}^{n-1} x_{red}^i}{n}, \frac{\sum_{i=0}^{n-1} x_{green}^i}{n}, \frac{\sum_{i=0}^{n-1} x_{blue}^i}{n} \right), \qquad (1)$$

And color variance, of the pixels can be computed as

$$\sigma^2 = \frac{1}{n}\sum_{i=0}^{n-1} \left((x_{red}^i - \mu_{red})^2 + (x_{green}^i - \mu_{green})^2 + (x_{blue}^i - \mu_{blue})^2 \right). \qquad (2)$$

The standard deviation σ can then be computed as square root of the variance. If it is less than a threshold value, the voxel is declared as photo consistent.

The consistency measure based on standard deviation works well when the object's surface color is homogeneous. Otherwise, it easily diverges. In order to handle this problem, Slabaugh et al. [3] proposed a new photo consistency measure called adaptive threshold. If a voxel is on an edge or on a textured surface, then the variation of the extracted color values is higher. Larger thresholds should be used so that the photo consistency measure converges. Assume that a voxel v, which is on an edge or on a textured surface, is visible from p views, and the standard deviations of these sets are σ_1, σ_2, σ_3 ...σ_n. These standard deviations should be high, since v is on an edge or on a textured surface. The average of these standard deviations, which is given in Equation 3, should also be high. By using this observation, authors define a new photo consistency measure called adaptive threshold as given in Equation 4.

$$\overline{\sigma} = \frac{1}{p-1}\sum_{j=0}^{p-1} \sigma_j \qquad (3)$$

$$consistent(v) = \left\{ \begin{array}{ll} true, & \sigma < \tau_1 + \overline{\sigma}\tau_2 \\ false, & otherwise \end{array} \right\} \qquad (4)$$

This measure brings an important advantage over the measure based on standard deviation: The value of threshold is variable according to the place of the voxel. If the voxel is on the edge or textured surface, this situation can be detected with high standard deviation in each image, and a greater threshold can be used. Adaptive threshold measure is actually superset of the measure based on standard deviation. The need for 2 thresholds is its main disadvantage.

Photo consistency of a set can also be defined using Minkowsky distances, L1, L2 and L∞. Minkowsky distance between two points x and y in <k is given in Equation 5.

$$L_p(x, y) = \left(\sum_{i=0}^{k} |x_i - y_i| \right)^{\frac{1}{p}} \qquad (5)$$

ISSN: 2248-9622 www.ijera.com Vol. 2, Issue 3, May-Jun 2012, pp.2422-2429

Assume that a voxel v is visible from p views, and the color sets extracted from these images are also known. Every color entity in each of these color sets should be in a certain distance to the color entities of the other sets. Through this idea, photo consistency of v is defined as in Equation 6. The distance between two color sets is given in Equation 7.

$$consistent(v) = \begin{cases} true, & \forall_{i,j}, consistent_{i,j}(v) \\ false, & otherwise \end{cases} \quad (6)$$

$$consistent_{i,j}(v) = \begin{cases} true, & \forall_{c_l \in \pi_i, c_m \in \pi_j}, L_p(c_l, c_m) < \tau \\ false, & otherwise \end{cases} \quad (7)$$

The most important benefit of using Minkowsky distance [4] as a photo consistency measure is the following. During the photo consistency check, if the voxel is found to be inconsistent, there is no need to continue to check photo consistency of that voxel. That means, having found a pair of colors whose difference is greater or equal to the threshold, voxel cannot be photo consistent.

In the next section, histogram method is explained in detail.

III. HISTOGRAM-BASED CONSISTENCY CHECK

Reconstruction of a 3 D model is simplified by assuming that images are calibrated. This implies that given a point in a 3-D scene, it can be determined where the point will appear on photographs. The surfaces can be 3- dimensionally modeled using voxels or other primitives such as polygons. Voxels, which are 3-D modeling elements are analogous to 2-D pixels and can be thought of as tiny as cubes found on a regular grid. the reconstruction algorithms, based on color consistency, typically test a large number of voxels to build 3-D scene. If they are found to have consistent color in various photographs, they are used to create the reconstructed object, else they are carved out. The voxels that are in undefined state are simply not used until their color consistency is determined.

Many different color space schemes are used to encode pixel color. Many of these color spaces have multiple dimensions. For example, the RGB color space has three dimensions: red, green, blue, and each color is assigned a numerical array value based on the amount of red, green and blue that is contained in the color of the pixel. For example, a color that is, 33% red, 57% green and 10% blue could be assigned an array of (33, 57, and 10). Voxels are generally visible from a number of pixels in the images. Hence, if V is a voxel or some other modeling primitive and i is the image, then Sv.i can be called the visibility set of voxel in from image i.

Histogram method [5] is a method of measuring color consistency that obtains two dimensional images and subdivides the images into image partitions (where image partitions are pixels or set of pixels). Each image partition has a color. Each pixel in the set of pixels, that are visible from the voxel, is placed in one of the histogram subdivisions, also called as a bin. The histogram obtained for all the images are compared for similarity.

All the algorithms use thresholding techniques like standard deviation , Minkowsky distance, adaptive etc in order to determine whether a particular voxel is consistent or not. However determining threshold values experimentally is a tough and time consuming task. The main problem is that there is no optimal threshold: areas with little texture are reconstructed best with a low threshold, while areas that are highly textured or contain sharp edges need very high thresholds.

Non Overlapping Histograms

The approach based on Histograms does not require any tunable parameters. Instead of pooling the pixels from all the views of a given voxel, this method uses a series of paired tests. For any carving iteration, the set of pixels that project to a given voxel for a given image is defined as: Vti where V is the voxel, t is the iteration number and i is the image number from which the pixels have been projected. Hence, the consistency of the voxel is defined as a paired test:

Consist (Vti, Vtj), where $i \neq j$

The projections of the voxels onto the image are divided in bins resulting in three dimensional histogram for each projection. Bins are blocks of range of intensities for each of the R, G, and B components. For instance, if number of bins for each component i.e. R, G, B is 4, and the bins are identified as 0, 1, 2,3, the corresponding intensity ranges would be 0-63, 64-127, 128-191, 192-255. The Histogram plots the number of pixels for each tonal value. Most often, the space is divided into an appropriate number of ranges, often arranged as a regular grid. The color histogram may also be represented and displayed as a smooth function defined over the color space that approximates the pixel counts. A histogram of an image is produced first by discretization of the colors in the image into a number of bins, and counting the number of image pixels in each bin. For instance, the red color values are encoded as 0 for 0 to 31, 1 for 32 to 63 and so forth. Same is

ISSN: 2248-9622 www.ijera.com Vol. 2, Issue 3, May-Jun 2012, pp.2422-2429

done for green and blue color. So a sub-space, defined as bin, in the color space can be identified by triple number ranging between 0 and 7. Given below is TABLE 1, with random set of data which explains the concept of bins.

TABLE 1
8*8*8 i.e. 512 bins – 8 bins for each channel

Red	Green	Blue	Pixel Count
0	0	0	7414
0	0	1	230
0	0	2	0
0	0	3	0
0	1	0	8
0	1	1	372
0	1	2	88
0	1	3	0
-			
-			
7	7	7	5392

Instead of directly comparing the sets of pixels, we say a voxel is consistent if all of the histograms of all views of the voxel intersect:

$$\forall\, k, l \; Hist(\pi ik) \cap Hist(\pi il) \neq \varphi \quad (8)$$

Most digital image formats provide a very large number of possible colors for each pixel. For example, some RGB images provide 256 possible values for each of the three color channels for a total of more than 16 million possible combinations of array values for the color of each pixel. However, fewer bins, such as 512 bins, can be used. In order to go from a 16 million bin histogram to a 512 bin histogram, larger bins can be used. These larger bins, which can be called histogram partitions, encompass several histogram subdivisions into a single histogram partition and reduce the overall number of bins in a histogram.

One advantage that a smaller number of larger bins provide is compensations for slight variation detected in color. For example, due to shortcomings of camera.

In RGB color space, numbers between 0 and 255 can be used to specify the amount of red, green and blue in a color. Since each color could then be assigned one of over 16 million array values(256*256*256) , this would lead to having process histogram that have over 16 million bins. Hence each of R,G,B values can be grouped into larger bins. This can be done by grouping values into 8 larger bins that contain original values: 0-31, 32-63 and so forth.
The larger bins are then numbered from 0-7 which leads to 8 bins for each channel and hence possible combination of 8*8*8 bins.

To check the consistency using histogram method, we specify that the two histograms intersect if, for alteast one bin, both histograms have non-zero counts. In other words two histograms intersect if alteast one pixel in one of the sets has a very similar color to a pixel in other set. If a set contains a very small number of pixels, or no pixels at all, then histogram may not provide a very effective measure. This intersection concept is extended to more than two histograms. For more than 2 histograms, every pair of histograms should be consistent i.e. for every pair of histograms in the histogram set, the above intersection rule should hold.
The histogram intersection only needs to test which histogram bins are occupied. Hence, only one bit is required per bin, or 512 bits per histogram. Histogram intersection can be tested with AND operations on computer words. Using 32-bit words, only 16 AND instructions are needed to intersect two histograms.

Overlapping Histograms

The voxel color consistency test, as described above, has a deficiency: it treats color values that fall near bin boundaries differently than other colors. For simplicity, consider colors that have the same green and blue values and consider the behavior of just the red value. In our specific implementation, a pixel with red value of 15 falls in the middle of its bin. Another pixel with a red value in the range 15 to 31 will be counted in the same bin. If these two pixels are in two different visibility sets, they have intersecting histograms and are considered consistent.

On the other hand, two pixels with red values 31 and 32 are close in color space, yet they are counted in different bins. These two red values would not be sufficient to pass the consistency test. Since the choice of bin boundaries is somewhat arbitrary, the color consistency test should be insensitive to their exact placement. Furthermore, color cannot be measured exactly. For example, a camera will measure slightly different color values on successive exposures of the identical scene. In addition, lighting usually changes slightly over time, causing minor shifts in color. Hence, for a number of reasons, we would like the consistency test to ignore small differences in color. We corrected the problems just described by using overlapping bins. In effect, this blurs the bin boundaries.

In our specific case, we enlarged the bins to overlap adjacent bins by about 20 percent. A pixel with a color falling into multiple overlapping bins is counted in each such bin. A pixel falling into just one bin is counted in just that bin, as before. This makes the consistency test insensitive to bin boundaries and small inaccuracies in color measurement. To avoid inaccurate decisions due to pixel values falling on boundaries on the bins, the boundaries are blurred, which means that a pixel value which falls near a boundary is considered to be present in both color spaces. Since the color space is in 3D, there can be case where a pixel falls under multiple sub-spaces or just a single sub-space.

The advantage of histogram based photo consistency measure is that there is no need for a preset threshold. Furthermore, paired tests can be very efficient in some circumstances. For instance, if the voxel is found to be inconsistent for a pair, there is no need to test other pairs of views for photo consistency.

IV. Histogram-based Consistency Test in Improved GVC

GVC determines visibility as follows. First, every voxel is assigned a unique ID. Then, an item buffer is constructed for each image. An item buffer contains a voxel ID for every pixel in the corresponding image. While the item buffer is being computed, a distance is also stored for every pixel. Each pixel also stores the item buffer identifier. A voxel V is rendered to the item buffer as follows. Scan conversion is used to find all the pixels that V projects onto. If the distance from the camera to V is less than the distance stored for the pixel, then the pixel's stored distance and voxel ID are over-written with those of V. Thus, after a set of voxels have been rendered, each pixel will contain the ID of the closest voxel that projects onto it. This is exactly the visibility information we need. Once valid item buffers have been computed for the images, it is then possible to compute the set vis(V) of all pixels from which the voxel V is visible. Vis (V) is computed as follows. V is projected into each image. For every pixel P in the projection of V, if P's item buffer value equals V's ID, then P is added to vis(V). To check the color consistency of a voxel V, we apply a consistency function consist() to vis(V) or, in other words, we compute histo-consist(vis(V)).

Given that vis(V) contains set of all the pixels from various images that are visible from voxel V, pixels belonging to the same item buffer, i.e. Pixels having the same item buffer identifier are added to one of the partitions/ bins of one histogram. This way there are as many histograms as the number of images involved in color consistency check of voxel. For each footprint where voxel projects, there exists one histogram. The histograms are compared for similarity as described in section 3. If each pairs of histograms intersect, the voxel is declared consistent, if not then the voxel is passed for further evaluation to stage 2 consistency check.

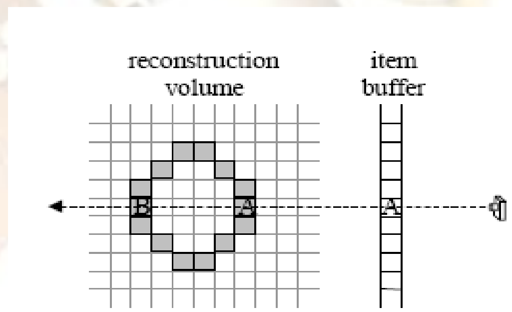

Figure 1 Item buffers used in GVC

Figure 2, shows flowchart representing Histogram Consistency Check on a Voxel. As every voxel is scanned one by one, this check is applied to it in order to reach to a decision whether the voxel should be a part of the 3- D object/ scene reconstructed or should be carved out.

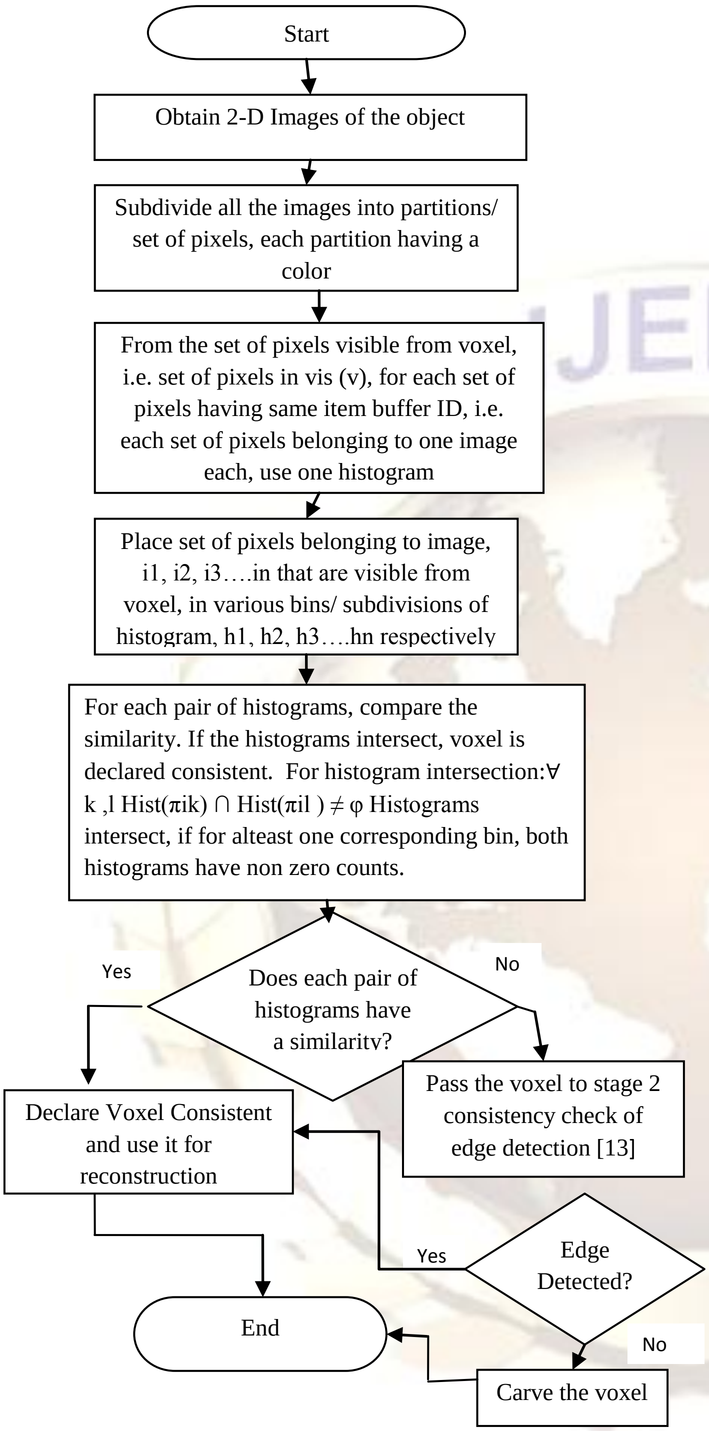

Figure 2 Flowchart representing Histogram Consistency Check on a Voxel

The pseudo code for the algorithm is as follows:

```
initialize
SVL
loop {
for all images
compute item buffer
accumulate color statistics into SVL voxels
for every voxel V in SVL {
if (V is histo-inconsistent) {

carve V (remove V from SVL){

add uncarved neighbors of V to SVL
}
}
if (no voxels carved)
done
}
```

Since carving a voxel changes the visibility of the remaining uncarved voxels, and since we use item buffers to maintain visibility information, the item buffers need to be updated periodically. GVC does this by recomputing the item buffers from scratch. Since this is time consuming, we allow GVC to carve many voxels between updates.

As a result, the item buffers are out-of-date much of the time and the computed set vis(V) is only guaranteed to be a subset of all the pixels from which a voxel V is visible. However, since carving is conservative, no voxels will be carved that shouldn't be. During the final iteration of GVC, no carving occurs so the visibility information stays up-to-date. Every voxel is checked for color consistency on the final iteration so it follows that the final model is color-consistent.

As carving progresses, each voxel is in one of three categories:

– it has been found to be inconsistent and has been carved;
– it is on the surface of the set of uncarved voxels and has been found to be consistent whenever it has been evaluated; or
– it is surrounded by uncarved voxels, so it is visible from no images and its consistency is undefined.

We use an array of bits, one per voxel, to record which voxels have been carved. This data structure is called carved in the pseudo-code and is initially set to false for every voxel. We maintain a data structure called the surface voxel list (SVL) to identify the second category of voxels. The SVL is initialized to the set of voxels that are not surrounded by other voxels. The item buffers are computed by rendering all the voxels on the SVL into them. We call voxels in the third category interior voxels. Though interior voxels are uncarved, they do not need to be rendered into the item buffers because they are not visible from any images. When a voxel is carved, adjacent interior voxels become surface voxels and are added to the SVL. To avoid adding a voxel to the SVL more than once, we need a rapid means of determining if the voxel is already on the SVL; we maintain a hash table for this purpose. The pseudo code for the same has been described below.

```
initialize SVL
for every voxel V
carved(V) = false
loop {
visibilityChanged = false
compute item buffers by rendering voxels on SVL
for every voxel V in SVL {
compute vis(V)
if (histo-consist(vis(V)) = false) {
visibilityChanged = true
carved(V) = true
remove V from SVL
for all voxels N that are adjacent to V
if (carved(N) = false and N not in SVL)
add N to SVL
}
}
if (visibilityChanged = false) {
save voxel space
quit
}
}
```

Figure 3, represents histograms for a surface voxel and for an interior voxel. A surface voxel usually has a color whose frequency is much higher than an interior voxel. Figure 4, shows representation of voxels in space. The 3 D space of object is subdivided into cube elements called voxels. To reconstruct the 3 D object using its 2- D images from various angles, voxels are scanned one by one and checked for the consistency. If the voxels are consistent, they are kept in the model and carved otherwise.

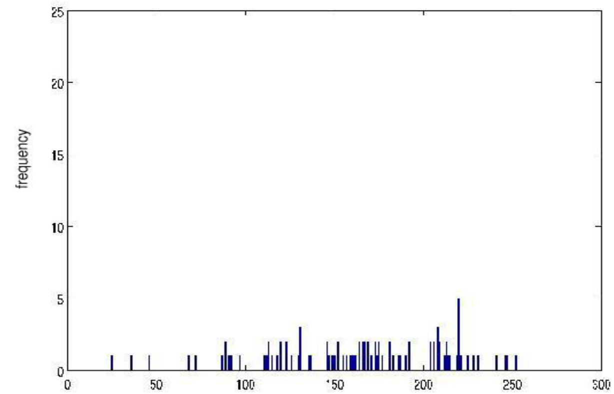

Figure 3 (a) An interior voxel

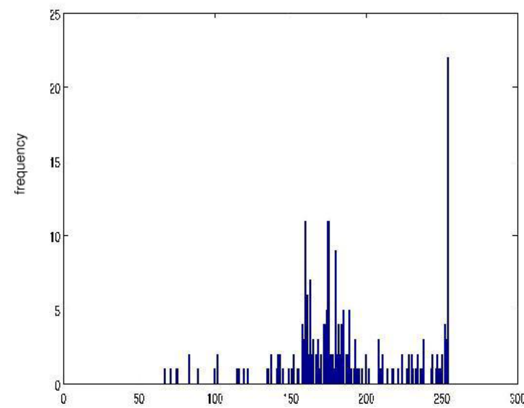

Figure 3 (b) A surface voxel

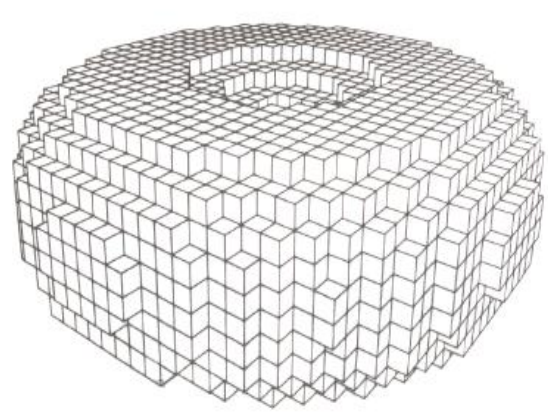

Figure 4. Example of voxel representation [14]

Fig. 5, shows application of histogram based consistency check using improved GVC. One of the photographed input images is shown in figure 5 (a). Figure 5 (b) shows reconstructed image using histogram consistency check with GVC.

Figure 5 (a) Photographed image

Figure 5 (b) Reconstructed Image using histogram consistency check

V. COMPLEXITY ANALYSIS

The number of comparisons in histogram consistency check during the reconstruction process is square of number of input images that are visible from the voxel under test. Hence the complexity of consistency test is of the order O (I2), where I is the number of input images visible from the voxel. As usually the number of images I is on an average 2-3, the number is histogram comparisons is manageable.

VI. CONCLUSION

Histograms are well suited for characterising multi modal distributions of color. For example, when multiple clusters are present in a set of colors, histograms doesn't blend the colors to obtain average color that may be close to any color in the set. In case there is an edge of red color and another of blue in the same footprint, unlike in methods using a variance value, the histogram based method would correctly record that some of the pixels are red in color and some of the pixels are blue, whereas in variance methods the mean is evaluated to be purple which is not consistent with the actual colors.

The histogram test avoids time-consuming experimentation because it is relatively insensitive to its parameter settings. This test does not require parameter tuning and can robustly reconstruct uniform and highly textured scenes. Given that our algorithm is a two stage consistency check algorithm, it checks for histogram consistency at stage one and edges at stage two, it doesn't let the edges carve out.

REFERENCES

[1] K Susheel Kumar, Vijay Bhaskar Semwal, and Shitala Prasad, "Generating 3D Model Using 2D Images of an Object" in International Journal of Engineering Science and Technology, 2011. Vol 3. pp 406-415

[2] 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

[3] 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.

[4] O. Ozun, U. Yilmaz and V. Atalay, "Comparison of Photo consistency Measures Used in Voxel Coloring," Msc. Thesis, Computer Vision & Remote Sensing, Berlin University of Technology, Germany, 2002

[5] 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

[6] O. Batchelor, R. Mukundan, R. Green "Incremental Voxel Coloring by Ray Traversal" in Computer Graphics, Imaging and Visualization, 2006 International Conference on Issue Date : 26-28 July 2006 On page(s): 396 - 401

[7] M. Heath, S. Sarkar, T. Sanocki and K. Bowyer, "Comparison of edge detectors: a methodology and initial study" in Computer Vision and Pattern Recognition, 1996. Proceedings CVPR '96, 1996 IEEE Computer Society Conference on Issue Date: 18-20 Jun 1996 On page(s): 143 – 14

[8] V. Chhabra, 2001. Reconstructing specular objects with image based rendering using color caching. Master's thesis, Worcester Polytechnic Institute

[9] Dianyong Zhang, Zhenjiang Miao; "Photorealistic 3d Volumetric Model Reconstruction By Voxel

Coloring" in Paparoditis N., Pierrot-Deseilligny M., Mallet C., Tournaire O. (Eds), IAPRS, Vol. XXXVIII, Part 3B – Saint-Mandé, France, September 1-3, 2010

[10] A. Broadhurst, R. Cipolla; "A Statistical Consistency Check for the Space Carving Algorithm" Dept. of Engineering, Univ. of Cambridge, Cambridge, CB2 1PZ

[11] Frank Porter, " Testing consistency of two histograms" , California Institute of Technology, Lauritsen Laboratory for High Energy Physics, March 7, 2008

[12] Patrick de Oude, "Two color consistency checks for voxel coloring", June 24, 2004

[13] Dr. Kalpana Yadav , Ms. Ankita, Megha Pahwa, Niharika Sachdeva, Shruti Choudhary, "Photorealistic 2D to 3D Reconstruction using Improved GVC-LDI" in International Journal of Computer Technology and Applications, Vol. 3, Issue 2, April,2012.

[14] J.D. Foley, A. van Dam, S.K. Feiner and J.F. Hughes. Computer Graphics: Principles and Practice. Addison-Wesly Publishing Co, section 12.6, 1990.