**An Experimental Study of 3D Models Generated with Simplified Voxel Coloring**

**Robert Love – CAP5416 – Fall 2009**

**Abstract**

An implementation of voxel coloring is used to build a 3-dimensional model of an object from a set of images taken at arbitrary locations with a commercially available compact camera. Several objects are examined to determine the effectiveness of the algorithm under differing experimental conditions.

**Background**

Voxel Coloring (VC)1,2, Generalized Voxel Coloring (GVC)3, and Space Carving4 are established techniques used for volumetric scene reconstruction from a set of images. All of these algorithms rely on the assumption that points on Lambertian surfaces project similar colors to any images from which they are visible. In addition to basic camera calibration, these methods require that a calibration grid be present in all images of the object to be modeled. A set of voxels are generated in 3D space and projected onto each image. The color of any given voxel is determined from the location of the voxel as projected on each image. Any voxels which are not color-consistent are carved from the final model. Voxels which are color-consistent are rendered according to the colors of the voxels as projected on the images.

 Determining (1) voxel visibility, (2) color-consistency, and (3) final color have received a great deal of research attention. Voxel based algorithms rely on the ability to identify from which image a voxel is visible to reduce computational costs. Ray tracing5, view morphing, light fields, lumigraph, stereo with multiple variations, and variational methods within a level set formulation have all been attempted to address voxel visibility.3 Issues due to object occlusion may be addressed to some extent with these techniques, but in general remains a challenge. Color consistency has been addressed using the variance, the L-1 norm, a histogram based method, and a mean charts color consistency method. Among these, only the mean charts approach does not require the user to specify a threshold.6 The result of the color consistency method generally determines the final color of the voxel.

Constraints on camera locations are not desirable for a realistic system. VC constrains the camera locations to simplify the visibility computation by requiring that the cameras are placed in such a way that the voxels can be visited on a single scan in front to back order relative to every camera. This means that the camera cannot surround the scene, constraining how well the object can be reconstructed. Space carving and GVC do not restrict the camera locations and ensure that the visibility of a voxel only increases as the algorithm runs, thus ensuring a conservative carving.

The algorithm presented in this project assumes that all voxels are visible from any image and assigns the color of the voxel to be the mean color of the voxel center across all images. This is not ideal and still limited in the case of multiple or occluding objects. However, it does not require any constraints on camera location. In addition, if the number of images is approximately balanced around the image, relatively good models may be generated by setting a threshold on the standard deviation of the RGB value of a voxel across all images without requiring the complexities of space carving or GVC. The algorithm presented also uses a voxel filter to threshold those voxels with very few neighbors and decreases the time required to render the object by only rendering the exterior surfaces of the object.

**Hardware Setup/Image Acquisition**

A Canon SD800 IS compact digital camera with resolution of 3072 x 2304 pixels is used to obtain images. A calibration grid under various orientations and 4 models (Winged Victory statue, Mickey Mouse statue, Gatorade bottle, Cynthia Goddard) are photographed without flash on different days (therefore different lighting) with 1 of 2 calibration grids.

**Camera Calibration (Intrinsic)**

Images of a calibration grid with 7x9 squares of size 30x30mm in various orientations are acquired in the .tiff format in various orientations and distances from the camera. Using .tiff files ensures that there will be no degradation in image quality throughout the process. Regions of 12x12 pixels are searched around the point clicked on the image to identify the location of the outside 4 vertices of the calibration grid (key points). The intrinsic parameters for the webcam are calculated with the “Camera Calibration Toolbox for Matlab” tool assuming a fixed focal length.8 The intrinsic parameters and associated uncertainties are shown below. 18 of the 20 images obtained are used in the final calibration.



**Figure 1. Grid Locations for Intrinsic Calibration with Respect to Camera Location (left), Predicted Values of Calibration Grid Points (Right)**

|  |
| --- |
| C:\Users\Robert\Desktop\voxel\cameracalparameters.jpg |

**Table 1. Intrinsic Calibration Results for SD800 Camera.**

**Image Acquisition and Calculation of Extrinsic Parameters**

Images are obtained taking care to ensure the 4 farthest points of the calibration grid (key points) and the full object may be seen in the image. The recommended minimum image set (9 images) is 0˚, 90˚, 180˚, 270˚ on level with model, 45˚, 135˚, 225˚, 315˚ just above model, and 1 image directly overhead of model as shown in Figure 2. The calibration toolbox is used to obtain the extrinsic parameters for each image by clicking the 4 farthest corners of the calibration grid. Errors for the extrinsic calibration are generally higher than those for the intrinsic calibration especially if the grid was near the edge of the image. An example reference frame from the extrinsic calibration is shown in Figure 2. The 3D points are projected on the image to test the calibration. The same algorithm uses the translation vector and rotation vector with the Rodrigues formula instead of the rotation matrix to project the voxel points from their locations in the world frame to locations in the image frame as displayed in Figure 2.

 

**Figure 2.** **Recommended Camera Locations for Imaging Model (left), Reference frame, Grid Points Projected from Extrinsic Calibration (middle) and Voxel Centers and Vertices in Image Frame (right)**

|  |
| --- |
| C:\Users\Robert\Desktop\voxel\cameracalparameters.jpg |

**Table 2. Extrinsic Calibration Results for Image Shown in Figure 2.**

**Algorithm Description**

The pseudo code for the voxel coloring algorithm as implemented in Matlab is shown below. The threshold for the standard deviation is imposed with the equation: μi=all images(σRi2+σGi2+σBi2)>threshold for all voxels. The algorithm determines the number of voxel neighbors (26 possible) and eliminates those which have less than 2 neighbors. It also specifies a floor threshold to eliminate the calibration grid from the model. The final output consists of only the exterior voxel faces, halving the render time.

|  |
| --- |
| **voxel pseudocode compviz.png** |

**Table 3. Pseudo-code for Voxel Coloring Algorithm.**

**Algorithm Analysis**

Both time and memory constraints are important considerations for the algorithm. Take n to be the number of voxels distributed initially and m to be the number of images used to generate the final 3D model. Since no visibility constraints are imposed, at present the voxel coloring is performed in a naïve manner where the final result has an asymptotic complexity of O(n^3m) in time and O(n^3) in space.

**Voxel Coloring**

A “standard” statue for voxel coloring of Mickey Mouse is examined to consider the effectiveness of the basic voxel coloring algorithm. The statue is fairly small compared to the calibration grid with large colored regions. The results shown in Figure 3 demonstrate how concave regions are difficult to model.



**Figure 3. Voxel Coloring for Mickey Statue, 20x20x20, 8x8x8mm Voxels, STD Threshold =0.2, 0.18.**

The voxel coloring results for the victory statue are shown below in Figure 4. The statue was chosen to examine the effects of little variations in color with large concave regions. Since no visibility constraints are imposed, the model is visible from any angle, although the color of a voxel is not guaranteed to be perfectly accurate. The lower resolution in space imposed by the cost of evaluating and rendering all voxels within the structures is seen to affect the quality of the reconstruction. One of wings of the victory statue and part of her legs are not generated fully, but the outline is clearly visible.



**Figure 4. Results of Voxel Coloring for 4 Views of the Victory Statue, 4x4x4mm Voxels, Neighbor Threshold=2, STD Threshold=0.095.**

The results for the Gatorade bottle at various resolutions are shown below. The smoothing result of not imposing a visibility constraint across all images and specifying voxel color as the average color across all images is evident as the letter G and lightning symbol are clearly present, but not nearly photo-realistic even in the highest resolution reconstruction. The lowest resolution construction struggles to obtain the full bottle while the highest resolution generates significantly large unwanted areas around the top of the bottle, demonstrating how the thresholds must be altered for each resolution.



**Figure 5. Results of Voxel Coloring for 3 Resolutions of the Gatorade Bottle: 20x20x20mm, 10x10x10mm, and 3x3x3mm Voxels, Neighbor Threshold=2, STD Threshold=0.2.**

Attempting to model large objects or people presents difficulties for voxel coloring since the smaller calibration grid can no longer be used. A calibration grid made of a blanket with 2 x 2in checkers is used. The voxel projections are prone to distortions since the grid is not perfectly regular or flat and errors are accentuated by the larger scale. Vibration may also be an issue when modeling people. Such large errors are present that only 9 of 20 images were usable. The final result is seen in Figure 6 below. Cynthia’s outline is visible, but the threshold needs to be better adjusted for future attempts. It is very expensive to obtain sufficient resolution at this large scale.

****

**Figure 6. Results of Voxel Coloring of Cynthia Goddard, 50x50x50mm Voxels, STD Threshold=0.135**

**Discussion of Experimental Variations**

A general guide is constructed to make researchers aware of pitfalls in the voxel coloring process. Variables that can substantially affect the results are described below based on the given stage where the variable is determined. The stages include hardware setup (HS), object selection (OS), calibration procedure (CP) and algorithm procedure (AP). The examples where this effect is seen are listed.

|  |  |  |
| --- | --- | --- |
| **Variable (Stage)** | **Example** | **Effect** |
| Image Resolution (HS) | All | Higher resolution makes voxel coloring tolerant of errors in calibration procedure and increases the accuracy of the voxel color determination. |
| Model Lighting/ Coloring (HS) | All | Objects should be lit from multiple directions if possible. Care must be taken for the camera not to inconsistently shade the object by coming between the light source and object. A large range of colors is helpful. |
| Image Distance from Model/ Model Location in Photo (HS) | Cynthia | Must ensure key points may be seen. Distortion increases towards the edges of the image. Therefore the camera should be far enough from the model to keep the model in the center of the image as much as possible. Note: camera orientation makes little difference. |
| # of Images (HS) | Mickey | More images, especially if balanced around model, improve result |
| Concave Region (OS) | Victory, Mickey | Concave regions in the object mandate that more voxels are used and result in more noise in the final result since voxels that should have been carved were not. Many concave regions degrades the final result. |
| Color Variation (OS) | Victory, Mickey, Gatorade | Setting voxel RGB equal to the mean value for all images allows models with small variations in color to be analyzed, generally large color variations are beneficial to distinguish the model from background. |
| Calibration Grid Size (CP) | Cynthia | Key points must be visible in all pictures. A larger calibration grid is helpful, but may accentuate projection errors. |
| Grid Integrity(CP) | Cynthia | Distorted calibration grid made finding results very difficult |
| VoxResolution(AP) | Gatorade | Increasing substantially improves results but also runtime |
| Vox Color (AP) | All | Mean color is a rough estimate, RGB histogram selection saturates colors but may be inaccurate especially for fewer images |
| Threshold (AP) | Gatorade | Must be changed for given Voxel Resolution, Better metrics available |

**Conclusions and Future Work**

The simplified version of voxel coloring presented provided results of reasonable accuracy despite the many assumptions made. The final output surfaces are not yet suitable for rapid prototyping from a pre-existing object, but it is hoped that several modifications may enable the results to achieve this stage of development soon. A visibility detector will be implemented to examine only those voxels which may be considered visible for any particular image and substantially decrease the run time of the final algorithm. A dataset of images with lower resolution would decrease runtime. A background with high standard deviation of RGB colors will also be constructed for future work.

**Appendices**

**Step by Step Users Guide**

A step by step “user’s manual” to the code posted online is provided below.

1. Print calibration grid from [www.vision.caltech.edu/bouguetj/calib\_doc/htmls/pattern.pdf](http://www.vision.caltech.edu/bouguetj/calib_doc/htmls/pattern.pdf). Affix the grid to a flat surface and take pictures of grid in various orientations while ensuring that the 4 far corners of same grid are visible in all images and that grid is flat.
2. Download and install the Matlab Camera Calibration Toolbox from: <http://www.vision.caltech.edu/bouguetj/calib_doc/>
3. Get Intrinsic Parameters with Camera Calibration Toolbox (Calibrate), select 4 points per image
4. Take pictures of object on grid from various perspectives, ensure same far corners of grid are visible in all images
5. Get Extrinsic Parameters with camera calibration toolbox, selecting the same 4 points per image in order consistently across all captured images.
6. Download Voxel zip file available at: <http://wordpress.ornithopters.com/about>
7. Determine approximate space where model is present by running only projections onto image, enter range to initialize the voxel space of interest
8. Decide on appropriate voxel scale, carving metric and threshold. Recommended to start: 15 x 15 x 15 mm voxel space (specify vdim based on size of you object), carving based on mean standard deviation across all normalized RGB values with threshold set to 0.15.
9. Run code, adjust voxel dimension and threshold as needed

**References**

1. Seitz, S., Dyer, C. R. “Photorealistic Scene Reconstruction by Voxel Coloring”, Proceedings of Computer Vision and Pattern Recognition Conference, 1997, pp. 1067-1073.
2. Seitz, S. M., Dyer, C. R., Photorealistic Scene Reconstruction by Voxel Coloring, International Journal of Computer Vision, v. 35, n. 2, p. 151-173, 1999.
3. Culbertson, W. B., Malzbender, T., Slabaugh, G., Generalized Voxel Coloring, Vision Algorithms Theory and Practice Worshop: 7th International Conference on Computer Vision, 1999.
4. K. N. Kutulakos and S. M. Seitz, “What Do N Photographs Tell Us about 3D Shape?”, TR680, Computer Science Dept. U. Rochester, January 1998.
5. Batchelor, O., Mukundan, R., Green, R., Incremental Voxel Colouring by Ray Traversal, Proceedings Computer Graphics, Imaging and Visualisation: Techniques and Applications, p. 396-401, 2006.
6. Adipranata, R., Yun, T. S., 3D Object Reconstruction Using Voxel Coloring with Mean Charts Color Consistency Checking, 3rd IASTED International Conference on Advances in Computer Science and Technology, ACST 2007, p. 206-211.
7. Montgomery, K., 3D Reconstruction,<http://biocomp.stanford.edu/3dreconstruction/>,2009.
8. Bouguet, J., “Camera Calibration Toolbox for Matlab” <http://www.vision.caltech.edu/bouguetj/calib_doc/>, 2008.