Embedded Voxel Colouring

Carlos Leung[1], Ben Appleton[1], and Changming Sun[2]

[1] Intelligent Real-Time Imaging and Sensing Group, ITEE,
The University of Queensland, Brisbane, QLD 4072, Australia
[cleung, appleton]@itee.uq.edu.au
[2] CSIRO Mathematical and Information Sciences,
Locked Bag 17, North Ryde, NSW 1670, Australia
changming.sun@csiro.au

Abstract. The reconstruction of a complex scene from multiple images is a fundamental problem in the field of computer vision. Volumetric methods have proven to be a strong alternative to traditional correspondence-based methods due to their flexible visibility models. In this paper we analyse existing methods for volumetric reconstruction and identify three key properties of voxel colouring algorithms: a *water-tight* surface model, a *monotonic* carving order, and *causality*. We present a new Voxel Colouring algorithm which embeds all reconstructions of a scene into a single output. While modelling exact visibility for arbitrary camera locations, Embedded Voxel Colouring removes the need for *a priori* threshold selection present in previous work. An efficient implementation is given along with results demonstrating the advantages of *posteriori* threshold selection.

1 Introduction

The reconstruction of a complex scene from multiple images is a fundamental problem in the field of computer vision. Given a set of images of a scene, the goal is to recover the 3D structure using knowledge of the camera geometry.

For dense 3D reconstruction there are two main approaches. Image correspondence techniques attempt to match primitives such as points, curves and regions between pairs of images to compute a disparity map [1,2,3]. While these techniques operate in 1D or 2D space, there is also a class of algorithms that operate in volumetric space. These algorithms project each image into a 3D volume, such that reconstruction of the scene can be computed directly from the resulting feature vectors. Unlike stereo methods, volumetric methods are able to reconstruct scenes from multiple views in a single step. It does not require multiple partial reconstructions then integration and can process scenes where few elements are visible from every camera.

Early work in reconstruction by volumetric modelling employed purely geometric knowledge [4]. One may additionally use photo-consistency to improve the 3D reconstruction. Under an appropriate reflectance model one may analyse the photo-consistency of each voxel in a scene to reconstruct the object's surface.

Voxel Colouring [5] is a volumetric algorithm that utilises photo-consistency. It begins with a volume of opaque voxels that encapsulates the scene to be reconstructed and converges towards the 3D object's surface, iteratively carving inconsistent voxels until all remaining voxels are photo-consistent. The major challenge for the carving algorithms is efficiently modelling occlusions. Voxel Colouring can only compute the 3D reconstruction correctly under the ordinal visibility constraint, which requires that the voxels closest to the cameras must be visited first. This near-to-far ordering relative to the camera allows voxels that are occluded to be evaluated for photo-consistency after the voxel that is occluding it has been visited. The disadvantage of the ordinal visibility constraint is that it cannot deal with arbitrary camera locations.

In recent years, algorithms have been proposed that allow for arbitrary camera placements while simultaneously modelling occlusions. Kutulakos and Seitz [6] proposed the Space Carving algorithm which performs multiple planar sweeps. Culbertson et al. [7] introduced the Generalised Voxel Colouring algorithm which maintains visibility information by storing the location of the nearest voxel to each pixel. Volumetric scene reconstruction algorithms have been surveyed in [8,9].

In this paper we analyse existing methods for volumetric reconstruction and present a new formulation that correctly models visibility for arbitrary camera locations. We describe the theory in Section 2 and describe the properties necessary for volumetric 3D reconstruction in Section 3. In Section 4, we present our Embedded Voxel Colouring algorithm, followed by the implementations and results of the algorithm in Sections 5 and 6.

2 Theory

The methodology of volumetric scene modelling is to compute the 3D reconstruction from the space described by the back-projection of each image into the scene. Under the pinhole camera model, the volume spanned by an image is the set of all projections through the camera centre that are within the bounds of the image. The volume represented by this image will be a frustum whose apex is the camera centre. During the process of reconstruction, the scene is a finite subset of the union of the pairwise intersections of cameras frustums. Each point in the reconstruction space will be classified for consistency by computing a metric from its set of visible projections.

A voxel is *visible* to a given pixel if the voxel projects to that pixel and there is a clear line of sight between them. We consider this relation to be reflexive, that is, if a voxel is visible to a pixel then the pixel is visible to that voxel. A voxel is *occluded* with respect to a given pixel if the voxel projects to that pixel but there is not a clear line of sight between them. This relation is also used reflexively.

Although the geometric relations between cameras are intrinsic to the projective volume, the lack of prior knowledge of the occlusions prevents the reconstruction from blindly using all of the available cameras. While Voxel Colouring as mentioned before is an effective algorithm for occlusion modelling, the ordinal visibility constraint is inescapable. This restriction on the sweep direction limits the placement of cameras. Given arbitrary views of a scene, Voxel Colouring may only utilise a subset.

Culbertson et al. [7] proposed an alternative to the ordinal visibility constraint for occlusion modelling. Instead of storing a boolean occlusion map for each pixel in each image, Generalised Voxel Colouring (GVC) stores all corresponding voxels for each pixel. Similar to the enhanced Space Carving algorithm of Kutulakos and Seitz [6], the goal of GVC was to model exact visibility with arbitrary camera placement.

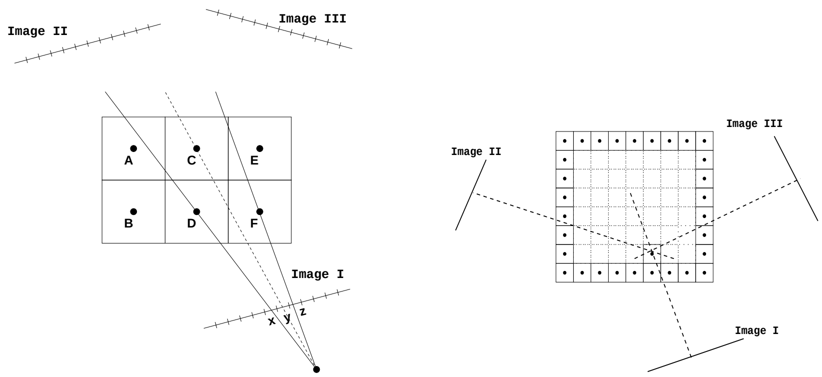
The GVC algorithm encompasses two different approaches to visibility modelling — GVC-IB and GVC-LDI. GVC-IB stores the identity of the closest voxel to each pixel, while GVC-LDI stores a list of the current surface voxels that project to that pixel. GVC-IB uses item buffering [10] which is similar to z-buffering in computer graphics. Querying the item buffer allows the visibility of any voxel to be immediately determined. The set of pixels visible to a voxel are then used to determine photo-consistency.

The disadvantage of GVC-IB is that the item buffer cannot be maintained efficiently. When a voxel is carved due to inconsistency, the algorithm must search for the next closest voxel. GVC-LDI overcomes this problem by trading off memory storage for computational efficiency. Using the concept of layered depth images [11], GVC-LDI maintains an ordered list of surface voxels that project to each pixel, resulting in a constant time search.

3 Water-Tight Surface Model

A general algorithm for Voxel Colouring must be implemented carefully so as not to depend on the ratio of voxel resolution to pixel resolution. For a high voxel resolution many voxels may project between pixel centres. This may confuse a naive rastering method, falsely classifying some voxels as invisible to the corresponding camera. For a low voxel resolution a single voxel may project to many pixels, all of which must account for the possibility of occlusion. Failure to do so may allows pixel rays to pass through occluding surfaces in the scene, falsely classifying some interior voxels as visible.

A simple example of the leakage of pixel rays through object surfaces is depicted in Figure 1(a). Consider the case where three views of a scene are available and the volumetric space has been divided into six cubic voxels. If voxels are projected as their centre points, voxels **D** and **F** do not project to pixel **y**. Consequently pixel **y** sees the interior voxel **C**. Thus in the computation of the photo-consistency of voxel **C**, projections from images I, II and III all contribute to the calculations, while in the true geometry, voxel **C** is only visible from images II and III. Incorrect visibility information causes errors in the photo-consistency computations, leading to voxels being incorrectly carved. Since the carving of a voxel affects the visibility of other voxels, these misclassifications



(a) (b)

Fig. 1. (a) A surface representation that is not water-tight leads to interior voxels becoming visible. (b) A leakage in projection can cause interior voxels to be carved, exposing other interior voxels which in turn could also be carved. A chain reaction occurs potentially leading to the carving of the entire object.

accumulate. Similar to puncturing a hole in a balloon, a leakage can lead to the removal of the entire object (Figure 1(b)).

A surface representation must be designed to avoid the false visibility problem. We refer to a surface representation as *water-tight* if rays cannot pass from the exterior to the interior without striking a surface voxel. Water-tight surfaces allow the correct modelling of occlusion for objects represented solely by their surface voxels. We require that voxel colouring algorithms use water-tight surface models to avoid the leakage problem. Water-tightness can be achieved if voxels are represented as volumes tessellating the scene space. For voxels represented as polyhedra, the complete polyhedra needs to be projected into an image to find all overlapping pixels.

An appropriate rastering method must also be chosen to avoid the false invisibility problem. Pixels must be represented as areas tessellating the image plane with any intersection between a projected voxel's area and a pixel's area considered as a potential occlusion. We assume that the first partial overlap of a pixel by a projected voxel occludes all further voxels, as depicted in Figure 2.

4 Embedded Voxel Colouring

We propose a voxel colouring algorithm that embeds the carvings for all possible consistency thresholds into one output.

The selection of an appropriate threshold for the consistency metric is critical to obtaining an accurate reconstruction. Too high a threshold will result in the inclusion of false surfaces and protrusions, while too low a threshold will remove real surfaces and introduce intrusions or etching. However it can be quite

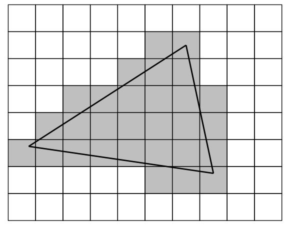


Fig. 2. A water-tight representation for the image is also important. Consider the bounds of a projection forming a triangle. The shaded pixels are all visible by the projected voxel.

difficult to select an appropriate threshold *a priori* to avoid these reconstruction artifacts. In practice a few simulations must be performed before selecting the final threshold for reconstruction [12]. As these simulations may take many minutes or even hours, this is naturally undesirable.

Ultimately the consistency threshold determines whether each voxel is included in the final scene. Each voxel should have a unique threshold at which it will be carved — it should not be possible that some lower consistency threshold would result in a voxel being left in the scene. We refer to this property as the *monotonicity* of a carving scheme. It is important to distinguish the monotonicity of carving from the monotonicity of metric as defined by [5].

When an inconsistent voxel is removed this causes other surface voxels which were previously occluded to become visible. Consequently their photo-consistency is reevaluated and their metric may rise above the consistency threshold. If so, they are added to a queue of voxels to be removed. Generalised Voxel Colouring uses a simple first-in-first-out queue to determine the order of removal [7]. However it is not clear that this maintains the monotonicity of carving. Lowering the consistency threshold may cause a voxel to become inconsistent earlier and consequently removed, altering the order in which voxels are removed. It is then possible that this will alter the shape to which the carving finally converges, violating the monotonicity of carving.

4.1 Monotonic Carving Order

Here we propose a simple scheme which guarantees the monotonicity of carving and allows the choice of consistency threshold *posteriori*. It produces a single output such that the reconstructions corresponding to all possible consistency thresholds are embedded as isosurfaces. In practice, such a scheme significantly reduces the effort required to select an appropriate consistency threshold.

Consider beginning with a very high consistency threshold such that no voxels will be removed from the scene. Incrementally reducing the threshold and allowing the carving to run to convergence will result in a monotonic set of carvings: each scene encapsulates all those of lower consistency threshold. Each voxel may then be labelled by the unique consistency threshold at which it was removed from the scene. These values may be stored and the subsequent volume be thresholded to obtain the scene corresponding to any desired consistency cutoff, allowing it to be determined after the algorithm has terminated.

To efficiently implement this scheme, we place all surface voxels in a priority queue ordered by their metric. Voxels of high metric are carved first while voxels of low metric are carved later. Voxels of equal metric should be carved simultaneously, otherwise their interaction due to changing visibility may cause one of them to decrease in metric. We explicitly track the global consistency threshold so that when a voxel is removed from the scene the consistency threshold is written to output. This threshold begins at infinity and decreases to the metric of the highest surface voxel whenever no further voxels may be carved at the current threshold. The algorithm has the same efficiency as a standard first-in-first-out scheme, and continues carving until no voxels are left. This produces a single output embedding all carvings of a scene.

4.2 Causality of Carving

As the reconstruction surfaces evolve, new surface voxels are created only as a result of a neighbouring surface voxel being removed. By the choice of a monotonic carving order, voxels are carved in decreasing order of consistency threshold. Therefore every voxel has a neighbour of equal or greater consistency threshold. By induction there exists a connected path of non-increasing consistencies from the volume boundary to any point in the scene. We may state this in its simplest form: there exist no local maxima in the embedded voxel colouring. We call this property *causality*.

Recall that the sequence of intermediate reconstructions are isosurfaces of the embedded voxel colouring. This leads to the following consequence of causality: while the surface of a carving may change topology as it evolves, splitting to represent separate objects, no new surfaces may be created.

5 Implementation and Algorithms

The algorithm is composed of four major blocks (Figure 3): the carving priority queue, the state volume, the surface voxel list, and the cameras. The carving priority queue is responsible for determining the order in which inconsistent voxels are removed from the scene. It issues commands to the state volume. The state volume maintains an efficient representation of the set of surface voxels. It is responsible for dynamically tracking the set of surface voxels. The surface voxel list stores the records for each surface voxel. This includes the list of pixels to which that voxel projects, the intermediate values for incremental metric computations, the distance of that voxel to each camera, and so on. The cameras are responsible for storing the image data and maintaining visibility lists. Visibility information is arbitrated between the surface voxel lists and the cameras.

We represent voxels as cubes tessellating the scene space and pixels as squares tessellating the image plane. This avoids the dual pitfalls of false invisibility and

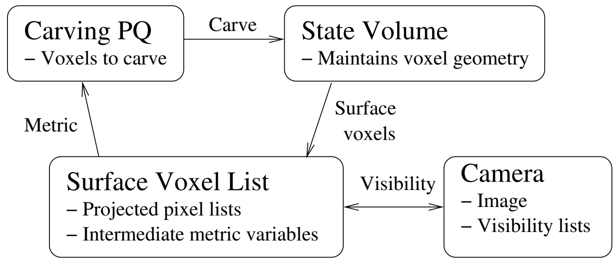


Fig. 3. A block diagram of the Embedded Voxel Colouring algorithm.

false visibility. Voxels are projected by their corners and the corresponding convex hull determines the projected polygon in the image plane. All pixels that intersect this polygon are considered for photo-consistency. When multiple pixels in one camera view the same voxel, their mean colour is used for consistency computations. Similar to Voxel Colouring we use variance as a consistency metric. Although this metric is not monotonic, it is widely accepted for its simplicity, statistical basis and efficient incremental computation [5].

Our algorithm is capable of correctly handling any ratio of image resolution to scene resolution. However, in order to accurately represent occlusions, we must ensure that each voxel projects to a sufficient number of pixels. If the image resolution is too low then unrelated voxels may appear to occlude one another. If the image resolution is too high then an excessive amount of memory is devoted to occlusion modelling. Empirically we find that each voxel should project to an area approximately double the size of a pixel in each camera.

We have proposed a method to compute a single output embedding all carvings of a scene. While in theory this may be used to compute the reconstructions for the full range of consistency thresholds, in practice, this is usually unnecessary. To reduce computation times, we initialise our input from a visual hull, and halt early when the consistency threshold drops significantly below the expected range.

6 Results

We present the results of Embedded Voxel Colouring on the "dinosaur" image sequence. The image set consists of thirty-six 720×576-pixel images of a dinosaur model spinning on a turn-table captured under a static light source. All tests were performed on a Sun UltraSPARC II 450MHz computer with 4GB of RAM.

Figure 4 (a, d) shows two images from the dinosaur image set and Figure 4 (b, c, e, f) shows examples of the different views rendered from the reconstructed dinosaur. There are a number of sources of modelling errors. Currently when a voxel is projected to a fraction of a pixel, this partial overlap is sufficient to claim that pixel as visible. While this ensures that the surface model is water-tight, the 3D reconstruction suffers from pixels which project to more than

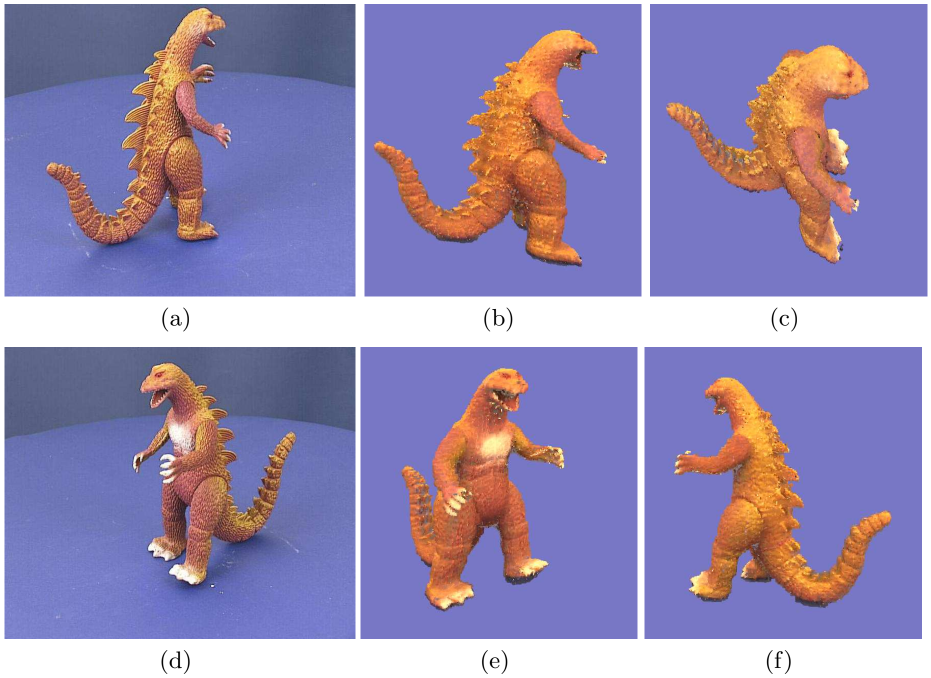


Fig. 4. Reconstruction of the dinosaur image sequence. (a, d) Selected images from the dinosaur image set. (b, c, e, f) New views generated from the 3D reconstruction.

one object in the scene, thus containing a mixture of colours. This effect can be observed in the 3D reconstruction where the colours of some voxels are mixed with the blue table colour. Furthermore we do not model shadowing effects. As the lights rotate with respect to the scene, voxels that change illumination are carved early.

Figure 5 shows the set of reconstructions embedded in one computation of the Embedded Voxel Colouring algorithm. The reconstruction for the choice of any threshold can be extracted from the embedded output, which is computed in a single sweep of the voxel volume. In previous work, the selection of the threshold determines the tradeoff between reconstruction accuracy and completeness. As a result, repeated simulations are often necessary in order to determine the optimal threshold. Embedded Voxel Colouring removes this need for repeated computations with the same speed and memory usage.

Due to memory limitations the largest reconstruction of the dinosaur was a $350\times350\times350$-voxel volume. Reconstructing the scene using all thirty-six full resolution images the computation of the embedded voxel colouring required 82 minutes. When the images were downsampled by a factor of two, sufficient for accurate occlusion modelling, the simulation required 53 minutes. Our computational results are an improvement upon previous work. The largest reconstruction demonstrated by GVC and GVC-LDI is a synthetic scene of resolution

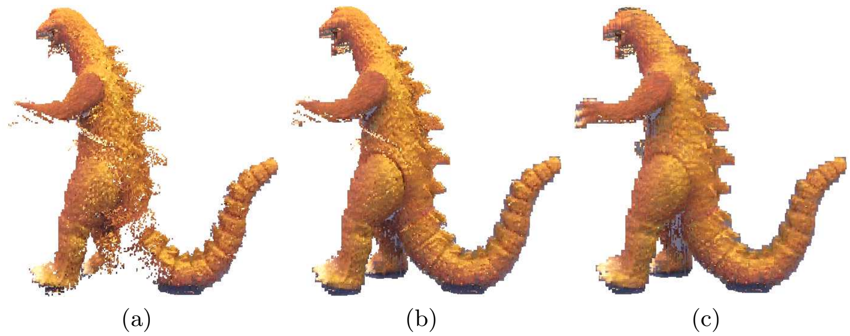


Fig. 5. The embedded voxel colouring contains the reconstructions for all consistency thresholds. Isosurfaces of the embedded output with (a) small threshold, (b) medium threshold, (c) large threshold.

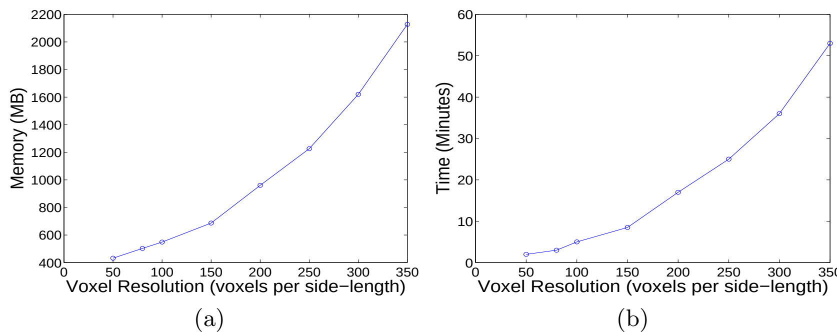


Fig. 6. Memory usage and computation time for different voxel resolutions under a constant pixel resolution

167×121×101 from seventeen 800×600-pixel images. For a similar computational time, we are able to reconstruct a scene more than 20 times as dense. The run-time statistics and memory usage for various voxel resolutions are presented in Figure 6. These results are based on reconstructions of the dinosaur using all thirty-six images downsampled by a factor of two.

7 Conclusion

We have presented a new Voxel Colouring algorithm which embeds all reconstructions of a scene in a single output. This is computed in a single sweep of the volumetric space.

In this paper we explored the properties necessary for volumetric 3D reconstruction under exact visibility and arbitrary camera placements. The importance of a water-tight surface representation was identified, and a Voxel Colouring algorithm that is consistent with the properties of water-tightness, monotonicity and causality was presented. The Embedded Voxel Colouring algorithm not only has the advantages of similar volumetric algorithms such as Generalised Voxel Colouring and enhanced Space Carving, it removes the need for *a priori* threshold selection. An efficient implementation has been given along with results demonstrating the advantages of *posteriori* threshold selection.

Acknowledgements

We would like to thank Jochen Wingbermühle of the University of Hannover for the dinosaur image set and Jiang Guang of the Chinese University of Hong Kong for providing us with the calibrations.

References

1. Fua, P.: From multiple stereo views to multiple 3D surfaces. International Journal of Computer Vision **24** (1997) 19–35
2. Roy, S., Cox, I.J.: A maximum-flow formulation of the N-camera correspondence problem. In: ICCV, Bombay, India (1998) 492–499
3. Sun, C.: Fast stereo matching using rectangular subregioning and 3D maximum-surface techniques. International Journal of Computer Vision **47** (2002) 99–117
4. Martin, W.N., Aggarwal, J.K.: Volumetric descriptions of objects from multiple views. IEEE Trans. on Pattern Analysis and Machine Intelligence **5** (1983) 150–158
5. Seitz, S.M., Dyer, C.R.: Photorealistic scene reconstruction by voxel coloring. In: CVPR, Puerto Rico (1997) 1067–1073
6. Kutulakos, K.N., Seitz, S.M.: A theory of shape by space carving. Technical Report TR692, Computer Science Dept., U. Rochester (1998)
7. Culbertson, B., Malzbender, T., Slabaugh, G.: Generalized voxel coloring. In: Vision Algorithms: Theory and Practice. Number 1883 in LNCS, Corfu, Greece, Springer-Verlag (1999) 100–115
8. Dyer, C.R.: 16. in Foundations of Image Understanding. In: Volumetric Scene Reconstruction From Multiple Views. Kluwer, Boston (2001) 469–489
9. Slabaugh, G., Culbertson, W.B., Malzbender, T., Schafer, R.: A survey of volumetric scene reconstruction methods from photographs. In: Volume Graphics 2001, Proc. of Joint IEEE TCVG and Eurographics Workshop, Stony Brook, New York, USA, Springer Computer Science (2001) 81–100
10. Weghorst, H., Hooper, G., Greenberg, D.P.: Improved computational methods for ray tracing. ACM Transactions on Graphics **3** (1984) 52–69
11. Shade, J.W., Gortler, S.J., He, L.W., Szeliski, R.: Layered depth images. Computer Graphics **32** (1998) 231–242
12. Stevens, M., Culbertson, B., Malzbender, T.: A histogram-based color consistency test for voxel coloring. In: International Conference on Pattern Recognition. Volume 4., Quebec, Canada (2002) 40118–40121