To Appear: International Conference on Computer Vision, 1999

Roxels: Responsibility Weighted 3D Volume Reconstruction

Jeremy S. De Bonet & Paul Viola
Learning & Vision Group
Artificial Intelligence Laboratory
Massachusetts Institute of Technology
545 Technology Square
Cambridge, MA 02139

EMAIL: jsd@ai.mit.edu & viola@ai.mit.edu
HOMEPAGE: http://www.ai.mit.edu/projects/lv

Abstract

This paper examines the problem of reconstructing a voxelized representation of 3D space from a series of images. An iterative algorithm is used to find the scene model which jointly explains all the observed images by determining which region of space is responsible for each of the observations. The current approach formulates the problem as one of optimization over estimates of these responsibilities. The process converges to a distribution of responsibility which accurately reflects the constraints provided by the observations, the positions and shape of both solid and transparent objects, and the uncertainty which remains. Reconstruction is robust, and gracefully represents regions of space in which there is little certainty about the exact structure due to limited, non-existent, or contradicting data. Rendered images of voxel spaces recovered from synthetic and real observation images are shown.

1 Introduction

Given information from very many cameras, one might hope that a completely veridical representation for the 3D scene could be computed. Such a representation would necessarily contain information about the shapes and locations of all objects, information about the colors and reflectance properties of each surface, and finally information about all light sources.

In this paper, we seek to compute a simpler model. One which represents the transparency and color of a voxelized representation of 3D space. While useful for a number of tasks, this representation ignores some of the more complex properties of image formation. For example, there is no attempt to measure or estimate the surface reflectance at each surface point, nor is there any attempt to estimate the scene illumination. Nevertheless, such a representation will contain information about the shapes and locations of objects. It is also contains information necessary to compute convincing synthetic views.

2 Previous Work

Their are two distinct classes of previous research on 3D volume reconstruction. Algorithms developed within the vision community typically assume that objects can be represented as completely opaque, and the effects both of transparency and aliasing can be ignored. In contrast, algorithms for medical imaging assume that volumes contain only semi-transparent tissue and the effects of occlusion can be ignored. In each case, these assumptions can be exploited in the design of special purpose volume reconstruction algorithms. However, even within their respective domains, these approximations are inaccurate. In natural imagery environments commonly contain transparent objects and because of data limitations, their exact locations can be uncertain; in medical imagery, bone and other solid tissue can cause occlusions.

The estimation of structure from natural imagery is a very diverse field which includes stereo, multibaseline stereo, and more general multiple camera approaches.[1] We will limit this review to closely related approaches.

The earliest approach which reconstructed a discrete or voxelated representation of 3D space was the cooperative stereo algorithm proposed by Marr

[1] Several reviews of stereo algorithms are available [1, 2].

and Poggio [5]. The Marr-Poggio approach was distinct in that it simultaneously represented and manipulated evidence for multiple disparities. This allowed for the initial consideration of several hypotheses which would eventually be pruned through subsequent competition.

In much the same spirit, recent work on stereo by Szeliski and Golland incorporates evidence for competing correspondences but also adds an explicit representation of partially transparent regions [8]. They note that even in the ideal case, where all the objects in the scene are completely opaque, a perfect voxelized representation requires transparency along the boundaries of objects where the voxels are only partially filled. In their approach to volume reconstruction, Szeliski and Golland use real valued transparencies to represent voxels which are partially occupied by opaque objects. An accurate model of partially occupied voxels is potentially useful for a wide range of scene properties including very fine structures, such as the hair or mesh and for the representation of semi-transparent materials such as colored glass.

A different approach for 3D reconstruction can be found in the computed tomography literature[2]. Solutions in this field typically ignore occlusion, as most materials are only partially opaque. The simplification allows reconstruction to be performed with efficient linear methods [7, 4, 9]. However, in the presence of opaque materials, this simplifying assumption leads to "ghosts" or "shadows" – false signals caused by the structures whose contributions should be occluded.

Recently, Seitz and Dyer have proposed an algorithm which computes a set of occupied voxels that is consistent with a large number of observed images [6]. Unlike Marr-Poggio, evidence for multiple correspondences is not explicitly represented, nor do multiple potential correspondences compete. Their approach, however, is distinguished by its efficiency, simplicity, and its explicit representation of occlusion.

The Seitz and Dyer algorithm makes a single pass through voxel space, first computing the visibility of each voxel and then its color. Their algorithm is based on a simple yet critical insight: each camera must agree on the color of an opaque voxel, but only when that voxel is *visible from that camera.* This approach yields fairly accurate 3D reconstructions, and the images rendered by this algorithm are quite impressive.

This paper presents a new approach for voxelized reconstruction called the Responsibility Weighted 3D Volume Reconstruction (Roxel) algorithm. The Roxel algorithm is an attempt to combine the best properties of all of the above approaches.

[2]This connection was pointed out to us by William Wells III

- The Roxel approach addresses two limitations of the Seitz and Dyer approach: i) the assumption that a pixel is either completely transparent or completely opaque; and ii) the definitive nature of the decision regarding this opacity which does not take into consideration alternative hypotheses which could better explain *all* the data simultaneously. In the Seitz and Dyer approach voxels can be incorrectly labelled opaque because of errors in calibration, image noise, and false correspondences.
- The Roxel approach is simpler and perhaps more efficient than the work of Szeliski and Golland.
- The Roxel approach can be used to reconstruct opaque objects, which cannot be accurately reconstructed with computed tomography algorithms.

Here we present a framework for volume reconstruction in which solid and transparent objects can be accurately represented. Furthermore, because the current approach formulates the problem as one of optimization over the the distribution of *partial* responsibility within a volume, uncertainty – due to lack of data, or perhaps contradictory data – can be captured as well.

3 A Unified Framework for 3D Voxel Models

To visualize a voxel space, one traces along a ray cast from each pixel to determine the sequence of voxels which are visible. The observed pixel intensity is a weighted combination of the colors along the ray and the weights are a function of the voxel transparencies.

The voxel space, $v(x, y, z)$, consists of a three dimensional array of colors $c(x, y, z)$ and opacities $\alpha(x, y, z)$. Observed in isolation, the observed color is a combination of the voxel's color and the color which lies behind:

$$c_{obs} = v \wedge c_b = \alpha * c + (1 - \alpha) * c_b \qquad (1)$$

An opaque voxel allows none of c_b to pass; a transparent voxel is entirely invisible. An arbitrary image of this volume can be computed:

$$I_k^s(u, v) = \quad v(< k, u, v, 0 >) \wedge v(< k, u, v, 1 > \wedge \ ... \\ ... \ \wedge v(< k, u, v, n >) \wedge c_{bg} \qquad (2)$$

where I^s denotes a synthesized image and c_{bg} the background color. $< k, u, v, i >$ is a ray casting operation which computes the voxel (x, y, z) which intersects a ray cast from pixel u, v in image k at depth

i. The compositing operator, $\wedge$, associates right to left. The values taken on by $< \cdot >$ are not necessarily integral, in this case bi-linear interpolation is used to determine values of $c(< \cdot >)$ and $\alpha(< \cdot >)$.

3.1 Direct Reconstruction Algorithms

Given the straightforward relationship between image obervations and voxel values one could directly search for $v(\cdot)$ such that

$$Cost(v) = \sum_j (I_j - I_j^s)^2$$

is minimized. While it may be theoretically possible to directly minimize this function,we do not attempt such a solution for two reasons: the predicted images are highly non-linear functions of the c and α, and there are a very large number of parameters. Instead the Roxel algorithm minimizes this function using an alternate decomposition.

Though this fact is somewhat hidden in the above notation, the observed pixel intensity is a weighted sum of the colors along the cast ray:

$$I_k(u,v) = \sum_i r_k(< k,u,v,i >) c(< k,u,v,i >)$$

where I_k is the kth image, and the weight

$$r_k(< k,u,v,i >) = \alpha(< k,u,v,i >) * \prod_{j<i} [1 - \alpha(< k,u,v,j >)] \quad (3)$$

and

$$r_k(< k,u,v,0 >) = \alpha(< k,u,v,0 >).$$

We will call these weights the *responsibility* of a voxel for the observation at a pixel. In order to simplify the notation in the remainder of this section the image pixels can be collected into a column vector, the colors into another column vector, and the responsibilities into a matrix with one row for each pixel and one column for each voxel: $I_k = R_k * C$. If the images are stacked into a single vector and the responsibilities stacked into a single matrix, the entire multi-camera reconstruction problem may expressed as: $I = R * C$. While the representation of I and C are reasonably straightforward, the size of R could present problems. Recall however, that R is incredibly sparse – the only non-zero responsibilities are those along rays cast from the pixels. Nevertheless, any direct approach for determining R and C is potentially very difficult.

3.2 Efficient Solutions for Direct Inversion

Although they do not formulate their work in this way, the Seitz and Dyer approach can be placed into the direct reconstruction framework. Their algorithm generates a binary responsibility matrix such that each row of R contains only a single non-zero entry, because in their model the intensity of each pixel is attributable to only a single voxel. The responsibility matrix is computed using the opacity heuristic mentioned above: the responsibility of a voxel is 1 if each of the cameras which can view the voxel agree on its intensity, and 0 otherwise. This matrix can be inverted trivially in order to find the voxel colors: the color for a voxel is the average of the colors observed in the pixels for which it is responsible.

Computed tomography (CT) can also be easily be expressed in this color / responsibility framework. The close relationship between 3D voxel reconstruction and computed tomography has been pointed out before[3]. In CT the responsibilities are fixed and have a simple form: the value of each pixel is the sum of color values along a cast ray. One approach for computing the voxel colors is to use *back-projection*: $\hat{C} = R^T I$ [4, 9]. This amounts to projecting the values of each pixel back out into the volume. While this is not quite correct, since $R^T \neq R^{-1}$, the algorithm is quite simple and the results are reasonable. A better approach, which produces images with sharper boundaries, computes $\hat{C} = (R^T R)^{-1} R^T I$ and is known as *filtered back-projection*. Because of the size of R, computing $(R^T R)^{-1}$ could potentially be very difficult. However, in the special case of CT scans acquired using regular geometry $(R^T R)^{-1}$ can be expressed as a convolution. As a result, filtered backprojection is quite efficient. Gering and Wells have directly applied filtered back-projection to 3D voxel reconstruction with some success [3]. The most salient drawback is that occluding properties of surfaces are never accounted for, and some of the "shadow" effects of standard CT reonstruction are still observed.

As we have seen, these two approaches solve an apparently intractible problem quite efficiently by making use of simplifying assumptions.

4 The Roxel Algorithm

The responsibility weighted 3D volume reconstruction (Roxel) algorithm is a multi-step procedure which alternates between estimation of the colors, estimation of responsibilities, and estimation of opacities. Given a set of images, I, and voxel responsibilities R, the voxel colors, C may be computed by inverting the

linear system (as in filtered backprojection). There is of course, a symmetric relation in which the colors and images can be used to compute responsibilities. These two steps can be combined into a multi-step reconstruction algorithm which gradually improves initial estimates for R and C. If implemented naively each of the steps in this process is prohibitively expensive. Simply inverting the linear system would require work which is proportional to the cube of the number of voxels. The Roxel reconstruction algorithm attempts to solve the transparent voxel coloring problem while preserving some of the efficiencies of the Seitz and Dyer algorithm and of filtered backprojection.

The Roxel algorithm is initialized with the simple linear responsibility matrix used in computed tomography. This corresponds to the notion that initially each voxel along a cast ray is equally responsible for that pixel.

In the first step of the algorithm agreement between multiple observations is used to gain an initial estimate of opacity; a voxel is likely to be responsible for an observation, and therefore opaque, if it could be responsible for multiple observations.

Step 1: Color Estimation

The color estimate for each voxel is the average over the pixels that can potentially observe it, weighted by the responsibility of that voxel for the color observed at each pixel:

$$\overline{C}(x,y,z) = \frac{\sum_k r_k(x,y,z) I_k(p_k(x,y,z))}{\sum_k r_k(x,y,z)},$$

where $p_k(x,y,z)$ projects a 3D voxel into the image plane of camera k. Note that this is a generalization of back projection. Since it is an estimate for the inverse of the linear system, it is most accurate when the responsibility matrix is sparse.

Step 2: Computation of Agreements

The view specific disagreement at a voxel is the squared difference between the voxel color estimate and the pixel color in image k:

$$d_k(< k,u,v,i >) = \left(I_k(u,v) - \overline{C}(< k,u,v,i >)\right)^2$$

The view specific agreement at each voxel is:

$$a_k(x,y,z) = e^{-\frac{1}{\sigma^2} d_k(x,y,z)}$$

where σ is a free parameter expressing the belief about the noise and calibration errors in the observations.

Using reasoning similar to that of Seitz and Dyer, a large value for agreement can lead to the conclusion that a voxel is opaque (since it is possible that the voxel "caused" the observation). A large value for agreement does not neccesarly imply opacity: agreement may be due to false correspondences between the observed images.

Step 3: Computation of Responsibilities

The next step in the Roxel algorithm is reminiscient of the Marr-Poggio competition among disparities: the agreements are normalized along observation rays so that they sum to one. This forces the voxels on the ray to jointly explain 100% of the observation. Voxels with large agreement dominate the sum and "win out" in the final competition for responsibility. Responsibility is an inherently view dependent quantity.

Agreement is normalized to form a set of responsibilities along each ray:

$$r_k(< k,u,v,i >) = \frac{a_k(< k,u,v,i >)}{\sum_j a_k(< k,u,v,j >)}$$

To understand the effect of these steps, consider the case where one voxel, $v(< k,u,v,i >)$, along the ray has near perfect agreement, while the other voxels along the ray do not agree well. In this case the responsibility for $I_k(u,v)$ will fall almost entirely on $v(< k,u,v,i >)$ (equivaliently only one of the entries in that row of R will be non-zero).

Step 4: Computation of Opacities

At this point the Roxel algorithm computes a set of view dependent opacities which are consistent with the view dependent responsibilities. Though the relationship between observed intensity and opacity is highly non-linear, there is a direct method for computing a set of opacities from a set of responsibilities:

$$\alpha_k(< k,u,v,i >) = \frac{r_k(< k,u,v,i >)}{1 - \sum_{j<i} r_k(< k,u,v,j >)}.$$

A globally consistent set of opacities is computed by the weighted average of the individual view estimates:

$$\overline{\alpha}(x,y,z) = \frac{\sum_k r_k(x,y,z)\alpha_k(x,y,z)}{\sum_j r_j(x,y,z)}.$$

Step 5: Re-estimation of responsibilities

The final step in the procedure computes a new set of responsibilites for each voxel using the aggregated

opacities:

$$r'_k(< k,u,v,i >) = \overline{\alpha}(< k,u,v,i >) \left[1 - \prod_{j<i} \overline{\alpha}(< k,u,v,j >) \right] \quad (4)$$

These responsibilities are then used in subsequent iterations of the process.

The entire Roxel procedure is repeated until the global opactiy estimate converges. At which point the global color $\overline{C}$ and transparency $\overline{\alpha}$ are extracted and combined to form the final semi-transparent voxelated space.

4.1 Algorithm Discussion

The Roxel algorithm progresses from an initial estimate of the volume as entirely trasparent, toward a state in which much of the volume is empty, and the observations are explained by a collection of semi-transparent and opaque structures.

In the initial phases of reconstruction, occlusions cannot be accurately determined and each image can potentially observe each voxel. However, because of occlusion, it is typically the case that only a few images actually observe a given voxel. As a result of this, intial estimates of voxel color agreement are inaccurate because they rely on some observations which are, in reality, occluded. Nevertheless, some voxels are sufficiently visible so that initial agreement estimates are reasonably accurate. Based on this information, the opacity of some voxels will be realized. The occlusions caused by these voxels are then incorporated into future color agreement estimates. As the algorithm progresses, images which do not observe a voxel because of occlusion are gradually phased out of color agreement estimates. In the final iterations accurate information about occlusion is available and the calculation of colors becomes more accurate since information from occluded viewpoints is disregarded completely.

One important aspect of the Roxel algorithm is that it equates transparency with uncertainty. This equivalence is justifiable: a voxel which contains completely opaque material with less than complete certainty is equivalent to a voxel which contains semitransparent material in that the expected observation of each is identical.

Suppose reconstruction is performed from a set of observed images which are completely white. From this information, we can be certain that there is something white in the scene, however, we can not be certain of its location or shape. In fact, there are a very large number of shapes which are consistent with this data. Using a-priori information we might attempt to pick the most likely shape, but this would only reflect our bias in the shape of the prior. Alternatively we could choose to represent the entire distribution of shapes explicitly, as a probability distribution. With a sufficiently agnostic prior, the posteori probability of each voxel being filled would be close to uniform. Subsequent processing can then be performed using the entire distribution. For example, one could render the expected image of this distribution; in which case each voxel is entirely repsonsible for a pixel some percentage of the time. In contrast, the Roxel algorithm represents this volume as a semi-transparent white fog. In this representation each voxel is responsible for some percentage of the observed intensity. In both interpretations the expected observations are indistinguishable.

Finally, it is clear that this paper is conspicuously lacking a proof for the convergence of the Roxels algorithm. While no proof currently exists, experiments imply that the process does reliably converge (see Figure 6 for two examples). In fact we have encountered no data set which has failed to converge. Unlike a gradient descent procedure, the Roxel algorithm does not require a "step size" parameter. Each step is direct algebraic invocation of some constraint on the eventual solution.

5 Experiments

Performance of the Roxel algorithm was examined on a variety of real and synthetic data. While the algorithm is quite general regarding the positions of cameras, in our experiments we positioned the cameras around a circle, pointed toward the center of the circle. The experiments used 36 camera positions.

In the first experiment a set of synthetic images were generated using a POV-Ray, a public domain graphics package. Thirty six 128x128 images were generated of a scene containing two chess pieces, a red rook and a white knight. The free parameter, σ, was set to 3% of the maximum color difference. This value for σ is realtively low, encompassing our knowledge that there is little noise or calibration error in the input images. The resulting volume contains 128x128x128 element and was computed using 3 passes through the Roxel algorithm. Figure 1 contains three images: an example input image, a horizontal cross-section of α, and a view synthesized from the recovered voxel space.[3] While in many ways the chess pieces are an

[3] Synthesized views were also rendered using POV-Ray.

"easy" synthetic dataset, it does serve to illustrate the accuracy of the technique.

The second experiment was also synthetic: a white ovoid and a transparent yellow box. All parameters were the same as in the first experiment. Results are displayed in Figure 2. The system is able to recover the transparency of the box, and the opacity of the ovoid. However, the structure is not recovered perfectly. We believe that this is due to the current scheme for estimating voxel colors (as a simple weighted average). We are currently working on improving these color esimates.

In order to facilitate experiments with real data, a scanning device was constructed that automatically captures images by rotating an inwardly pointing camera around a stationary observation platform. The radius of the circle swept out by the camera is 1.5 feet. Images were acquired with a standard Pulnix color camera equipped with a 35mm lens. For each experiment, a set of 36 images were acquired from positions distributed uniformly around the circle.

In the third experiment images of a plastic children's toy were acquired. The images are 320x240 and reconstruction volume is 320x320x240. A larger value of 8% was used for σ to compensate for noise in the imaging system, and slight errors in angular position. Four iterations of the Roxel algorithm was used to recover the volume. Figure 3 contains an input image, cross-section, and a synthesized view.

In the fourth experiment 160x120 images of a plastic dinosaur were used to reconstruct a 160x160x120 volume. Once again σ was set to 8%. The results for this scene are shown in Figure 4.

The fifth experiment was designed to demonstrate the Roxel algorithm on a real transparent scene. A set of lego bricks were placed inside of a frosted glass jar. The images used are 160x120 and reconstruction volume is 160x160x120. Parameters of the Roxel algorithm: 5 iterations and $\sigma = 8\%$. The results for this scene are shown in Figure 5.

Finally, since this paper does not include a proof of the convergence of the Roxels algorithm, the convergence properties were tested on two of these image sets. Figure 6 plots the change in α versus iteration for the data shown in Figures 4 and 5. For each time step the *largest* change in α across all the voxels is plotted. Convergence is apparent after no more than a few iterations.

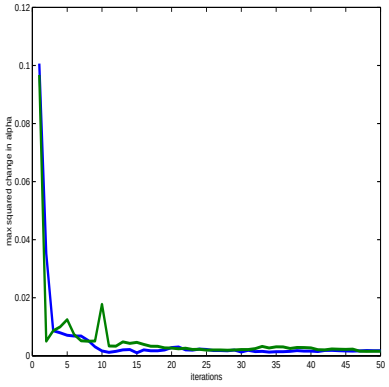

Figure 6. Convergence results for "Dinosaur" and "Legos in a Jar". At each point in time the largest change in α across all voxels is plotted.

6 Conclusions

The Roxel reconstruction algorithm is able to reconstruct volumetric representations of 3D space from a collection of observed images. By incorporating information from throughout the volume in determining opacity, accurate reconstruction of opaque and transparent materials can be performed. The Roxel algorithm is able to reconstruct volumes from natural imagery which contain both opaque and partially transparent materials. Reconstructions represent the ambiguity in regions of space in which there is little certainty about the exact structure due to limited, nonexistent, or contradicting data.

Acknowledgments

This work was supported by a grant from NTT and a fellowship from Microsoft. The original idea for this work grew out of a discussion with Steve Seitz.

References

[1] S.T. Barnard and M.A. Fischler. Computational stereo. *Surveys*, 14(4):553–572, December 1982.

[2] U.R. Dhond and J.K. Aggarwal. Structure from stereo: A review. *SMC*, 19(6):1489–1510, November 1989.

[3] David T. Gering and William Wells III. Object modelling using tomography and photography. *CVPR Workshop on Multi-view modeling and Analysis of Visual Scenes*, 1999.

[4] G. Herman and A. Naparstek. Fast image reconstruction based on a radon inversion formula ap-

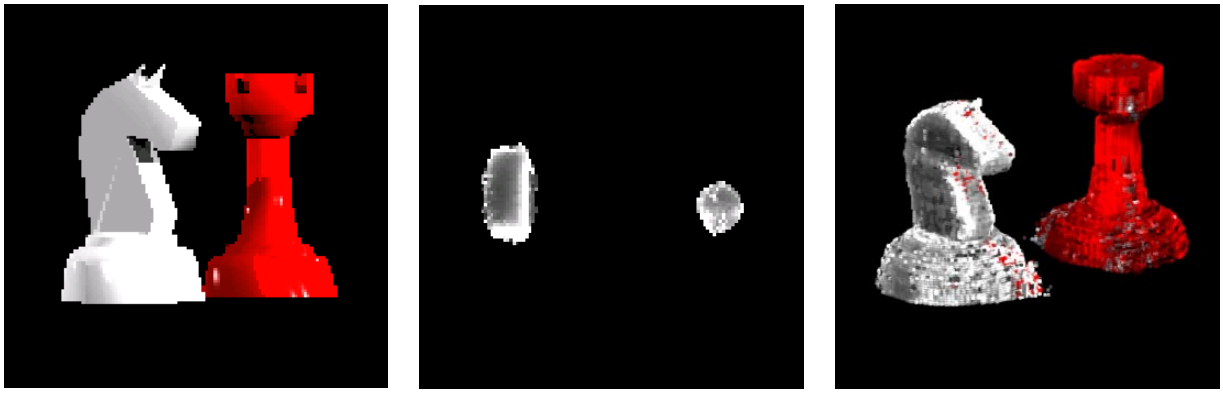

Figure 1. Results on synthetic images of chess pieces. Left: an original image. Center: A cross-section of α taken near the top of the objects. Right: Synthesized image.

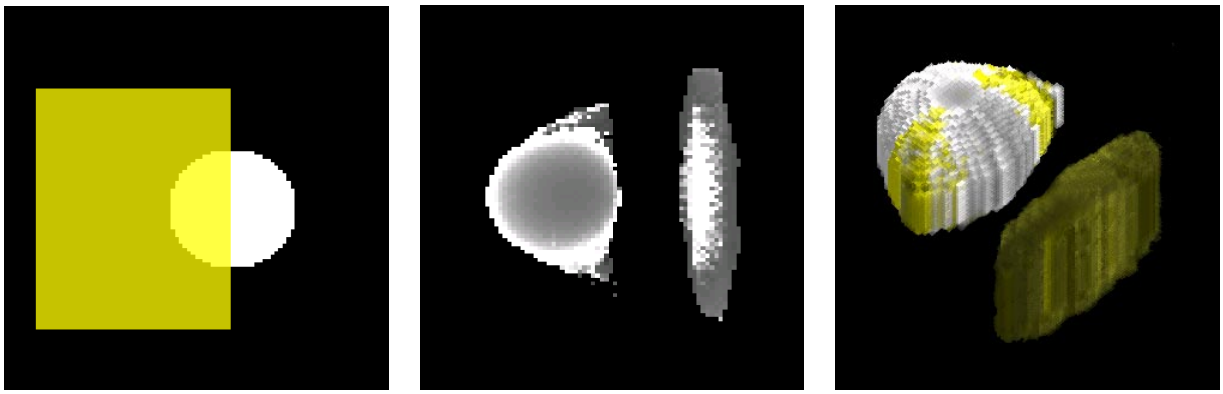

Figure 2. Results on synthetic images of transparency. Left: an original image. Center: A cross-section. Right: Synthesized image.

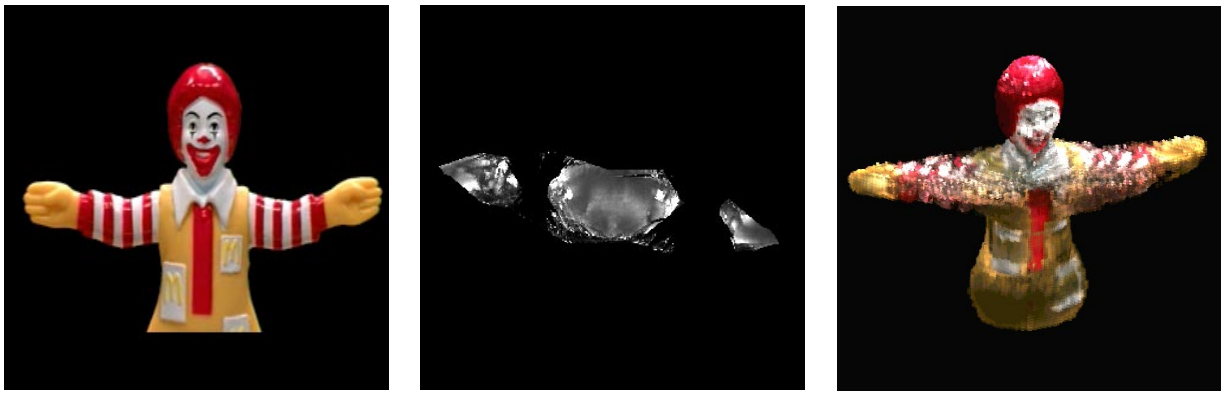

Figure 3. Results on images of a plastic children's toy. Left: an original image. Center: A cross-section showing the body and elbows. Right: Synthesized image.

Figure 4. Results on images of a plastic dinosaur. Left: an original image. Center: A cross-section showing the body and tail. Right: Synthesized image.

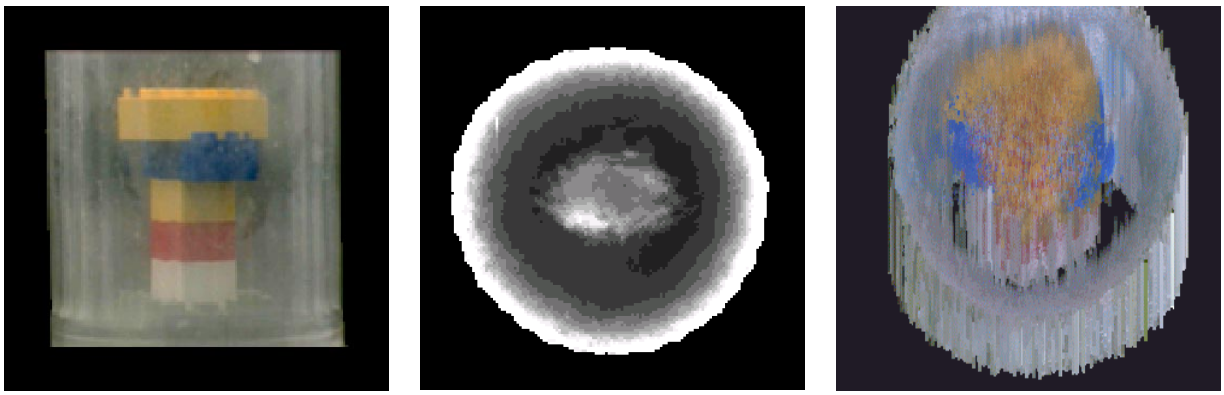

Figure 5. Results on images of legos in a jar. Left: an original image. Center: A cross-section showing the the jar and the legos. Right: Synthesized image.

proppriate for rapidly collected data. *SIAM Journal of Applied Mathematics*, 33:511–533, 1976.

[5] David Marr and Tomaso Poggio. Cooperative computation of stereo disparity. *Science*, 194:283–287, 1976.

[6] S.M. Seitz and C.R. Dyer. Photorealistic scene reconstruction by voxel coloring. In *CVPR97*, pages 1067–1073, 1997.

[7] L. Shepp and B. Logan. The fourier reconstruction of a head section. *IEEE Trans-actions on Nuclear Science*, NS-21:21–43, 1974.

[8] R. Szeliski and P. Golland. Stereo matching with transparency and matting. In *ICCV98*, pages 517–523, 1998.

[9] S. Webb, editor. *The Physics of Medical Imaging*. Institute of Physics Publishing, 1988.