Background Subtraction Using Self-Identifying Patterns

Mark Fiala and Chang Shu
Computational Video Group
National Research Council Canada
Montreal Road, Building M-50
Ottawa, Ontario, Canada K1A 0R6
{mark.fiala, chang.shu}@nrc-cnrc.gc.ca

Abstract

Separation of object foreground from background is used in 3D model creation and *matting* in video production. Robust background subtraction techniques that function in uncontrolled lighting environments would be useful for many applications. We introduce a method using bi-tonal self-identifying patterns as a background that can be used to recognize the foreground object despite the background intensity and colour being non-uniform across the image. Detected pattern points are used to sample the black and white colour levels in several image points. A surface is fitted to both the black and white colour levels allowing an estimated background image to be created. The background image is then subtracted from the original image to isolate the foreground objects. The method of using self-identifying patterns also provides the camera-pattern pose for use in 3D model creation. A visual hull 3D model can be created by identifying the outline of an object from several known camera poses. Examples of this method applied to both *matting* and 3D model creation are given. Experimental results are shown.

Keywords: background subtraction, 3D modeling, space carving, self-identifying markers

1 Introduction

Many imaging and graphics applications require segmenting foreground object from background pixels in an image or video. This problem is called *matting* in the film-making industry and *keying* in video production. In image processing, this problem is often referred to as background subtraction. Studios take great care to attempt to create a uniform background so that image pixels can be trivially segmented by thresholding their difference from a set colour. This uniformity of background pixels is not easy to achieve outside a studio.

A segmented image is useful for 3D model construction. A visual hull 3D model can be created by identifying the outline of an object from several known camera poses [6]. 3D modeling can be performed by space carving if the pose is known and the image is segmented into object and background pixels.

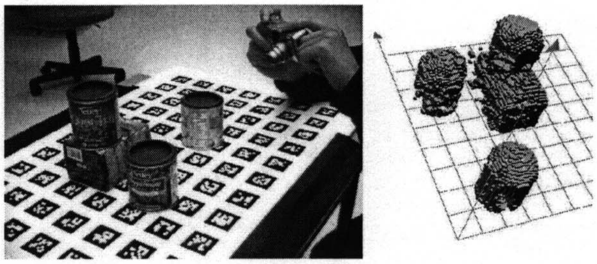

Figure 1: *3D model construction: A set of images are captured from different angles to automatically create a visual hull model. The background is a planar array of self-identifying markers (ARTag markers). The pattern-camera pose is calculated for each frame.*

To address both the background subtraction and camera pose determination problem, a background pattern of self-identifying patterns can be used. Points in the image where part of this pattern is detected with a high level of confidence can be used to create surfaces to estimate the variation across the image. We use a bi-tonal pattern, consisting of just black and white sections. A surface of the RGB appearance as a function of image coordinates is created for both black and white pattern shades, this is used along with knowledge of the full pattern to create what that background would look like in the absence of the object. This estimated background image is then subtracted from the original image to produce a foreground/background mask image (commonly communicated as an *alpha channel*).

The background pattern is an array of self-identifying *ARTag* markers which provide pattern-image point correspondences used to calculate the camera-pattern pose. This allows the foreground/background mask to be used to carve out 3D voxels to create a visual hull.

The 3D modeling can be done as simply as in Fig. 1 where a set of digital camera images are captured from different angles of the object or scene on the self-identifying background. The user does not need to measure where the images are taken from, knowledge of the camera parameters (focal length only in our experiments) is all that is needed along with the images to create a 3D model.

Examples are shown for both the image matting and 3D model making applications of using self-identifying pattern backgrounds.

2 Previous Work

2.1 Background subtraction

Background subtraction techniques fall into two groups. One group works with a known or controlled environment and the other group works on images with natural scenes. Among the techniques in the first group, blue-screen matting is the oldest and most used method, especially in the film-making industry. Smith and Blinn [12] gives a good description and analysis of this previously considered "black art", which involves experienced users to tune a few parameters. In particular, they provide an effective solution in which the background consists of two shades of colour. Their method, however, requires two shots of the scene, one for the background, one for the background with the subject. Qian and Sezan [9] also pre-shoot the background and compute the difference between the background image and the image with the subject. They apply diffusion techniques to improve the boundary areas.

The work on natural image matting has concentrated on estimating the probability of each pixel to decide whether it belongs to the background or the foreground. Ruzon and Tomasi [10] and Chuang et al. [1] demonstrate good results by using a Bayesian approach. More recently, Sun et al. [13] propose to use the gradient of the alpha channel and solve a Poisson equation to estimate the foreground and the background. All these methods rely on the user to carefully specify a *trimap*—regions that belong to the background, regions belong to the foreground, and region that is unknown.

2.2 Model building

Many systems have demonstrated the capability of constructing 3D models from multiple images. In this application, one not only needs to extract the foreground subject from the image, but also needs to decide the poses of the cameras. In a recent work, Matusik et al. [8] uses a plasma monitor displaying a two-coloured background. They pre-record the monitor image and then put the subject on a turntable in front of it. A commercial software package from Canon [2] uses a circular pattern to calculate the camera poses. They rely on controlled uniform background and the user editing to separate the object from its background. The object has to be elevated above the self-identifying pattern and the user must manually select and reject images with incorrectly determined foreground outlines.

Our system is similar to the Canon software, which uses a self-identifying pattern to decide the camera poses. However, unlike previous methods that require pre-recording of background or user assistance we extract the subject from the background and determine the camera poses automatically.

2.3 Self-identifying patterns: ARTag

Self-identifying patterns are special marker patterns that can be placed in the environment and automatically detected in camera images. Also known as *fiducial marker systems*, a library of these patterns and the algorithms to detect them help to solve the correspondence problem. Self-identifying marker systems such ARToolkit [4] and ARTag [3] are typically used for applications such as calculating camera pose for augmented reality and robot navigation.

The ARTag self-identifying pattern system was employed in our system as a background, objects would be imaged in front of a planar background pattern of ARTag fiducials at known locations. The patterns are used for two purposes; it is used to reliably identify which image points belong to the background, and to calculate camera pose for 3D model reconstruction.

ARTag was chosen because of its availability (can be downloaded from[1] and added to user programs), its robustness to lighting variation, its very low false positive detection rate, and its very low inter-marker confusion rate (falsely identifying the marker ID). ARTag fiducials are square planar bi-tonal patterns which have a square border and internal 36 bit digital pattern (Fig. 2).

[1] www.artag.net

Fig. 2 shows the markers being automatically located in an image. ARTag has some robust features that allow it to detect markers when partly occluded, however this "incomplete marker detection" can be turned off so that only complete background markers are used as sample background points.

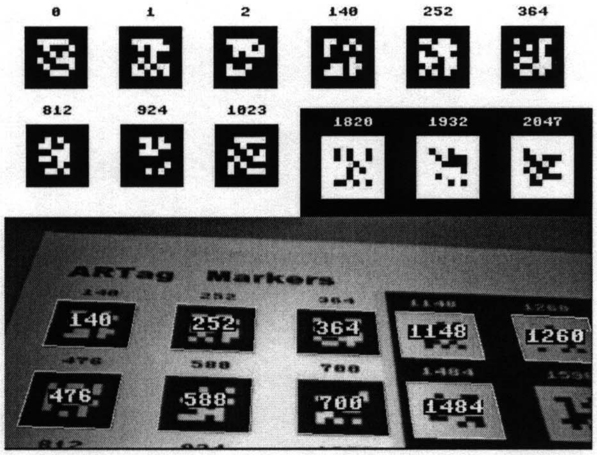

Figure 2: *ARTag self-identifying patterns. ARTag markers are bi-tonal planar marker patterns consisting of a square border and a 6x6 interior grid of cells representing logic '1' or '0'. (Top) 12 out of the library of 2002 markers are shown. (Bottom)ARTag self-identifying markers located in the image. The ID and sub-pixel location of the four corners are provided for each marker located, they are overlaid on the image with a quadrilateral and ID number in this image for visualization of the marker detection stage.*

3 Approach

The foreground can be separated from the background using background subtraction. If the background could be estimated despite variations in its appearance, due to lighting, then similar results may be achieved outside the controlled lighting of a studio.

Our method uses known points in the background to create a surface for each colour channel. To automatically identify the background, a self-identifying pattern of ARTag markers is employed. The detected ARTag marker points are used to calculate the pattern to image homography which is used in creating the estimated background for each image. The projection matrix is also calculated for each image and used to space carve a 3D model.

4 Background Sampling

The background pattern is bi-tonal (two shades of colouring on the pattern surface), however there will likely be variation in the RGB value across the image of the same background shade. Several samples are taken across the image and used by the procedure of Section 5 to estimate the RGB value at other points in the image. Since our pattern is bi-tonal this is performed twice, one surface is generated for white and one for black.

The ARTag markers have a very low false positive detection error rate so it is robust to use points inside the marker to sample the RGB value (Fig. 3). In our experiments 50-100 points are found for each shade for each image.

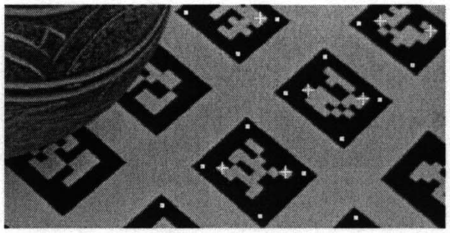

Figure 3: *Sampling of background points. Points inside the ARTag markers are known to belong to the background. Samples of the RGB values are taken for points corresponding to white and black points in the ARTag marker. White samples are shown as white crosses, black samples are shown as white dots. Only completely unoccluded markers are used for sampling.*

5 Lighting Approximation

Different points on the pattern surface receive different lighting. The intensity and colour of the light reflected from different parts of the surface may vary greatly. In order to reconstruct the image of the pattern with similar lighting conditions as the original image, we need to know the lighting conditions on the surface of the pattern. At the sample points, we record the RGB values at these locations. We estimate the image appearance on the entire surface of the pattern by interpolating the sampled data.

Assume we have a set of n sample points $P_1, P_2, \ldots, P_n$. Let the pixel coordinate of each point P_i be $\mathbf{x}_i = (u_i, v_i)$, and let the the RGB values of the pixel at P_i be (r_i, g_i, b_i). We want to find three bivariate continuous functions f_r, f_g, and f_b such that $r_i = f_r(u_i, v_i)$, $g_i = f_g(u_i, v_i)$, and $b_i = f_b(u_i, v_i)$. This is a typical scattered data interpolation problem. This problem arises from many fields in science and engineering and there are many solutions.

Lee et al. [7] uses B-splines to approximate irregularly sampled image data to solve the image warping problem. B-spline offers great smoothness and efficient evaluation. However, it relies on determining many variables including the degrees, the knot vectors, and the control points. The least square fitting is often ill-conditioned when data points are clustered.

In this work, we use natural neighbour interpolation [11]. The basic idea of natural neighbour interpolation is to compute the value of the interpolating function at a point based on a weighted average of the data values of its neighbouring points. The method pre-process the input data by constructing a Voronoi diagram of these points. The Voronoi diagram, which is the geometric dual of Delaunay triangulation [14], subdivides the image plane into disjoint cells, each corresponding to a data point and containing the points in the plane that are closer to this point than any other data points. When evaluating the value of a new point, we first insert the new point into the Voronoi diagram to compute its own new cell. The intersection of the new cell with the old cells results in a set of convex polygons. The areas of these polygons represent the contributions of these neighbouring points to the new points. The function value of the new point is computed as the weighted average of the values of the neighbouring points—weighted by the areas.

A natural neighbour interpolant can be made C^1 continuity by estimating the gradient at each data points. It is a more robust method than the B-spline fitting method because it uses only area averages. Its use of area averages also avoids the unpleasant effect caused by clustered data points, a situation that may happen in our application.

Our sample points are all lying in the interior of the image. Therefore, the interpolated function is only defined on the convex hull of the sample points. As we need to provide the estimation for entire image, we extrapolate the interpolated function to the boundary of the image. We do this by including the four corners of the image and use the colour value of the points that are nearest to them for the interpolating values.

Fig. 5(a) shows the sample points and their colour values from the image in Fig 4(a). Fig. 5(b) shows the interpolated function plotted as a surface.

6 Background Reconstruction

6.1 Homography and projection matrix calculation

Let $\mathbf{X} = [X\ Y\ Z\ 1]^\top$ be a point in 3-space and $\mathbf{x} = [x\ y\ 1]^\top$ be its image. Then they can be related by

$$\alpha\mathbf{x} = \mathbf{K}[\mathbf{R}\ \mathbf{t}]\mathbf{X} = \mathbf{P}\mathbf{X}, \qquad (1)$$

where $\mathbf{R}$ is a rotation matrix, $\mathbf{t}$ is a translation vector, and α is an arbitrary constant. The projection matrix $\mathbf{P}$ is required for the 3D modeling. $\mathbf{K}$ is a 3×3 matrix containing camera intrinsic parameters

$$\mathbf{K} = \begin{bmatrix} f_u & s & u_0 \\ 0 & f_v & v_0 \\ 0 & 0 & 1 \end{bmatrix}$$

where f_u and f_v are the focal lengths, s is the skew factor, and (u_0, v_0) is the principal point.

We assume the pattern lies in the $Z = 0$ plane in the world coordinate system. Then

$$\alpha\mathbf{x} = \mathbf{K}[\mathbf{r_1}\ \mathbf{r_2}\ \mathbf{r_3}\ \mathbf{t}] \begin{bmatrix} X \\ Y \\ 0 \\ 1 \end{bmatrix} = \mathbf{K}[\mathbf{r_1}\ \mathbf{r_2}\ \mathbf{t}] \begin{bmatrix} X \\ Y \\ 1 \end{bmatrix}.$$

A point $\mathbf{x} = [xy1]^\top$ on the pattern is related to its image $\mathbf{x}' = [x'y'1]^\top$ by a homography $\mathbf{H}$, $\alpha\mathbf{x}' = \mathbf{H}\mathbf{x}$, where

$$\mathbf{H} = \mathbf{K}[\mathbf{r_1}\ \mathbf{r_2}\ \mathbf{t}]. \qquad (2)$$

The homography matrix $\mathbf{H}$ can be solved by using four or more pairs of matching points. Let $\mathbf{h} = [h_{11}, h_{12}, h_{13}, h_{21}, h_{22}, h_{23}, h_{31}, h_{32}, h_{33}]^\top$, the homogeneous equation $\mathbf{x}' = \mathbf{H}\mathbf{x}$ can be written in the form

$$\mathbf{A}\mathbf{h} = \mathbf{0} \qquad (3)$$

where each matching pair of points $\mathbf{x}'_i \leftrightarrow \mathbf{x}_i$ makes two rows of the matrix $\mathbf{A}$:

$$\begin{bmatrix} \mathbf{0}^\top & -\mathbf{x}_i^\top & y'_i\mathbf{x}_i^\top \\ \mathbf{x}_i^\top & \mathbf{0}^\top & -x'_i\mathbf{x}_i^\top \end{bmatrix}.$$

If we have n matching points, the matrix $\mathbf{A}$ is $2n \times 9$. By constraining $||\mathbf{h}|| = 1$, equation 3 can be solved by singular value decomposition. The solution is the eigenvector of $\mathbf{A}^\top\mathbf{A}$ with least eigenvalue [5].

Once the homography $\mathbf{H}$ is determined, the full rotation matrix $\mathbf{R}$ can be estimated from $\mathbf{r_1}$ and $\mathbf{r_2}$. The projection matrix $\mathbf{P}$ can then be found from Eqn. 1.

In our experiments we assumed $\mathbf{K}$ to be constant across the sequence, the center of projection (u_0, v_0) to be the image center, and the skew factor s to be zero.

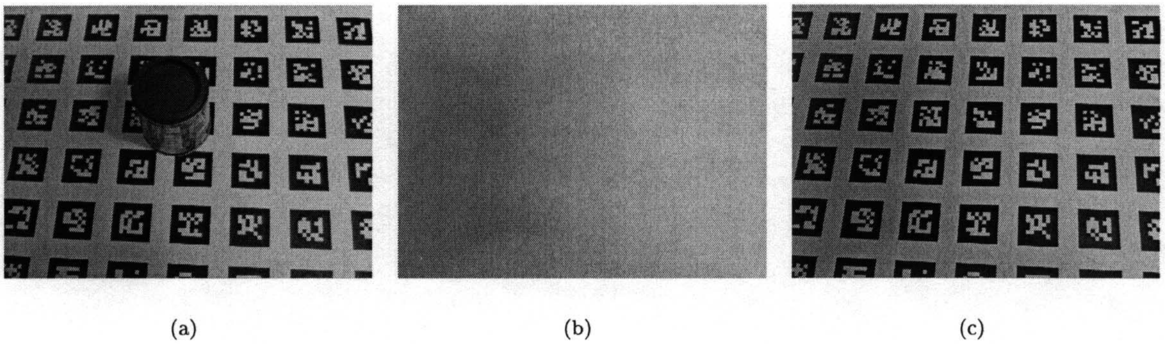

Figure 4: Approximating lighting conditions. (a) Original image; (b) Light intensity estimation using samples from the white points; (c) Reconstructed the pattern image.

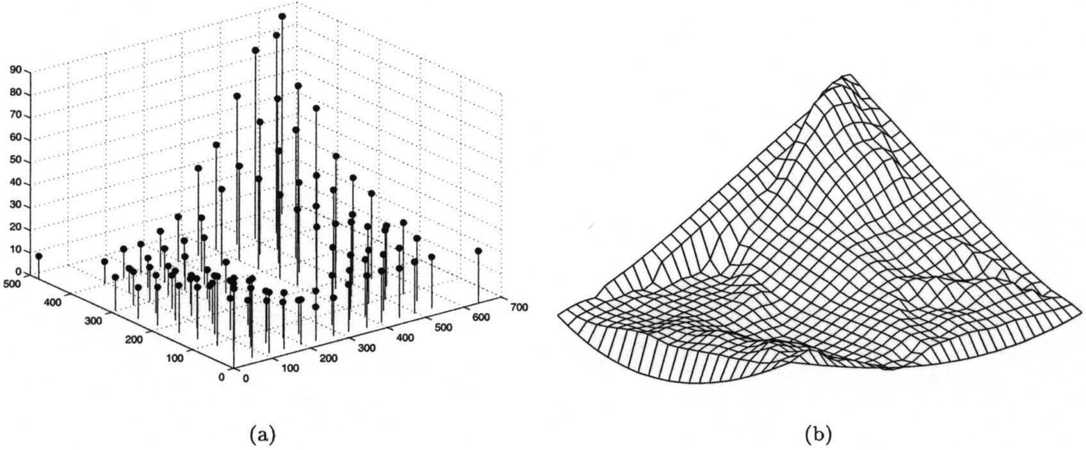

Figure 5: Sample data and the interpolation function plotted as a surface.

6.2 Binary pattern

For each image $\mathbf{I}_i$, the homography $\mathbf{H}_i$ is used to create a binary image of the background. The self-identifying pattern is an array of ARTag markers in known positions and a binary image is created for it. $\mathbf{H}_i$ is used to map this over to line up with the camera view as shown in Fig. 6.

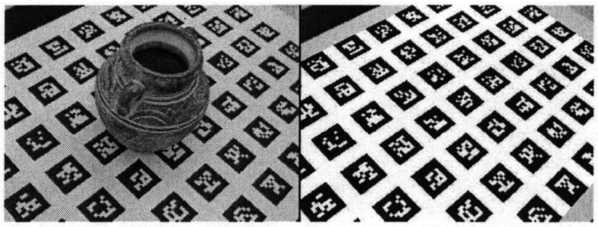

Figure 6: *(Left) Original image. (Right) Binary image of background converted to line up with the camera view with the homography calculated using the self-identifying pattern. (Right) shows what the pattern would be from that viewpoint despite occluding objects.*

6.3 Background reconstruction

The binary image from Section 6.2 above describes for each image pixel whether it would have a black or white shade assuming no foreground objects. This is used to obtain an RGB pixel from either of the two surfaces created by the method of Section 5. Performing this for each pixel yields the estimated background image, this is the estimate of what the camera would have seen had the foreground objects not been present. This is analoguous to having captured two images without moving the camera, one with the objects present and one with them absent. Examples of estimated background images can be seen in Fig. 4(c)(right) and in the second row of Fig. 7.

6.4 Background subtraction

Simple image subtraction can be performed between the original image and estimated background, with the result thresholded to produce a binary mask image. This mask image was filtered with the erode and dilate morphological operators to produce a cleaner mask image. The mask image is bi-tonal and each pixel simply indicates if it belongs to the object or background.

7 Results and Applications

7.1 Model building

3D models of objects were created with our system by capturing a set of images with varying viewpoints with the self-identifying pattern background. The ARTag planar array allows the projection matrix to be calculated (for a calibrated camera). From each camera image, the mask image and projection matrix is calculated. These are used to carve out a voxel space to create a 3D model.

Four experiments are shown, an object or scene set on the ARTag array and a set of images were taken without moving the object or adjusting the camera zoom. The experiment was performed in a room with diffuse lighting to avoid hard shadows. Fig. 7 and Fig. 8 show the results, each 3D voxel model was generated from 14 to 19 colour images of resolution 640x480 pixels using a Canon PowerShot S60 consumer digital camera. The focal length and aspect ratio was measured of the camera (complete calibration to find the image center and radial distortion, etc was not performed).

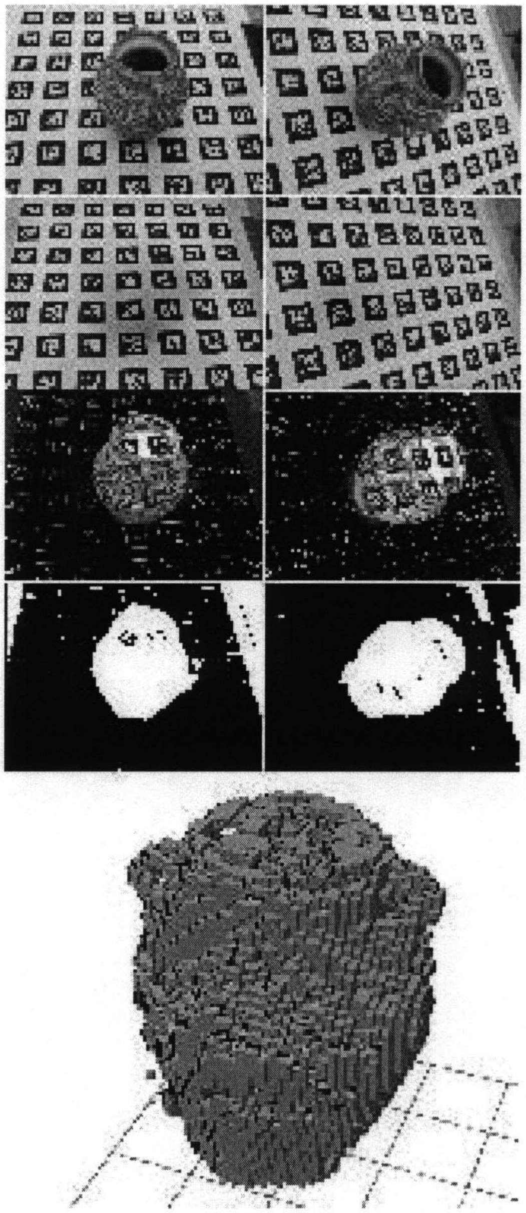

Figure 7: *Stages of 3D model creation. Voxel model (bottom) generated from 14 views, two of these images are shown (left and right columns). The stages are; original image (top), estimated background (2nd down), difference image (3rd down) and the binary mask. The visual hull model is created from space carving of the voxels from binary masks from the 14 image.*

Fig. 7 shows a 3D model generated of a vase along with the stages of processing for 2 of the 14 images. The ARTag patterns are detected and white and black sample points found which are known to be from the background. The white and black levels are then estimated for the image and used to create an estimated background which is subtracted from the original image using the euclidean distance of the RGB points. This greyscale difference image is thresholded to produce a binary image. Morphological operators were applied to clean up this binary image; one iteration of dilation, followed by two iterations of erosion, and finally by a single iteration of dilation produced the mask image used to carve out the voxel space. The projection matrix was calculated for each image using the ARTag pattern.

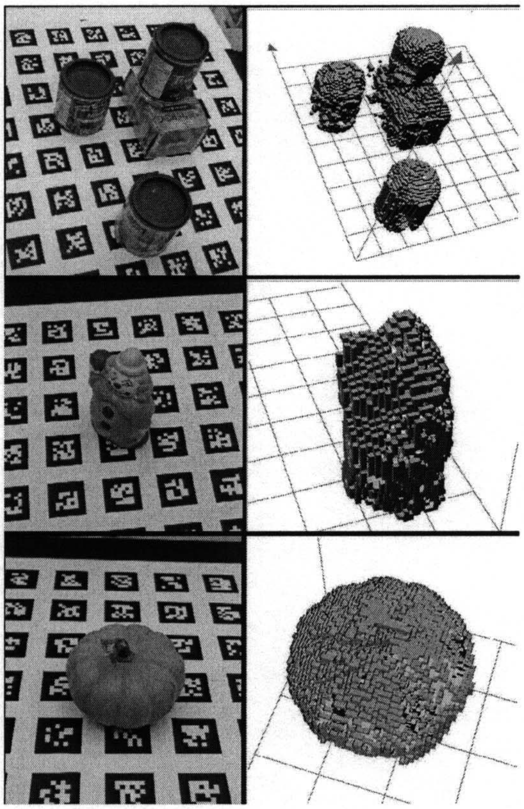

Figure 8: *Three more examples of model making. Input images (left column) were 640x480 from a consumer digital camera. 15 image were used for the tea boxes scene (top), 18 images for the clown (middle) and 19 for the pumpkin (bottom). All images taken were used, none had to be manually removed from the set. 3D models (right column) have some holes where the object colour too closely matched the white or black from the background pattern.*

The voxel space was carved simply by projecting each voxel center point into each of the mask images and removing those voxels that projected to a background pixel (black pixels in the mask images in Fig. 7). This results in a visual hull voxel model of the object.

There are some errors caused by when sections of the object have a similar colour to pixels in the background pattern, causing the binary mask to have gaps which create holes in the 3D models.

This demonstrates the feasibility of this background subtraction technique using self-identifying patterns, however future work is needed to carry this toward the goal of a system for creating 3D models useful for public use. Computer animators, game developers, etc expect a 3D texture mapped mesh that is not just the visual hull. Further voxels can be removed using photo-consistency methods and/or the voxel model can converted into a 3D mesh or NURB surface description that can be refined to best fit to the input image set.

7.2 Image composition

Our background subtraction technique can be used for making composite images as show in Fig. 9. The "holes" in the binary mask image occur when subject pixels have the same colour as the estimated background pixels, this was more of a problem with the human subjects than with the objects of Section 7.1 because of a similar shade of black existing in both the pattern and the people. Future work to improve this could involve using connectivity to remove holes in the main object, or using colours other than white and black for the self-identifying pattern background.

8 Conclusions

We have presented a method for segmenting foreground and background in an image and demonstrated its usefulness by applying the technique in two applications; building 3D models from multiple images, and image compositing. From an image of the subject in front of a planar array of self-identifying patterns (ARTags markers), our system separates the background and the subject automatically. The ARTag markers also relate the feature points in the images to their corresponding points in the pattern, resulting in automatic determination of the camera pose for each image. Therefore, 3D models can be created from voxel space carving.

The key to the success of our background subtraction system is that we estimate the lighting conditions on the surface of the pattern by interpolating samples from the ARTag feature points. This allows us to reconstruct the image of the background pattern under uncontrolled lighting conditions.

Our research suggests several intriguing directions for future work. One extension to the current system is to apply more sophisticated statistical techniques like the one suggested by Chuang et al. [1] to refine the boundaries between the subject and the background. Another possible extension is to make the algorithm run in real-time so that it can be used in online video matting applications.

Acknowledgements:

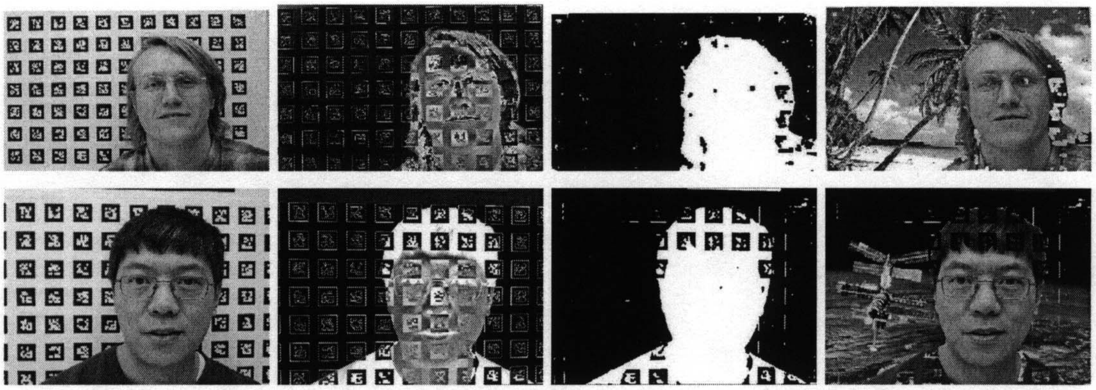

Figure 9: Image composition.

The authors would like to thank Pavel Sakov of the CSIRO for his natural neighbour interpolation code.

References

[1] Yung-Yu Chuang, Brian Curless, David H. Salesin, and Richard Szeliski. A bayesian approach to digital matting. In *Proceedings of IEEE CVPR 2001*, volume 2, pages 264–271. IEEE Computer Society, December 2001.

[2] Canon Research Centre Europe. *Canon 3D software object modeller*. `http://www.cre.canon.co.uk/3dsom.htm`.

[3] M. Fiala. Artag, an improved marker system based on artoolkit. In *National Research Council Publication NRC 47166/ERB-1111*, 2004.

[4] I. Poupyrev H. Kato, M. Billinghurst. *ARToolkit User Manual, Version 2.33.* Human Interface Technology Lab, University of Washington, 2000.

[5] R. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision.* Cambridge University Press, 2000.

[6] A. Laurentini. The visual hull concept for silhouette based image understanding. *IEEE PAMI*, 16(2):150–162, 1994.

[7] Seunyyong Lee, George Wolberg, and Sung Yong Shin. Scattered data interpolation with multilevel B-splines. *IEEE Transactions on Visualization and Computer Graphics*, 3(3):228–244, 1997.

[8] Wojciech Matusik, Hanspeter Pfister, Addy Ngan, Paul Beardsley, Remo Ziegler, and Leonard McMillan. Image-based 3D photography using opacity hulls. In *SIGGRAPH '02: Proceedings of the 29th annual conference on Computer graphics and interactive techniques*, pages 427–437. ACM Press, 2002.

[9] R.J. Qian and M.I. Sezan. Video background replacement without a blue screen. In *Proceedings of the International Conference on Image Processing (ICIP'99)*, pages 143–446, 1999.

[10] Mark A. Ruzon and Carlo Tomasi. Alpha estimation in natural images. In *Proceedings of IEEE CVPR 2000*, volume 1, pages 18–25. IEEE Computer Society, 2000.

[11] R. Sibson. A brief description of natural neighbour interpolation. In V. Barnett, editor, *Interpreting Multivariate Data*, pages 21–36. Wily, 1981.

[12] Alvy Ray Smith and James F. Blinn. Blue screen matting. In *SIGGRAPH '96: Proceedings of the 23rd annual conference on Computer graphics and interactive techniques*, pages 259–268. ACM Press, 1996.

[13] Jian Sun, Jiaya Jia, Chi-Keung Tang, and Heung-Yeung Shum. Poisson matting. *ACM Trans. Graph. (SIGGRAPH 2004)*, 23(3):315–321, 2004.

[14] D. F. Watson. Computing the n-dimensional Delaunay tessellation with application to Voronoi polytopes. *The Computer Journal*, 24(2):167–172, 1981.