Dense 3-D Structure from Image Sequences Using Probabilistic Depth Carving

Annie Yao and Andrew Calway

Department of Computer Science
University of Bristol, UK

{yao,andrew}@cs.bris.ac.uk

Abstract

We describe an algorithm to determine dense 3-D structure in a static scene from an image sequence captured by a moving camera. Metric camera motions are first determined using a recursive structure from motion algorithm based on tracked feature points. Dense depth information for a subset of key frames is then obtained using a novel *probabilistic depth carving* algorithm - analogous to space carving - in which depth probabilities obtained locally about the key frames are combined in 3-D space. An important component in this process is that opacity and occlusion relationships are modelled explicitly, enabling consistent combination of the depth probabilities. Results of experiments on a real sequence illustrate the effectiveness of the approach.

1 Introduction

Extracting dense 3-D structure of a static scene from an image sequence captured by a moving camera has received considerable attention in recent years.This has been fuelled by advances made in determining estimates of 3-D camera motion, typically using a sparse set of tracked feature points and without the need for pre-calibration. This enables dense 3-D surface estimates to be obtained using triangulation once corresponding points amongst the frames have been established. These can be identified either explicitly by matching at pixel resolution amongst the frames, often referred to as *multiview stereo* [6, 8, 7], or implicitly by sampling in 3-D space, as in *space carving* [9, 1, 4, 5]. The former gives dense depth estimates at pixel resolution with respect to each frame, whilst the latter gives increased 3-D resolution, but at the expense of increased computational cost.

In multiview stereo, the known camera motions constrain correspondences and matching becomes a search in depth for each position in a given frame. Matching similarities need to be based on pixel differences in order to obtain dense correspondences but interpreting these between and across frames is a difficult task; the presence of occlusions, shadows, reflectance, perspective distortions, lack of texture, etc, can all cause mismatches. Although these effects are minimised by matching nearby frames, this in turn leads to increased ambiguity in 3-D estimation when attempting to triangulate near-parallel rays. Conversely, frames far apart can give increased 3-D accuracy but establishing dense correspondence is more difficult. Some frames are also likely to reveal parts of the scene occluded from others and thus selective correspondence amongst subsets of

BMVC 2003 doi:10.5244/C.17.22

frames is often necessary. Simple extensions of two frame stereo matching are therefore rarely effective for the multiview case and recent work has concentrated on more sophisticated approaches. This includes the use of additional constraints combined with dynamic programming or graph cut methods [6, 8, 7], correspondence linking across frames [8, 6], locally adaptive matching [7], temporal selection [7], combined models of disparity, colour and opacity [11], and hybrid edge and region matching [6].

Space carving [9] avoids the problem of having to establish dense correspondences amongst the frames. Instead, 3-D space is quantised into voxels which are projected onto the frames to assess their validity as surface points. Voxels are visited in order from near to far and invalid voxels are removed or 'carved away'. Once complete, the remaining voxels then define the surfaces within the scene. Validity assessment is usually based on the colour variance amongst the projected points assuming a Lambertian reflectance model. The method can give dense 3-D surface estimates and tackles occlusions in a natural way. However there are drawbacks. Sampling 3-D space typically requires a large number of voxels, resulting in high computational costs. Voxel processing order is also important for efficient implementation and practical use relies on the 'plane sweep algorithm' which precludes arbitrary view point combination. The use of hard decisions on voxel validity can also be problematical, with early errors cascading to produce holes in the estimated structure. Recent work on modifications has addressed some of these problems, notably the use of probabilistic frameworks to avoid making hard decisions and to allow the combination of arbitrary viewpoints [5, 4, 1].

The work described here is a multiview stereo method but one which incorporates some benefits from space carving. Dense depth estimates are first obtained for a subset of key frames using local correspondences amongst a small number of adjacent frames. The estimates are then combined in depth space using a form of space carving - known as 'depth carving' - based on explicit models of opacity and occlusion. In effect, we use local multiview correspondence about the key frames to obtain an initial set of coarse depth estimates and then combine this set to improve depth accuracy. The combination uses a probabilistic formulation in which the likelihood that scene structure exists at a given depth is based on a combined visibility and opacity constraint amongst the key frames. This promotes depths consistent with the constraint, whilst penalising those which are not, hence the term 'depth carving'. The approach has the advantage of avoiding the complex correspondence matching across distant frames needed in multiview schemes such as [8, 6], and at the same time allowing arbitrary combination of views, unlike traditional space carving. It has similarities with the probabilistic formulations of space carving as described in [5, 1], whilst the use of explicit models for opacity and occlusions resembles in part the multiview stereo work of Szeliski and Golland [11]. In the next section we outline the depth carving algorithm, Section 3 then gives details of the probabilistic framework and Section 4 presents results of experiments on a real sequence.

2 Depth Carving

Given a 3-D static scene viewed by a moving camera, we assume that the relative position and orientation of the camera at specific time intervals are known. This is illustrated in Fig. 1, where for frame k the centre of projection of the camera is located at the 3-D position $\mathbf{C}_k$. Specifically, if $\mathbf{X}_k$ denotes the position vector of a 3-D point $\mathbf{X}$ defined with

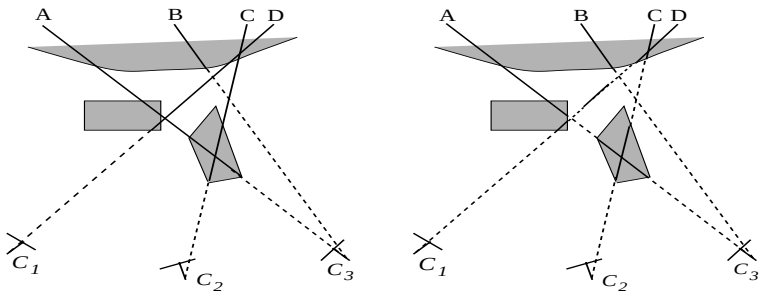


Figure 1: Depth carving: (a) Visibility values for viewing rays A-D in 3 camera frames C_1, C_2 and C_3 (bold $\equiv$ occluded, dotted $\equiv$ visible); (b) opacity values resulting from depth carving (bold $\equiv$ opaque, dotted $\equiv$ free space).

respect to the *k*th camera frame, then we have reliable estimates of the relative motion of the camera between frames *k* and *j*, ie $\mathbf{X}_k = R_{kj}\mathbf{X}_j + \mathbf{T}_{kj}$, where the matrix R_{kj} defines the 3-D rotation and the vector $\mathbf{T}_{kj}$ defines the offset between the camera positions. Here we use the recursive structure from motion algorithm developed by Azarbayejani and Pentland [2] to obtain the 3-D motion estimates based on a sparse set of tracked feature points. Further details are given in Section 4.

The depth carving technique is based on two properties of 3-D points in a scene - *opacity* and *visibility* - and a relationship between them which we call the *visible-opacity constraint*. We define opacity and visibility as follows.

Opacity. Assuming that the scene consists only of opaque objects, then the opacity associated with a 3-D point is defined as

$$\alpha(\mathbf{X}) = \begin{cases} 1 & \text{if } \mathbf{X} \text{ is an interior point} \\ 0 & \text{if } \mathbf{X} \text{ is a point in free space} \end{cases} \quad (1)$$

where an 'interior point' is a point within an object. This is a restricted form of the opacity used by Bonet and Viola [4]; unlike them we do not consider the case of transparent objects being in the scene for which $0 < \alpha(\mathbf{X}) < 1$. Note that if $\alpha_k(\mathbf{X}_k)$ denotes the opacity defined with respect to the *k*th camera frame, then the values $\alpha_k(a\mathbf{X}_k)$, for $a > 0$, define the intersection of the scene objects with the ray originating from $\mathbf{C}_k$ and passing through the point $\mathbf{X}$ (we call this the 'viewing ray'). Examples are shown in Fig. 1b, where the opaque points for viewing rays B and C are shown in bold and the free space points are shown dotted. Note that the opacity for surface points is not defined; they occur at the transition between free space and opacity (or *vice versa* for surfaces facing away from the camera, such as along rays A and C in Fig. 1b).

Visibility. The visibility $\upsilon(k,\mathbf{X})$ indicates whether the point $\mathbf{X}$ is visible in frame *k*, ie

$$\upsilon(k,\mathbf{X}) = \begin{cases} 1 & \text{if } \mathbf{X} \text{ is visible in frame } k \\ 0 & \text{if } \mathbf{X} \text{ is occluded from frame } k \end{cases} \quad (2)$$

where visible points are all free space points and occluded points can be interior as well as free space or surface points. Thus, if $\upsilon_k(k,\mathbf{X}_k)$ denotes the visibility defined with respect to the *k*th frame, then the values $\upsilon_k(k,a\mathbf{X}_k)$, for $a > 0$, define 3-D points visible along

the viewing ray $a\mathbf{X}_k$. This is shown in Fig. 1a, where the visible points along rays A-D are shown dotted and the occluded points are shown in bold. Note that in general $\upsilon(k,\mathbf{X})$ need not equal $\upsilon(j,\mathbf{X})$, $k \neq j$, and also that $\upsilon_k(k,\mathbf{X}_k)$ is directly related to $\alpha_k(\mathbf{X}_k)$ along the same ray, ie

$$\upsilon_k(k,\mathbf{X}_k) = \begin{cases} 1 & \text{if } \alpha_k(a\mathbf{X}_k) = 0 \quad \forall \quad 0 < a < 1 \\ 0 & \text{otherwise} \end{cases} \tag{3}$$

where again the visibility of the surface point closest to the camera along this ray is not defined - the location of the surface boundary corresponds to the transition between visibility and occlusion.

Based on these two properties, given an opaque scene and the condition that *every* free space point is visible in *at least one* frame, we have the following relationship between opacity and visibility

$$\alpha(\mathbf{X}) = \prod_k [1 - \upsilon(k,\mathbf{X})] \tag{4}$$

where the product is over all the camera frames. This is the *visible-opacity constraint*. It follows from two observations. First, by definition, an interior point must by occluded in every frame, ie for each frame k, $\upsilon(k,\mathbf{X}) = 0$ for all $\mathbf{X}$ such that $\alpha(\mathbf{X}) = 1$. Secondly, since every free space point is visible in at least one frame, then for every point $\mathbf{X}$ for which $\alpha(\mathbf{X}) = 0$ there exists at least one frame k_o such that $\upsilon(k_o,\mathbf{X}) = 1$. Thus eqn (4) will assign $\alpha(\mathbf{X}) = 1$ for every interior point and $\alpha(\mathbf{X}) = 0$ for every free space point, which agrees with the definition in eqn (1). Of course in practice we cannot realistically expect to view every free space point in a scene. However, note that given a surface point visible in frame k, the constraint in eqn (4) will hold for all the free space points lying along the viewing ray in front of the surface and all opaque points along the same ray beyond the surface, *even if* other points in the scene are hidden from all the frames. In other words, the constraint enables the structure of all visible surfaces in a scene to be obtained from the combination of visibility values $\upsilon(k,\mathbf{X})$, irrespective of the number or position of the frames. This is illustrated in Fig. 1b, which shows the result of applying the constraint using the visibility along rays A-D in Fig. 1a. Here the opacity values are correct for rays B and C since the free space points occluded from the respective frame are visible in at least one other frame. In contrast, some of the values for rays A and D are incorrect, with free space points being identified as opaque since they are occluded from all three frames.

The above relationships underlie the depth carving algorithm. Starting from initial estimates of the visibility $\upsilon(k,\mathbf{X})$, the constraint in eqn (4) provides a means of improving the estimates using iterative refinement - we can use the initial visibility estimates to obtain the opacity values from eqn (4) and then use these to obtain updated visibility values based on the relationship in eqn (3), and so on, hence building up opacity and visibility values with respect to each frame. Thus, it can be seen as a process of removing, or 'carving away', depths that correspond to free space, whilst retaining those opaque points which are supported by the visibility values from all the other frames, ie the depth corresponding to a point $\mathbf{X}$ is retained iff $\upsilon(k,\mathbf{X}) = 0, \forall\, k$. Hence the term *depth carving*. In practice this is best achieved using a probabilistic approach and in the next section we describe a sequential Bayesian formulation for the refinement process.

At this point it is worth noting that the above approach has similarities with the spacing carving algorithm described by Agrawal and Davis [1]. However, it also has important

differences. Although they use a similar refinement process, they use different constraints based on a likelihood measure for a visible surface point being at a given point along each viewing ray, rather than the opacity and visibility properties used in our formulation. As a consequence they need to use temporal selection of frames in order to avoid valid estimates being carved away due to occlusion. In contrast our formulation allows all the frames to be combined for each point being considered, giving increased potential for refinement and avoiding the need for temporal selection.

3 Probabilistic Framework

Our overall aim is to obtain estimates of the opacity values $\alpha(\mathbf{X})$ for 3-D points in the scene. We can then determine the surface structure from the transitions between free space and opacity as noted in the previous section. However, since we are using a multi-view stereo approach, we concentrate here on obtaining estimates of the opacity values at discrete depths along viewing rays passing through each pixel in a subset of key frames. This results in depth estimates with respect to each frame. These opacity values at pixel resolution would need to be combined in order to obtain estimates of $\alpha(\mathbf{X})$ using, for example, surface interpolation techniques as in [8]. This issue is not considered here.

The task is therefore to obtain estimates of $\alpha_k(\mathbf{X}_k)$ at discrete points $\mathbf{X}_k$ along each viewing ray. These are limited to 0 or 1 and so we treat it as a binary decision problem, ie a point is either in free space or is opaque, and obtain solutions using a Bayesian probabilistic formulation. Specifically, we use iterative refinement to update the probability that a point $\mathbf{X}_k$ is opaque given a measure of how well probabilities for visibility in all the frames support the visible-opacity constraint. This can be formulated as a sequential Bayesian update [3] in which previous estimates of the probability form the prior and the likelihood is given by a constraint support measure, ie for $n > 0$

$$P_{n+1}(\alpha_k = 1) \;=\; \frac{R_n\, P_n(\alpha_k = 1)}{R_n\, P_n(\alpha_k = 1) + (1 - R_n)\,(1 - P_n(\alpha_k = 1))} \tag{5}$$

where we have omitted the dependence on the point $\mathbf{X}_k$ for ease of notation. The likelihood R_n given the current state follows from eqn (4)

$$R_n \;=\; \prod_j \Big[1 - P_n(\upsilon_j(j, \mathbf{X}_j) = 1)\Big] \;=\; \prod_j P_n(\upsilon_j(j, \mathbf{X}_j) = 0) \qquad n > 0 \tag{6}$$

where as before $\mathbf{X}_j$ denotes the position vector for the point $\mathbf{X}_k$ defined within the jth camera frame and the product is over the subset of key frames. The iterative process in eqn (5) therefore promotes depths which support the visible-opacity constraint and penalises those that do not, hence arriving at a consistent solution across the key frames.

The above iterative process requires an expression for the probability that a point is occluded, $P_n(\upsilon_k(k, \mathbf{X}_k) = 0)$, and suitable initial conditions. Given updated probabilities for the opacity at the nth iteration we can base the former on the relationship in eqn (3): along a viewing ray, points beyond a point with high probability of being opaque should have high probabilities of being occluded (and *vice versa*). Let $\mathbf{Z}_i$ denote 3-D points along a viewing ray for frame k, then we use the following expression for the probability of occlusion

$$P_n(\upsilon_k(k, \mathbf{Z}_i) = 0) \;=\; [1 - P_n(\alpha_k(\mathbf{Z}_i) = 1)]\, f(\mathbf{Z}_i) \;+\; P_n(\alpha_k(\mathbf{Z}_i) = 1) \qquad n > 0 \tag{7}$$

(Carry out each step for each pixel in each frame. Initialise $n = 0$.)

1. Initialise the opacity probabilities so that $P_0(\alpha_k(\mathbf{Z}_i) = 1) = 0.5$, where the discrete points $\mathbf{Z}_i$ lie along the pixel viewing ray.
2. Initialise the occlusion probabilities $P_0(\upsilon_k(k, \mathbf{Z}_i) = 0)$ using eqns (7), (8) and (9), with $m(\mathbf{Z}_i)$ replacing $P_n(\alpha_k(\mathbf{Z}_i) = 1)$ in eqn (7).
3. Update opacity probabilities $P_{n+1}(\alpha_k(\mathbf{Z}_i) = 1)$ using eqns (5) and (6).
4. Update occlusion probabilities $P_{n+1}(\upsilon_k(k, \mathbf{Z}_i) = 0)$ using eqns (7) and (8).
5. Terminate if converged; otherwise increment n and goto to step 3.

Figure 2: The depth carving algorithm.

where $f(\mathbf{Z}_i)$ is a monotonic function which tends to 1 if $\mathbf{Z}_i$ is likely to be occluded and to 0 if it is likely to be visible. We used the following form for this function

$$f(\mathbf{Z}_i) = 1 - \exp(-s^2/2\sigma_f^2) \qquad s = max\{\Lambda_i\}/max\{\Lambda\} \tag{8}$$

where Λ and Λ_i are the set of opacity probabilities for all points along the ray and all points in front of Z_i, respectively. Thus, $P_n(\upsilon_k(k, \mathbf{Z}_i) = 0)$ will tend to 1 if $\mathbf{Z}_i$ is preceded by a point with relatively high probability of being opaque and tend to $P_n(\alpha_k(\mathbf{Z}_i) = 1)$ otherwise, ie for visible points it assumes the same (low) probability of opacity. The variance term σ_f^2 controls the fall off between these two cases. In practice, since the iterative process is applied to discrete points along viewing rays in each key frame, we interpolate a value for $P_n(\upsilon_j(j, X_j) = 0)$ in eqn (5) from surrounding pixels and depths.

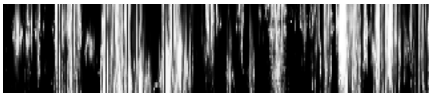
The required initial conditions are $P_0(\alpha_k = 1)$ for each frame and $P_0(\upsilon_k(k, \mathbf{Z}_i) = 0)$ along each viewing ray. We make no *a priori* assumptions about the structure in the scene and so set $P_0(\alpha_k = 1) = 0.5$ for all points, ie points are equally likely to be opaque or in free space. Establishing initial probabilities for occlusion is more tricky without *a priori* structure information. However we can obtain estimates of the likely position of visible surface points in the scene with respect to each frame by using traditional correspondence matching. If we view the resulting matching similarities as opacity probabilities then we can generate initial occlusion probabilities using eqn (7). This is valid since along each viewing ray the visibility is determined solely by the location of the surface. We can obtain estimates of this by forming correspondences about each key frame using nearby frames. As noted earlier these local correspondences are less likely to be affected by mismatches than using correspondences across more distant frames and their lack of accuracy is not critical here: since they are used to bootstrap our probabilistic method, we are more interested in the relative likelihoods - providing we obtain high probabilities for depths on or in front of a visible surface then the probabilistic combination with other frames will resolve any ambiguities.

Thus for a point $\mathbf{X}_k$ along a viewing ray in frame k, we obtain a matching similarity based on local correspondences within a set of nearby frames Γ_k as follows

$$m(\mathbf{X}_k) = \exp(-med\{[I_k(\mathbf{x}_k) - I_j(\mathbf{x}_j)]^2, j \in \Gamma_k\}/2\sigma_m^2) \tag{9}$$

where $med\{\}$ denotes the median operation and $I_k(\mathbf{x}_k)$ is the intensity value in frame k at

(a) (b)



(c)

Figure 3: (a) Frame from desk sequence with scanline; (b) Depths with highest matching similarity using local correspondence matching; (c) depth matching similarities for pixels along the scanline in (a).

the pixel $\mathbf{x}_k$ corresponding to the projection of $\mathbf{X}_k$. Recall that $\mathbf{X}_j$ is the position vector of point $\mathbf{X}_k$ defined within the jth camera frame and hence $\mathbf{x}_j$ is the corresponding point of $\mathbf{x}_k$ in frame j. In practice we interpolate the intensity in frame j from surrounding pixels. We use the median of the squared intensity differences here in order to incorporate a degree of robustness and σ_m is set to reflect the expected variance between the nearby frames. The matching costs $m(\mathbf{X}_k)$ along a given ray are then used to generate the initial probabilities for the visibility using eqn (7) as discussed above. This completes the description of the depth carving algorithm; a summary of the steps is given in Fig. 2.

4 Experiments

The algorithm was tested on an image sequence consisting of 76 frames of a desk scene captured using a hand-held video camera. One of the key frames used in the depth carving is shown in Fig. 3a. The camera moves in front of the scene keeping the desk lamp roughly in the centre of the frame. There is significant occlusion of the background by the lamp and the scene contains a number of areas with little or no texture as well as small detail such as the lamp stand. As noted earlier we use the recursive structure from motion algorithm developed by Azarbayejani and Pentland [2] to obtain the 3-D camera motions. This requires a set of tracked feature points (we used the KLT tracker [10] to give around 20 corresponding points across the sequence) and gives estimates for the point depths and the camera focal length in addition to the motion. The estimation of focal length avoids the need for pre-calibration and gives metric estimates of the structure and motion.

We set the range of depths examined along each viewing ray based on the depths obtained from structure from motion algorithm and used 33 discrete depths per ray. The initial matching similarities were obtained by analysing 10 adjacent frames around each of 14 key frames which were each separated by 5 frames. A plot of matching similarities for each depth and for each pixel along the scanline shown on the key frame in Fig. 3a is

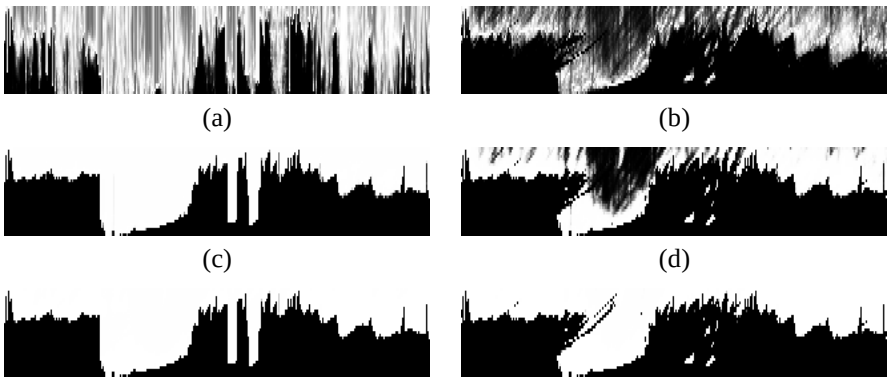
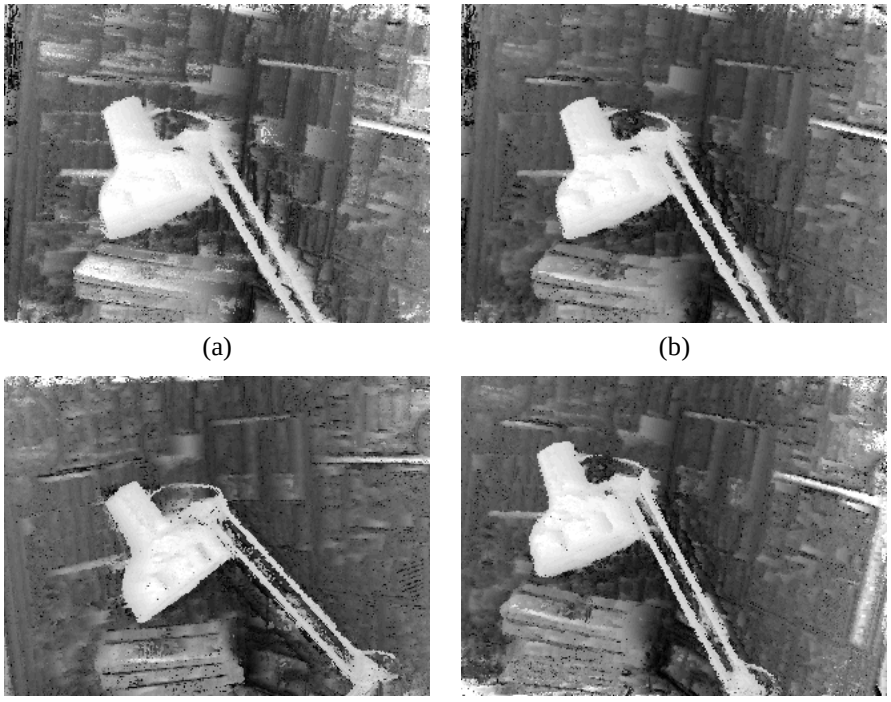


Figure 4: Depth carving results: (a,c,e) Occlusion probabilities for pixels along the scanline after 1st, 4th and 6th iterations; (b,d,f) Corresponding opacity probabilities.

shown in Fig. 3c with the closest depth at the bottom. Fig. 3b shows the depths with the highest matching similarity. Note that although selecting the most likely depth from the local matching gives acceptable estimates at object boundaries, there are significant errors across the whole frame, notably in regions lacking in texture, as would be expected. This is confirmed by the ambiguous similarities across the depths in Fig. 3c.

The results of using the depth carving algorithm are shown in Fig. 4. Estimated probabilities for occlusion and opacity for depths at each pixel along the scanline of the same key frame after the first iteration of the algorithm are shown in Figs 4a-b, respectively. The occlusion probabilities are those used to generate the opacity probabilities for that iteration. For the early iterations we used a larger value for the variance parameter σ_f^2 in the function $f()$ used to obtain the occlusion probabilities; starting at 1.0 and reducing it linearly to 0.25 by the 4th iteration. The effect of the larger variance is to increase the relative probabilities of points immediately behind points with high probability of being opaque. We found this to be advantageous for the first few iterations since it helps to establish the opacity of the surface structure in the scene; the remaining interior points are then picked up in later iterations using a smaller variance. This can be seen in Figs 4a-b, where the occlusion probabilities tend to be larger for closer depths and the opacity values are high around the surface boundaries. Note also that the occlusion probabilities are quite poor, with large portions of visible free space being classified as opaque. This results from the ambiguous depth similarities in Fig. 3c. Despite this, the opacity probabilities in the vicinity of surfaces are reasonable, with both the lamp shade and the stand being clearly identified. Figs 4c-f show the corresponding probabilities after the 4th and 6th iterations. These show the convergence of both occlusion and opacity values, and in particular the emergence of the free space area behind the left side of the shade as the depths are carved away by the other views, as well as the retention of the lamp shade and stand.

These results are confirmed by the depth values shown in Figs 5a-b. Here we selected the closest depth with opacity probability above a given threshold, ie corresponding to the visible surface points in the frame. The threshold value is not critical given the convergence of the probabilities as indicated above. Fig. 5a shows the result after the 1st iteration for the key frame in Fig. 3a and Fig. 5b the result after the 6th iteration. Figs 5c-



(a) (b)

(c) (d)

Figure 5: Visible surface depths: (a)-(b) after 1st and 6th iteration for view in Fig. 3a; (c)-(d) for two other views after 6th iteration.

d show the results after the 6th iteration for two other key frames. Note the improvement after 6 iterations and how the depth discontinuities have been detected in all the views. The depth errors in the background are less satisfactory and are primarily remnants of the false foreground depths detected by the initial local correspondences which were not carved out when combined with other views. We anticipate that these would be removed if the sequence contained views from additional directions; a hypothesis supported by the fact that the errors lie primarily along the dominate horizontal motion direction.

5 Conclusions

We have presented a novel multiview stereo algorithm which incorporates some of the benefits of space carving defined within a probabilistic framework. One of the main contributions is that it provides a coherent approach for combining depth information from wide baseline views, incorporating explicit relationships between visibility and opacity, without the need for complex correspondence linking across frames. A key factor is that the combination process utilises all of the frames, avoiding temporal selection and resulting in an opacity representation which is globally consistent, ie the opacity values determined for a given view include those for points which are occluded from that view and which are consistent with that determined for other views. This is in contrast with the usual multiview approach in which a post processing operation is adopted to 'build'

a consistent representation. The results presented here are encouraging and illustrate the potential of the approach. Our future work will concentrate on improvements, including the use of multiresolution processing and the incorporation of spatial and depth smoothness constraints.

Acknowledgements

The authors are grateful to the Independent Television Commission, UK, for financial assistance and to the reviewer who pointed out an error in the original definition of the visible-opacity constraint.

References

[1] M Agrawal and L.S Davis. A probabilistic framework for surface reconstruction from multiple images. In *Proc Conf on Computer Vision and Pattern Recognition*, 2001.

[2] A Azarbayejani and A P Pentland. Recursive estimation of motion, structure and focal length. *IEEE Trans on Patt Analysis and Machine Intell*, 17(6):562–575, 1995.

[3] J.O Berger. *Statistical Decision Theory and Bayesian Analysis*. Springer, New York, 1985.

[4] J.S Bonet and P Viola. Roxels: Responsibility weighted 3d volume reconstruction. In *Proc Int Conf on Computer Vision*, 1999.

[5] A Broadhurst, T.W Drummond, and R Cipolla. A probabilistic framework for space carving. In *Proc Int Conf on Computer Vision*, 2001.

[6] N.L Chang and A Zakhor. Constructing a multivalued representation for view synthesis. *Int Journal of Computer Vision*, 2(45):157–190, 2001.

[7] S.B Kang, R Szeliski, and J Chai. Handling occlusions in dense multi-view stereo. In *Proc Conf on Computer Vision and Pattern Recognition*, 2001.

[8] R Koch, M Pollefeys, and L Van Gool. Multi viewpoint stereo from uncalibrated video sequences. In *Proc European Conf on Computer Vision*, 1998.

[9] K.N Kutulakos and S.M Seitz. A theory of shape by space carving. *Int Journal of Computer Vision*, 3(38):199–218, 2000.

[10] J. Shi and C. Tomasi. Good features to track. In *Proc Conf on Computer Vision and Pattern Recognition*, pages 593–600, 1994.

[11] R Szeliski and P Golland. Stereo matching with transparency amd matting. *Int Journal of Computer Vision*, 32(1):45–61, 1999.