Interpolating Novel Views from Image Sequences by 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 a novel approach to view interpolation from image sequences based on probabilistic depth carving. This builds a multivalued representation of depth for novel views consisting of likelihoods of depth samples corresponding to either opaque or free space points. The likelihoods are obtained from iterative probabilistic combination of local disparity estimates about a subset of reference frames. This avoids the difficult problem of correspondence matching across distant views and leads to an explicit representation of occlusion. Novel views are generated by combining pixel values from the reference frames based on estimates of surface points within the likelihood representation. Efficient implementation is achieved using a multiresolution framework. Results of experiments on real image sequences show that the technique is effective.

1 Introduction

There is great interest in developing algorithms to interpolate novel views of a static scene from a set of reference views. Applications range from web based virtual tours to video compression. Previous approaches either make direct use of 3-D representations [9,14,12,5,13,8,6] or employ 3-D information implicitly within image based transformations [15,11,2,10]. Our interest here is in techniques that use calibrated views from image sequences, usually obtained from a structure from motion algorithm, and dense correspondences to determine view centred depth maps [14,12,8]. Novel views are then generated by surface fitting and re-projecting pixel values into the virtual frames. These methods have considerable flexibility in terms of the range of views that can be interpolated and apart from visibility constraints they impose few restrictions on the positioning of the reference views. The difficulties are in obtaining robust dense depth estimates from correspondences and in deriving surface representations which result in convincing novel views. It is well known that establishing dense correspondences is problematic: wide baselines result in mismatches, whilst narrow baselines give low accuracy and increased noise sensitivity. Textureless regions also complicate the matching process. Surface fitting is also problematic, especially around depth discontinuities. Although various global optimisation techniques have been used to improve matters [12,8], the high computational costs involved make them unsuitable when dealing with a large number of reference views.

T. Pajdla and J. Matas (Eds.): ECCV 2004, LNCS 3022, pp. 379–390, 2004.

In this paper we present a new approach to view interpolation which addresses these problems. We borrow ideas from space carving [13] and in particular its recent probabilistic formulations [6,1]. For a given viewpoint, we build a multivalued depth representation in which depth samples along a pixel viewing ray are classified as belonging either to an opaque point or to a free space point. However, unlike space carving, we build our representation not from projected pixel differences, but from disparity (depth) estimates obtained locally about reference frames. Likelihood values for depth classification are then obtained using an iterative combination of these disparity estimates based on explicit models of opacity and occlusion. In this respect we are motivated by the work of Agrawal and Davis [1] and Szeliski and Golland [17]. We call this process *probabilistic depth carving*. In essence, the local depth estimates are 'triangulated' in 3-D space in order to carve away free space points. This yields more reliable and accurate depth representations than the individual local estimates, while at the same time avoiding the problem of establishing correspondences between wide baseline views. It also provides a coherent framework within which to combine widely disparate views of a scene. Moreover, importantly for view interpolation, view centred depth representations can easily be constructed for virtual frames from those derived for the reference frames, leading to a straightforward process for generating novel views which avoids surface fitting.

The principles of depth carving were set out in [18]. In this paper we concentrate on its use for view interpolation and describe an efficient multiresolution implementation. The next two sections provide an overview of the basic ideas and the theoretical framework. The probabilistic and multiresolution implementation are then described in Sections 4-6 and Section 7 describes the view interpolation process. Examples are presented for two real image sequences.

2 Depth Carving

Consider the two examples illustrated in Fig. 1. In each case a novel view is being generated in frame V based on local correspondence matching about three reference frames A, B and C. The scene in Fig. 1a consists of a smooth surface, primarily textureless, but with a central textured region indicated by the zig-zag line. Pixel viewing rays from the reference frames are indicated as $a1$-$c1$ and $a2$-$c2$ and along each ray we have indicated in bold the range of depths having high matching values in the local correspondence analysis. Thus rays $a1$ and $c1$ have a wide range of likely depths due to the lack of texture on the surface, whilst rays $b1$ and $a2$-$c2$ have a narrow range since they intersect the textured region (we assume for simplicity that the autocorrelation function of the texture has a symmetric narrow peak about zero).

In depth carving we seek to classify sample points along viewing rays as either being opaque or in free space. This is achieved by combining the depth likelihoods obtained from the local correspondence matching using 'triangulation'. For example, in Fig. 1a, point P along the novel viewing ray $v1$ is in free space, although due to the textureless surface this cannot be determined from

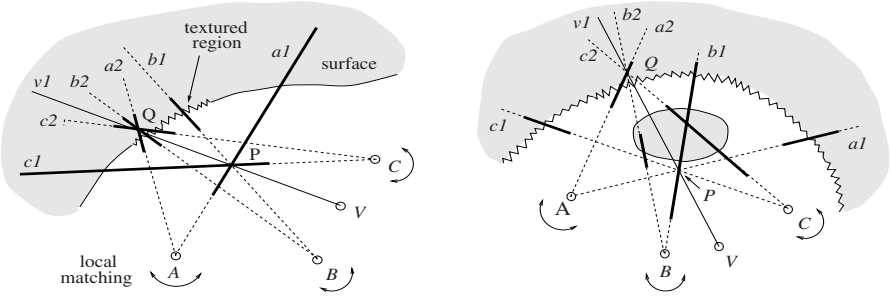


Fig. 1. Depth carving combines matching similarities obtained from local correspondences to obtain improved depth representations.

views A and C; the depth is ill-defined along the rays $a1$ and $c1$. However this can be resolved using view B: ray $b1$ intersects the textured region and thus has well defined depth beyond point P, enabling the latter to be correctly classified as free space. In effect, ray $b1$ carves away the free space in front of the textured surface. This illustrates a useful property: reliable depth estimates obtained about certain reference views due to the presence of textured regions can be used to sort out ambiguous estimates elsewhere. In contrast, the classification of point Q along $v1$ is straightforward since it lies within the textured region and is visible in all three views, giving a high consensus for opacity amongst rays $a2$-$c2$.

Visibility is important in depth carving. Because of possible occlusion it is not sufficient to simply combine the local depth likelihoods directly to classify points along viewing rays. This can be seen from the example in Fig. 1b, which consists of a textured background surface and a textureless foreground object. Point Q in this figure is opaque and this is supported by the depth likelihoods along the ray $a2$. However the point is occluded from views B and C and consequently the depth likelihoods along rays $b2$ and $c2$ correspond to the occluding front surface. Thus if point Q is to be correctly classified this occlusion needs to be accounted for; in effect, cancelling out the free space classification implied by rays $b2$ and $c2$. As discussed below, this can achieved by incorporating explicit models of opacity and occlusion into the carving process. The example in Fig. 1b also further illustrates the advantages of the carving process. Point P lies in free space just in front of the foreground object. Rays $a1$ and $c1$ intersect textured regions on the background and hence carve away point P, removing the ambiguity along ray $b1$ caused by the lack of texture on the foreground object. Repeating this for other viewing rays will carve away the free space around the object. Thus, in principle and with enough reference views, the 3-D volume occupied by the object can be determined despite the lack of texture on its surface.

The key point about depth carving is therefore its ability to combine local depth likelihoods from disparate views, enabling well-defined depths to correct ambiguities in other views. Notably, since it is not based on forming correspondences across all the reference views, it also allows disparate views to be com-

bined without requiring them to contain common elements, a property often absent from other approaches. In the next section we define the explicit models of opacity and occlusion which are needed to implement depth carving.

3 Visible Opacity

The implementation of depth carving is based on two properties of points in a 3-D scene - opacity and visibility - and on a relationship between them which we call the visible-opacity constraint. These are defined as follows.

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

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

where an 'interior point' is a point within an object. This is a restricted form of the opacity used by Bonet and Viola [5]; unlike them we do not consider the case of transparent objects being in the scene for which $0 < \alpha(\mathbf{X}) < 1$. Note that the opacity for surface points is not defined since they occur at the transition between free space and opacity.

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

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

where all visible points are free space points and occluded points can be interior as well as free space or surface points. Note that in general $\upsilon(k, \mathbf{X})$ need not equal $\upsilon(j, \mathbf{X})$, $k \neq j$, and also that $\upsilon(k, \mathbf{X})$ is directly related to the $\alpha(\mathbf{X})$ along the same viewing ray from frame k, ie

$$\upsilon(k, \mathbf{X}) = \begin{cases} 1 \text{ if } \alpha(a\mathbf{X} + (1-a)\mathbf{P}_k) = 0 \quad \forall \quad 0 < a < 1 \\ 0 \text{ otherwise} \end{cases} \tag{3}$$

where $\mathbf{P}_k$ denotes the centre of projection (COP) for frame k. Thus, $\mathbf{X}$ is only visible in frame k if all points preceding it along the viewing ray lie in free space and the visibility of the closest surface point is not defined - the surface boundary corresponds to the transition between visibility and occlusion.

Visible Opacity. Given the above two properties, given an opaque scene and the condition that *every* free space point is visible in *at least one* frame, the following relationship between opacity and visibility can be derived [18]

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

where the product is over all the frames. This is the *visible-opacity constraint* and it plays a central role in the depth carving process. The classification of opaque

and free space points follows directly from the constraint (Fig. 1): a point is classified as opaque if it is occluded from all the reference frames; and classified as free space if it is visible in at least one reference frame. Of course in practice we cannot realistically expect to be able to view every free space point. However, the only consequence is that those points will be mis-classified as opaque (they will not be carved away), which for view interpolation is not so critical - we would not expect to be able to interpolate parts of a scene that are occluded from all of the reference frames.

The above constraint therefore provides a means of determining opacity from visibility. Although we do not have direct access to the latter, we can however obtain initial approximations using the local correspondence matching about the reference frames. These provide an indication of likely visible surface points along viewing rays, albeit with a degree of ambiguity as discussed earlier, and hence can be used to determine initial visibility estimates. The above relationships between opacity and visibility then provide a means of improving the estimates via 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 from the relationship in eqn (3), and so on, until convergence. 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.

The method has similarities with the space carving algorithm described by Agrawal and Davis [1]. However, they use different constraints based on the likelihood of a visible surface being at a given point, rather than the opacity and visibility properties used in our formulation. Consequently they need to use temporal selection of frames to avoid valid estimates being carved away due to occlusion. In contrast, our approach allows all the frames to be combined for each point, which is simpler and gives increased potential for refinement.

4 Probabilistic Depth Carving

The opacity and visibility values are limited to 0 or 1 and so we treat their estimation as a binary decision problem and obtain solutions using a Bayesian probabilistic formulation. Specifically, we use iterative refinement to update the probability that a point $\mathbf{X}$ 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 [4] in which previous estimates of the probability form the prior and the likelihood is given by a constraint support measure, ie for $n > 0$

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

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

$$R_n \;=\; \prod_k [1 - P_n(\upsilon_k = 1)] \;=\; \prod_k P_n(\upsilon_k = 0) \qquad n > 0 \tag{6}$$

where $\upsilon_k \equiv \upsilon(k, \mathbf{X})$ and the product is over all of the reference frames. The iterative process in eqn (5) therefore promotes depths which support the visible-opacity constraint and penalises those that do not.

The above iterative process requires an expression for the probability that a point is occluded in a given frame, ie $P_n(\upsilon_k = 0)$, and suitable initial conditions. Given probabilities for the opacity at the nth iteration we base the former on eqn (3): along a viewing ray, points beyond a point with a high probability of being opaque should have high probabilities of being occluded. Hence we use the following expression for the probability of occlusion

$$P_n(\upsilon_k = 0) \;=\; [1 - P_n(\alpha = 1)]f(\mathbf{X}) \;+\; P_n(\alpha = 1) \qquad n > 0 \tag{7}$$

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

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

where Λ and Λ_i are the set of opacity probabilities for all points along the ray and all points in front of $\mathbf{X}$, respectively. Thus, $P_n(\upsilon_k = 0) \rightarrow 1$ if $\mathbf{X}$ is preceded by a point with a relatively high probability of being opaque and tend to $P_n(\alpha = 1)$ 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.

The required initial conditions are $P_0(\alpha = 1)$ and $P_0(\upsilon_k = 0)$ along each viewing ray for each reference frame. We set $P_0(\alpha = 1) = 0.5$ for all points, ie points are equally likely to be opaque or in free space, and obtain the initial probabilities for occlusion using eqn (7), replacing the opacity probabilities with similarities derived from local correspondence matching about each reference frame as discussed in the previous section. For a point $\mathbf{X}$, we obtain a matching similarity $m(k, \mathbf{X})$ with respect to frame k based on local correspondences within a set of nearby frames Γ_k as follows

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

where $med\{\}$ denotes the median operation and $I_k(\mathbf{x}_k)$ is the intensity value in frame k at point $\mathbf{x}_k$ corresponding to the projection of $\mathbf{X}$ (in practice we interpolate the intensity from surrounding pixels). We use the median of the squared intensity differences here to give a degree of robustness and σ_m is set to reflect the expected variance between nearby frames. This completes the description of the depth carving algorithm; a summary is given in Fig. 2.

5 Multiresolution Implementation

In practice we need to implement a discrete version of the above process, based on a finite number of discrete 3-D points and using discrete reference frames. An efficient means of doing this is to maintain a separate opacity function for each reference frame along with its visibility function, ie associated with a given

(Carry out each step for each 3-D point. Initialise $n = 0$.)

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

Fig. 2. The probabilistic depth carving algorithm.

frame k are opacity values $\alpha(k, \mathbf{X}_i)$ and visibility values $\upsilon(k, \mathbf{X}_i)$, where the discrete 3-D points $\mathbf{X}_i$ are defined along each of the pixel viewing rays. This enables efficient updating of the occlusion probabilities using the opacities along the viewing rays as required in step 4 of Fig. 2. The combination of visibility values to update the opacity in step 3 is then achieved by interpolating amongst the closest 3-D points within the visibility function of each reference frame. Following depth carving, the result is therefore a set of opacity and visibility functions with respect to each reference frame.

We also use a multiresolution implementation to improve computational efficiency by directing the depth sampling towards significant areas such as in the vicinity of surface boundaries. This is done using a course-to-fine focusing strategy. We first generate levels of a Gaussian pyramid for each frame [7]. Starting at the coarse resolution level, we compute matching similarities for each reference frame and each viewing ray at a small number of discrete points within a pre-determined depth range as in eqn (9). Depth carving then proceeds as detailed in Fig. 2 for each 3-D point. The procedure is then repeated for the next higher resolution level of the pyramid using an increased number of sample points. This is achieved by defining a set of 'child points' along the viewing rays corresponding to the 4 child nodes in the Gaussian pyramid for each 'parent point' in the previous level, with the depths of the child points evenly distributed about that of the parent. Carving then proceeds for the child points but excluding those for which consistent opacity classification was achieved at the parent level. The opacity for these points is set to that of the parent point. We define a parent point to be consistent if its opacity classification matches that of all neighbouring points along adjacent pixel viewing rays. In other words, we propagate down consistent opacity values, hence avoiding depth carving within the corresponding depth ranges at the next level. This course-to-fine focusing then continues until the highest resolution level is reached.

6 Depth Carving Examples

Examples illustrating the depth carving process are shown in Figs. 3 and 4. Reference frames from two sequences are shown in Figs. 3a and 4a: the first is a

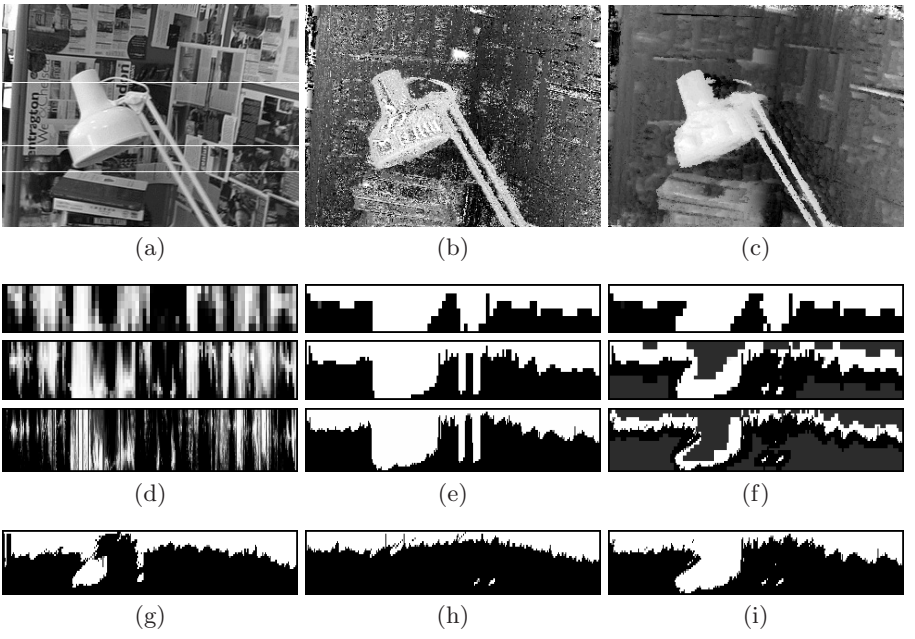


Fig. 3. Multiresolution depth carving for a reference frame in the desk sequence.

desk scene with 76 frames in which the camera translates and rotates horizontally whilst keeping the lamp roughly in centre view; and the second is a garden scene with 161 frames in which the camera moves around an ornament by approximately 180 degrees whilst keeping it roughly in centre view. Both sequences contain significant occlusion and the garden sequence contains widely disparate views with very few common features across all the frames. We calibrated the sequences using the recursive structure from motion algorithm described in [3] to obtain relative metric 3-D camera positions for each frame and sparse depth estimates corresponding to tracked feature points. The latter were obtained using the KLT tracker [16]. For the garden sequence we processed the frames in three sections due to features moving out of view.

The results of the multiresolution depth carving process for the desk sequence can be seen in Figs. 3d-i. We used 16 reference frames each with 20 neighbouring frames to obtain the local matching similarities. We set the matching and visibility parameters σ_m and σ_f as described in [18]. The plots in Fig. 3d show the matching similarities obtained along the middle scanline shown in Fig. 3a at the sample depths on each level of the Gaussian pyramid, where the closest depth is at the bottom of the plot. Figures 3e-f show the visibility and opacity values respectively for the same scanline and depths following convergence of the carving process. The shaded areas in the opacity plots indicate that opacity values are propagated down from the previous level and Fig. 3i shows the final opacity values obtained by combining the levels. Final opacity values for the top and bottom scanlines are shown in Figs. 3g and 3h, respectively. Fig. 3c shows

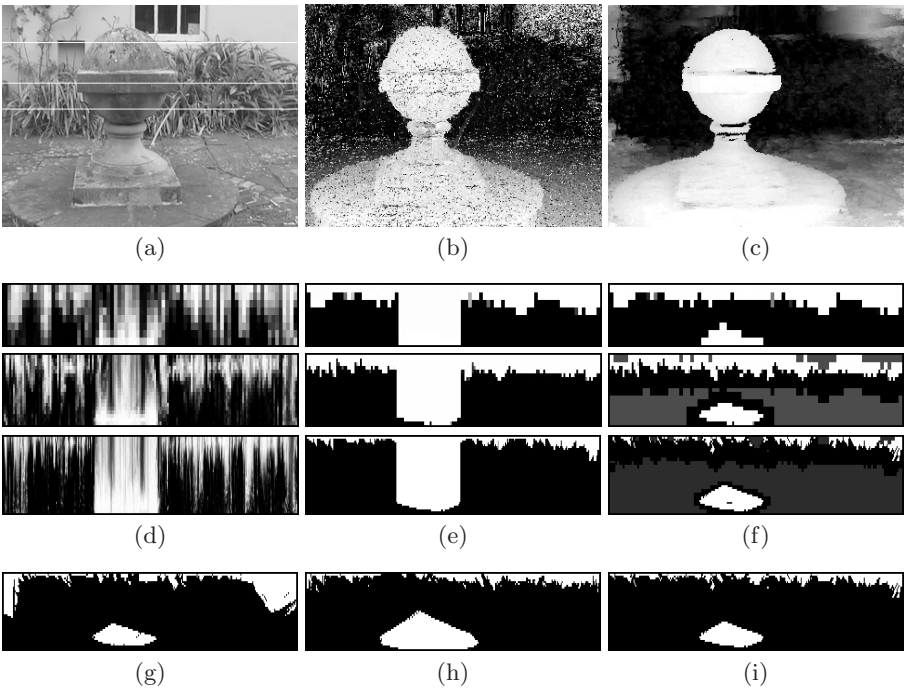
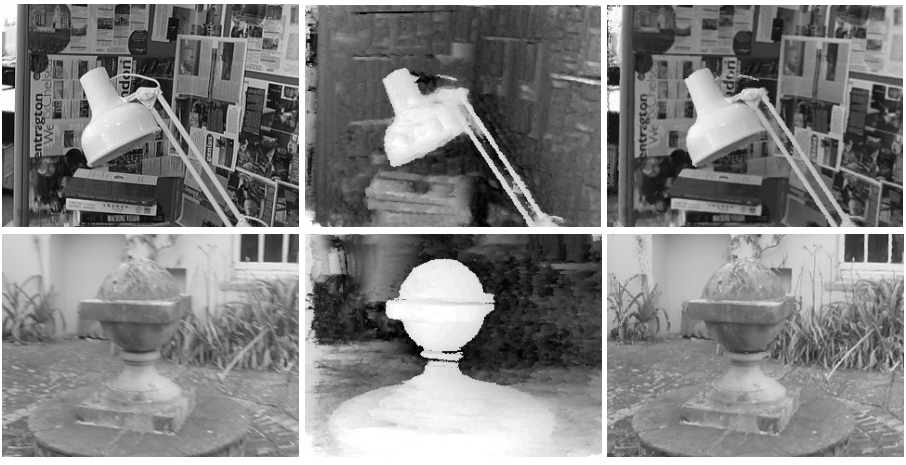


Fig. 4. Multiresolution depth carving for a reference frame in the garden sequence; (d-f) and (i) refer to the bottom scanline in (a) and (g,h) refer to the top and middle scanlines, respectively.

the closest depth with opacity probability above a given threshold, ie corresponding to the visible surface points in the frame. For comparison, Fig. 3b shows the depths with the highest local matching similarity. A similar set of results for the garden sequence are given in Fig. 4, where we used 15 reference frames each with 20 neighbouring frames.

The main observation from these results is that despite the high level of ambiguity in the depth estimates obtained from the local correspondence matching (see Figs. 3b,d and 4b,d), the depth carving process correctly determines the opacity classification for the significant parts of the scene. For example, in the desk sequence, large portions of the free space surrounding the lamp shade and stand have been successfully carved away, leaving only those parts not visible in any of the reference frames. Similar comments apply to the results for the garden sequence, with the free space around the ornament being carved away. Note in particular the correct isolation of the lamp stand and cable in the desk sequence and the curved surface carved away for the front of the lamp shade and the ornament. The latter is not apparent in the local depth estimates and this provides a good illustration of how depth carving can combine disparate views to improve estimates of scene structure. Note also from Figs. 3f and 4f how the multiresolution implementation successfully focuses the depth sampling at each level around the key surface areas, thus helping to minimise computational cost.



(a) (b) (c)

Fig. 5. View interpolation for intermediate frames in the original sequence: (a) original frame; (b) reconstructed visible surface depths; (c) interpolated view.

Less satisfactory, however, is the amount of noise present in the opacity estimates for the background in the two sequences. This is caused by false foreground depths detected by the initial local correspondences due to textureless regions which are not carved away when combined with other views. However, as noted earlier, in textureless regions such errors are less critical for view interpolation.

7 View Interpolation

Having determined opacity and visibility representations for each reference frame, it is straightforward to generate corresponding functions for virtual frames using the visible-opacity constraint in eqn (4). For each sample point along each viewing ray in the virtual frame we interpolate visibility values $v(k, \mathbf{X})$ with respect to each reference frame and then combine them using eqn (4) to give the opacity in the virtual frame. This becomes a computationally inexpensive binary operation if we first make a maximum likelihood assignment of opacity values for the reference frames and is one of the key advantages of the depth carving framework. An example is shown in Fig. 5b which shows the visible surface depth estimates for an intermediate frame in the original sequence between two reference frames. The essential scene structure has clearly been maintained within this reconstructed view. Further examples for virtual frames away from the original camera trajectories are shown in Fig. 6a.

Novel views can be generated from the visible surface depth maps using reprojection. Thus, for pixels in a virtual frame, we collect pixel values from the reference frames, excluding those frames from which the point in question is occluded. These values are then processed to determine the likely value in the new view. This raises the issue of how best to combine pixel values to produce

(a) (b) (c)

Fig. 6. Interpolated novel views: (a) reconstructed visible surface depths; (b) interpolated novel views using depth maps in (a); (c) other interpolated views.

realistic views [9,10]. Here, we opted for two simple combination strategies: the median of the collected pixel values; or using the pixel value from the closest reference frame. The former proved marginally better for the desk sequence since it gives a degree of robust averaging which results in better consistency. This works because of the limited variation in lighting effects between the different views. In contrast there are significant light changes in the garden sequence and in this case taking pixel values from the closest (non-occluded) reference frame proved more effective. Figure 5c shows reconstructed versions of the intermediate original frames shown in Fig. 5a based on the visible surface depth maps shown in Fig. 5b. Given the simplicity of the pixel combination algorithm, these results are good and they illustrate clearly the validity of the opacity representation derived from the depth carving. Similar comments apply to the novel views generated away from the original camera paths as shown in Fig. 6, although there are some small ghosting effects and holes appearing in garden sequence, especially around the base of the ornament, due to errors in the depth maps.

8 Conclusions

The view interpolation algorithm presented here has several key benefits. It enables the generation of view-centred depth representations without having to establish dense correspondence between reference views. This is useful for sequences containing widely disparate views. The visible opacity model allows the generation of depth representations and novel views without recourse to explicit surface fitting. Significantly, the depth carving allows ambiguities in depth estimates obtained in one view to be resolved using more reliable estimates obtained from different views and relating to different parts of the scene, for example within textured areas. We are not aware of this technique being used before and our

results suggest that it has considerable potential. We are currently investigating the use of local spatial and depth constraints and more sophisticated methods for combining pixels when generating interpolated views.

Acknowledgement. The authors are grateful to the Independent Television Commission, UK, for financial assistance.

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. S Avidan and A Shashua. Novel view synthesis in tensor space. In *Proc Conf on Computer Vision and Pattern Recognition*, 1997.
3. A Azarbayejani and A P Pentland. Recursive estimation of motion, structure and focal length. *IEEE Trans on Pattern Analysis and Machine Intelligence*, 17(6):562–575, 1995.
4. J.O Berger. *Statistical Decision Theory and Bayesian Analysis.* Springer, New York, 1985.
5. J.S Bonet and P Viola. Roxels: Responsibility weighted 3d volume reconstruction. In *Proc Int Conf on Computer Vision*, 1999.
6. A Broadhurst, T.W Drummond, and R Cipolla. A probabilistic framework for space carving. In *Proc Int Conf on Computer Vision*, 2001.
7. P.J Burt and E.H Adelson. The laplacian pyramid as a compact image code. *IEEE Transactions on Communications*, 31(4):532–540, 1983.
8. N.L Chang and A Zakhor. Constructing a multivalued representation for view synthesis. *Int Journal of Computer Vision*, 2(45):157–190, 2001.
9. P.E Debevec, C.J Taylor, and J Malik. Modeling and rendering architecture from photographs: a hybrid geometry- and image-based approach. In *Proc ACM SIGGRAPH*, 1996.
10. A Fitzgibbon, Y Wexler, and A Zisserman. Image-based rendering using image-based priors. In *Proc Int Conf on Computer Vision*, 2003.
11. S.J Gortler, R Grzeszczuk, R Szeliski, and M.F Cohen. The lumigraph. In *Proc ACM SIGGRAPH*, 1996.
12. R Koch, M Pollefeys, and L Van Gool. Multi viewpoint stereo from uncalibrated video sequences. In *Proc European Conf on Computer Vision*, 1998.
13. K.N Kutulakos and S.M Seitz. A theory of shape by space carving. *Int Journal of Computer Vision*, 3(38):199–218, 2000.
14. P.J Narayanan, P.W Rander, and T. Kanade. Constructing virtual worlds using dense stereo. In *Proc Int Conf on Computer Vision*, 1998.
15. S Seitz and C.R Dyer. Physically-valid view synthesis by image interpolation. In *Proc IEEE Workshop on Representation of Visual Scenes*, 1995.
16. J Shi and C Tomasi. Good features to track. In *Proc Conf on Computer Vision and Pattern Recognition*, 1994.
17. R Szeliski and P Golland. Stereo matching with transparency amd matting. *Int Journal of Computer Vision*, 32(1):45–61, 1999.
18. A Yao and A Calway. Dense 3-d structure from image sequences using probabilistic depth carving. In *Proc British Machine Vision Conference*, pages 211–220, 2003.