??, ??, 1–23 (??)

Efficient Dense Stereo with Occlusions for New View-Synthesis by Four-State Dynamic Programming

A. CRIMINISI, J. SHOTTON, A. BLAKE, C. ROTHER, P.H.S. TORR

HTTP://RESEARCH.MICROSOFT.COM/VISION/CAMBRIDGE/I2I

Received ??; Revised ??

Abstract.

A new algorithm is proposed for efficient stereo and novel view synthesis. Given the video streams acquired by two synchronized cameras the proposed algorithm synthesises images from a virtual camera in arbitrary position near the physical cameras. The new technique is based on an improved, dynamic-programming, stereo algorithm for efficient novel view generation. The two main contributions of this paper are: i) a new four state matching graph for dense stereo dynamic programming, that supports accurate occlusion labelling; ii) a compact geometric derivation for novel view synthesis by *direct* projection of the minimum cost surface. Furthermore, the paper presents an algorithm for the temporal maintenance of a background model to enhance the rendering of occlusions and reduce temporal artefacts (flicker); and a cost aggregation algorithm that acts directly in the three-dimensional matching cost space.

The proposed algorithm has been designed to work with input images with large disparity range, a common practical situation. The enhanced occlusion handling capabilities of the new dynamic programming algorithm are evaluated against those of the most powerful state-of-the-art dynamic programming and graph-cut techniques. The accuracy of disparities is also evaluated against the standard Middlebury error metrics. A number of examples demonstrate the robustness of the algorithm to artefacts in stereo video streams. This includes demonstrations of cyclopean view synthesis in extended conversational sequences, synthesis from a freely translating virtual camera and, finally, basic 3D scene editing.

Keywords: Dense stereo, image-based rendering, video-conferencing, gaze correction.

1. Introduction

This paper addresses the problem of novel-view synthesis from a pair of rectified video streams with specific emphasis on gaze correction for one-to-one teleconferencing. With the rise of live chat technologies[1], it is envisaged that the PC will increasingly be used for interactive visual communication. One pressing problem is that any camera used to capture images of one of the participants has to be positioned offset from his or her gaze (*cf.* fig. 1 and fig. 2). This can lead to lack of eye contact and hence undesirable consequences for human interaction [GTZ+00].

One might think that if it were possible to drill a hole in the centre of a computer screen and place a camera there, that would achieve the desired viewpoint. The first problem with this solution is that "porous" screens do not exist; but even if they did the user would be required always to look at the centre of the monitor, where the extra camera had been inserted. However in a messaging session the user looks at the communication window (where the other person's face appears) which can be displaced and moved around the screen at will (fig. 2). Therefore, the camera needs to be placed behind the communication window on the screen; but this cannot be achieved with available hardware and therefore a software solution is sought.

Previously proposed approaches can be broadly categorized as *model-based* or *image-based*. One model-based technique is to use a detailed 3D head model,

input left image input right image

Fig. 1. **Failure of eye contact.** In one-to-one videoconferencing, cameras located on the frame of the computer monitor fail to capture gaze correctly. Here a person looks at the centre of the screen but, in the images captured by cameras mounted on either side of the computer monitor, he does not appear to be looking directly ahead. The proposed algorithm is capable of synthesizing a virtual view that procures eye contact.

texture map it and re-project it into the required viewpoints. Whilst this can be successful [Vet98, YZ02], it is limited to imaging heads with no hair or neck. Nor can it deal with occlusion events such as a hand in front of the face. A more general approach, proposed here, is to use image-based rendering techniques (*IBR* [CW93]) to synthesize novel views from two input images. The entire input images, as opposed to the head only, are processed, thus avoiding the detection and modeling of heads with all the associated problems. Though we focus on the gaze correction application, the algorithm developed in this paper is of general nature and can be applied to different IBR scenarios.

Many popular IBR algorithms combine a depth map with input images to produce synthetic images. In order to generate a depth map a dense stereo algorithm is required, a substantial review of which can be found in [SS02], in which the authors evaluate a number of existing dense-stereo techniques. But this evaluation may not be sufficient for our purposes as: (i) the range of disparities considered in [SS02] is smaller than in our application (0-29 pixels there, whereas we typically consider 0-80 pixel disparities); (ii) we are primarily interested in new-view synthesis so it does not matter if the disparities are relatively inaccurate in texture-less image regions; all that matters is that the new view is well synthesized (as noted in [Sch99, Sze99]); (iii) we consider long video sequences so temporal stability is a significant issue.

In the past, research on dense stereo reconstruction has been directed largely towards the accurate recovery of disparity maps, though not entirely [BM92]. We have found that while inaccurate disparities may still

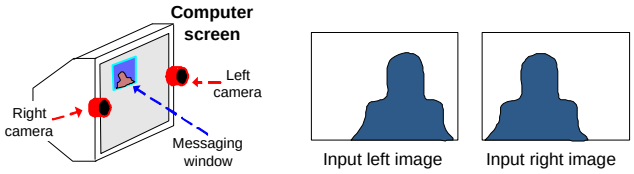


Fig. 2. **The basic teleconferencing setup** considers two cameras placed on the frame of a computer monitor. A window for viewing the remote participant is marked in blue on the computer screen. The algorithm described in this paper achieves a corrected gaze image in an efficient and compelling way.

produce acceptable synthesized images over matched regions, inconsistent occlusion maps lead to unacceptable artefacts. Therefore, in this paper we focus on accurate occlusion modelling and detection.

According to the evaluation in [SS02], two of the most powerful dense stereo techniques use Graph cut (GC) [KZ02] and loopy belief propagation [SSZ02]. However, both of these are currently too computationally intensive for real-time applications and, since near real time performance is one of the goals of this paper, we turned our attention to more efficient algorithms such as Epipolar-line Dynamic Programming [OK85], commonly referred to as DP. The DP algorithm described in [CHRM96] has previously been demonstrated for cyclopean view interpolation [COL93] in video[2]. In the basic form of the DP algorithm, in order to obtain computational efficiency, observations consist of single-pixel intensities. This, together with the fact that pairs of corresponding scanlines are considered independently, introduces a number of artefacts which corrupt the quality of the output reconstruction, especially for large disparity ranges as fig. 3b shows.

In particular, DP-based algorithms for novel view synthesis are characterized by three kinds of artefacts: (i) artefacts produced by mismatches (horizontal streaks due to inconsistencies between adjacent scanlines); (ii) the "halo" in the regions where the background is visible in only one of the two input views (occlusion); and (iii) flickering synthesized pixels, caused by matching ambiguities. The first two kinds of artefacts are static, while the latter is temporal in that it appears when processing sequences of stereo images. This paper sets out to address and solve those kinds of artefacts while maintaining high computational efficiency.

Our new contributions have two aspects: accurate generation of occlusion maps and efficient new-view

a **b** **c**

Fig. 3. **Fast cyclopean view synthesis by conventional DP.** (a,c) Input left and right views, respectively; (b) Cyclopen view synthesized by dynamic-programming [COL93]. Note that gaze is correct in the cyclopean view. The algorithm runs at near real-time rate, but produces significant artefacts in the synthesized cyclopean image.

rendering. For the first we propose a new DP algorithm acting on a *four*-state matching graph. New labels are introduced for occlusions, and the cost function is extended to favour: (a) good grouping of occlusions, (b) formation of solid occlusion regions at the boundaries of foreground objects, and (c) inter-scanline consistency. For the second aspect we introduce minimum-cost surface projection as a compact technique for generating synthetic views from arbitrary virtual cameras, *directly* from the minimum-cost surface obtained during the DP process[3]. This technique avoids the explicit construction of a 3D mesh model or depth map.

Paper outline. Section 2 reviews the state of the art in dense stereo via DP, and consequent issues for novel view rendering, particularly of occluded regions. The main contribution of this paper is described in sections 3 and 4 which introduce our improved multi-state, dense-stereo algorithm. Section 5 illustrates the cost filtering algorithm for inter-scanline consistency. Section 6 presents a comparative evaluation of performance of our technique with respect to disparity estimation and occlusion detection. Realistic synthesis of occluded regions is discussed in section 7 and virtual-view generation and rendering in section 8. Finally, section 9 demonstrates the effectiveness of the proposed techniques with a number of examples where both static images and entire sequences are generated for various virtual camera locations.

2. Background on Dynamic Programming and Novel-view Synthesis

This section reviews the principles of dynamic-programming algorithms for dense stereo [CHRM96, OK85] and discusses issues related to the synthesis of cyclopean images from the two input views.

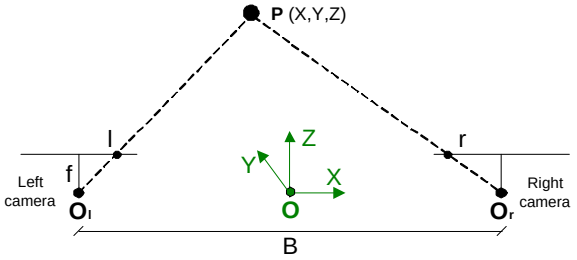


Fig. 4. **Basic camera configuration and notation.** $\mathbf{O}_l$ and $\mathbf{O}_r$ are the optical centres of left and right cameras respectively, f is the focal length of the cameras (assumed identical for both cameras) and B is the baseline between the two optical centres. The origin of the reference coordinate system X, Y, Z is denoted $\mathbf{O}$.

2.1. Conventional dynamic-programming

Figure 4 shows a plan view of the camera setup. The left and right cameras provide us with the synchronized and epipolar-rectified input videos[4] . The focal length is denoted f, and B is the distance between the two optical centres (the baseline). A Cartesian coordinate system is chosen with origin at the mid-point between the left and right optical centres. A 3D scene point $\mathbf{P}$ is projected into the two input image planes in corresponding image points at positions l and r relative to the respective image centres. The distance $d = l - r$ is commonly known as disparity. We refer to the images corresponding to a virtual camera, with optical centre in the origin $\mathbf{O}$, as *cyclopean* images. As will be demonstrated, our algorithm is not restricted to cyclopean views only but is capable of generating virtual images from arbitrary viewpoints.

The diagram in fig. 5a represents the matching graph for a pair of corresponding scanlines in the two input images [OK85, CHRM96]. Note that, since $l >= r \ \ \forall \mathbf{P}$ (*i.e.* disparities $d = l - r$ are always non-negative), then it is only ever necessary to consider the lower half of the matching graph (grey area in fig. 5a). The limiting, zero-disparity case $l = r$ corresponds to points at infinity. The 45-degree line in fig. 5a is termed the "virtual scanline" for reasons that will become obvious in the next section. The local cost of matching a pixel at position l along the left scanline with a pixel at position r along the right scanline is denoted $M(l, r)$. In conventional DP, the cost $M(l, r)$ may be defined simply as the square difference of pixel intensities, though more elaborate measures based on patches, colour, wavelets etc., can be used.

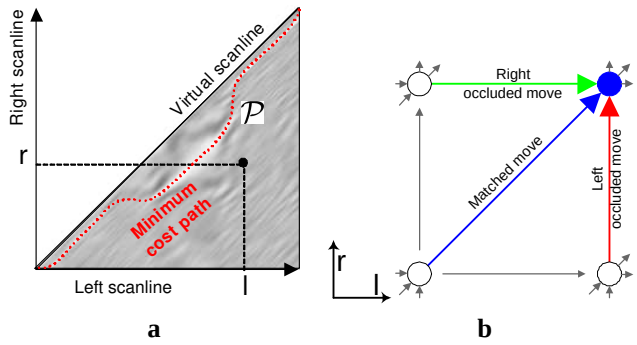


Fig. 5. **Conventional dynamic-programming.** (a) The two-dimensional matching graph on which DP is based. Each node in the planar graph corresponds to a pair of pixels, in the left and right scanlines. A matching cost $M(l,r)$ is associated to each node and the goal is to find a minimum-cost path (shown in red) joining the two opposite corners of the graph. Bright colouring indicates high pixel similarity, *i.e.* low values of $M(l,r)$. (b) A blown-up view of (a) showing the three allowed moves between pixel pairs [COL93]; circles represent nodes of the graph in (a).

Standard 3-move DP. Dynamic-programming consists of two passes: forward and backward [CHRM96]. The forward step constructs a matrix of cumulative matching costs $\mathtt{C}$ by the following recurrence:

$$\mathtt{C}(l,r) = \min \begin{cases} \mathtt{C}(l-1,r) & + \ \beta \\ \mathtt{C}(l-1,r-1) & + \ M(l,r) \\ \mathtt{C}(l,r-1) & + \ \beta \end{cases} \qquad (1)$$

where $\mathtt{C}(l,r)$ indicates the cumulative cost of the path from the point $(0,0)$ to the point (l,r). Note that only three moves are permitted: a horizontal, possibly *occluded*, move, a diagonal *matched* move and a vertical, possibly *occluded* move (fig. 5b). Thus, 45-degree segments in the minimum cost path correspond to fronto-parallel surfaces (constant disparity); vertical and horizontal segments represent either occlusions or non-fronto-parallel surfaces. The cost of a horizontal/vertical move, which may indicate occlusion, is β. When matching costs $M(l,r)$ are normalised so that $0 \leq M(l,r) \leq 1$, a value of $\beta = 0.3$ yields good results on a variety of images. At each iteration the minimum cost between the three possible moves is chosen and a table of backward links is stored for use in the second pass of DP.

The backward pass of the algorithm follows the saved back-links; starting from $(l = W, r = W)$ where W is the image width, to the origin $(l = 0, r = 0)$. This defines the minimum-cost path $\mathcal{P}$ as the sequence of visited nodes.

For each pair of scanlines, given their matching path $\mathcal{P}$:

- For each point $\mathbf{p} \in \mathcal{P}$
  1. take the colours $I^l(l)$ and $I^r(r)$ of the corresponding pixels l and r in the left and right scanlines, respectively;
  2. compute the average value $\tilde{I} = \frac{1}{2}(I^l(l) + I^r(r))$;
  3. project the newly obtained pixel orthogonally to the virtual image scanline, into the virtual image point $\mathbf{v}$; *i.e.* $I^v(v) = \tilde{I}$.

Fig. 6. **Cyclopean view synthesis from direct projection of the minimum-cost path.**

Limitations of conventional DP. The three-move model is limited since it fails to distinguish completely between occluded and non-occluded moves. One of the main contributions of this paper will be to expand the set of permitted moves to support unambiguous detection and classification of occlusion events.

2.2. Direct cyclopean-view synthesis from DP

This introductory section explains how cyclopean views can be generated directly from the minimum-cost paths estimated by conventional DP. Special attention is paid to the synthesis of pixels in occluded regions. The basic cyclopean-view synthesis algorithm is described in fig. 6 and illustrated in fig. 7a.

The algorithm in fig. 6 applies to matched pixels *only* and occluded areas must be treated differently. Indiscriminate application of the algorithm in fig. 6 to occluded and unoccluded points alike, would produce a distorting effect, a "halo" around the foreground objects in the cyclopean view. An example is shown in fig. 8 where the frame of the door and the edge of the whiteboard have been deformed into curves which follow the outline of the foreground head. The "halo" artefact is much more noticeable and disturbing when video sequences are reconstructed in this way.

Fronto-parallel assumption for occlusion filling. In order to overcome the halo effect occluded pixels must first be reliably detected. For those pixels it is necessary to make a plausible assumption about underlying 3D structure since this information is not available given the absence of a stereo match. One effective assumption is that of a fronto-parallel background [Sch99]. As illustrated in fig. 7c, filling of the

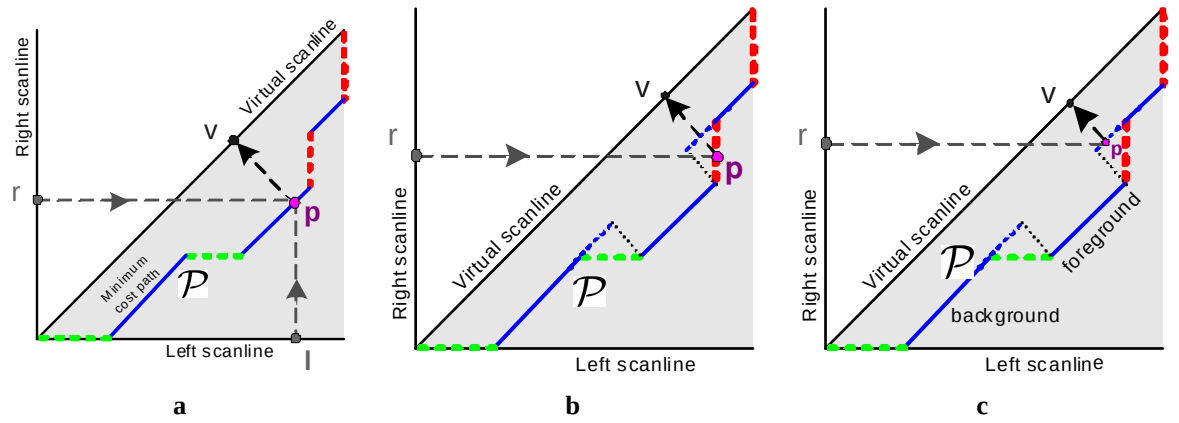


Fig. 7. **Generating the cyclopean view.** (a) A matched point $\mathbf{p} \in \mathcal{P}$ is projected orthogonally onto its corresponding point v on the virtual scanline. The luminance value of the virtual pixel v is the average of the corresponding pixels l and r on left and right images, respectively. (b) *Halo:* treating the occluded segments in $\mathcal{P}$ in the same way as the matched segments produces a lens-like effect that we call the "halo" artefact (fig. 8). (c) *Fronto-parallel occlusion synthesis:* the halo effect is largely removed if a fronto-parallel background assumption is made: an occluded point $\mathbf{p}$ on the continuation of the background is projected orthogonally onto its corresponding point v on the virtual scanline.

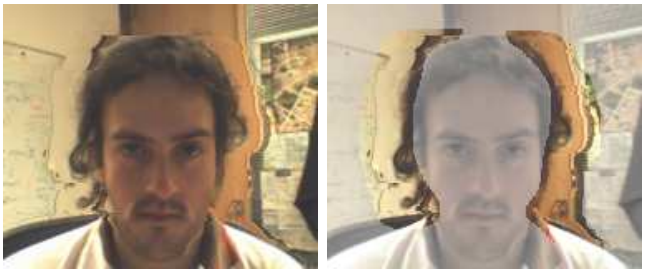


Fig. 8. **Halo artefact.** (a) A cyclopean view reconstructed by applying the algorithm in fig. 6 to both matched and occluded segments of the recovered minimum-cost path. A "halo" of deformed background objects is visible around the head. (b) Regions over which the halo effect occurs are highlighted.

occluded regions can be achieved under the fronto-parallel assumption by extending the background at constant disparity. Fig. 7c shows how, for a left occlusion (vertical dashed line), the values of the virtual pixels are taken *only* from the right image: $I^v(v) = I^r(r)$, and vice-versa.

The reset artefact. The fronto-parallel approximation can be applied only if occluded regions are correctly detected. Detection errors (fig. 9b) cause the sampling of "source" pixel values from incorrect locations in the input images — the *reset* artefact (*cf.* fig. 22). Accurate detection of occlusion is clearly paramount.

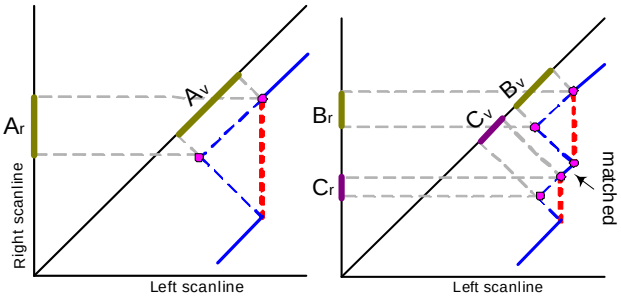


Fig. 9. **The reset effect.** (a) The fronto-parallel approximation used for filling occluded regions. For the left occlusion marked in red (dashed), the region A_v in the cyclopean image is copied from the corresponding region A_r in the right image. See also fig. 7c. (b) A small error in the detection of the occluded region, *e.g.* a small matched region inside a large occlusion, produces a large error in the cyclopean synthesized scanline. In fact, the "source" regions B_r and C_r are quite different from A_r and far apart from each other. This produces visible artefacts as illustrated later.

The next two sections introduce our improved DP algorithm for accurate occlusion detection.

3. Extending the set of basic moves for unambiguous occlusion modelling

In the standard DP approach slanted (*i.e.* non fronto-parallel) surfaces in space are modelled as a combi-

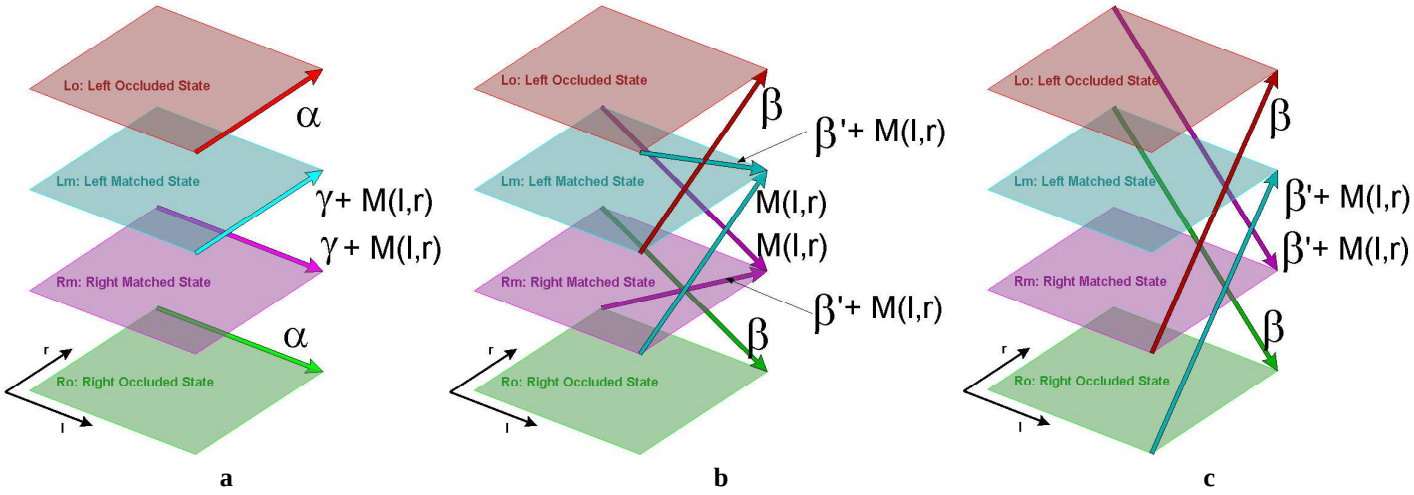


Fig. 10. **The proposed four-state model for DP.** The graph associated with our new DP algorithm occupies *four* planes, with 14 allowed state transitions: (a) state-preserving transitions and (b,c) between-state transitions. Each permitted state transition (shown by arrows) has been labelled with the associated cost — see text.

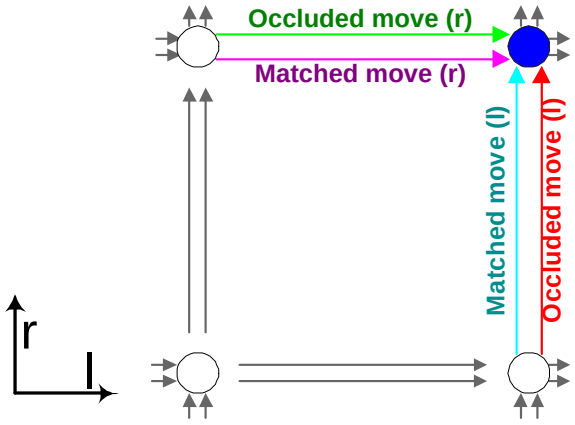


Fig. 11. **The four-move model for Dynamic Programming** allows two matched moves (marked in magenta and cyan), and two occluded moves (green and red).

nation of diagonal and horizontal or vertical moves in the matching graph. In order to disambiguate between horizontal/vertical *matched* moves and true occlusion events, the new model has two types of horizontal moves and two types of vertical moves, matched and occluded. Since a line at any orientation can always be approximated by a sequence of horizontal and vertical matched moves, the diagonal matched move of the basic DP model is eliminated without loss. This defines the *four-move model* of fig. 11, to be compared with fig. 5b.

In recent work [CSBT03] we have tried the five-move model, including a matched diagonal move, as suggested also by Ishikawa *et al.* [HD98]; but we have found the four-move model as reliable as the five-move model and simpler. In the four-move model, every possible path through the cost space has equal length (Manhattan distance) so that the costs of alternative paths are truly comparable. Finally, the four move model lends itself to a proper statistical interpretation [KCB$^+$05b].

4. Imposing constraints on occlusions by DP on a *four-state* graph

Thanks to the four-move model, matches and occlusions are now unambiguously labelled here, unlike the conventional three-move model. This section describes another evolution of our DP model which imposes prior constraints on runs of occluded and matched pixels. This is achieved by a four-state matching graph with two occluded states L_0 (occluded in left image) and R_0 (occluded in right image) and two matched states L_m (left-matched) and R_m (right-matched) (see fig. 10 and fig. 13). In contrast, conventional DP runs on a single planar graph.

The four-state model reflects naturally the persistence of each of the states. For instance long runs of occlusions can be favoured by setting a high cost for entering or leaving an occluded state (L_o or R_o). Similarly, it is desirable to bias *against* runs of matched moves in R_m or L_m, ensuring that surfaces close to

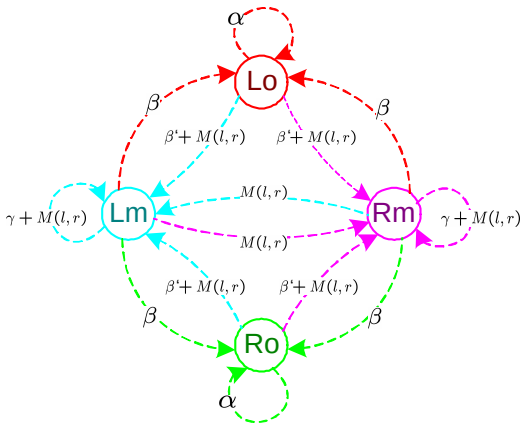


Fig. 12. **Finite State Machine Representation.** Our 4-state algorithm can be represented as a finite state machine. The four states correspond to the basic four permitted matched and occluded moves. The edge labels represent the costs associated to the 14 different transitions between states.

fronto-parallel are preferred, as in the conventional DP. Slanted surfaces are thus described by oscillations of the optimal path between the L_m and R_m states (fig. 13). The 4-states framework includes four different cumulative cost matrices: $\mathtt{C}_{Lo}$, $\mathtt{C}_{Ro}$, $\mathtt{C}_{Lm}$ and $\mathtt{C}_{Rm}$, one for each state in the graph. The elements of the cumulative cost matrices are initialised to $+\infty$ everywhere except in one row of the right occluded plane, where:

$$\mathtt{C}_{Ro}[i,0] = i\alpha \quad \forall i = 0 \dots W-1 \ . \tag{2}$$

The forward step of 4-state DP computes the four cumulative cost matrices according to the following recursion:

$$\mathtt{C}_{Lo}[l,r] = \min \begin{cases} \mathtt{C}_{Lo}[l,r-1] & + & \alpha \\ \mathtt{C}_{Lm}[l,r-1] & + & \beta \\ \mathtt{C}_{Rm}[l,r-1] & + & \beta \end{cases} \tag{3}$$

$$\mathtt{C}_{Lm}[l,r] \ = M(l,r) + \min \begin{cases} \mathtt{C}_{Lo}[l,r-1] & + & \beta' \\ \mathtt{C}_{Lm}[l,r-1] & + & \gamma \\ \mathtt{C}_{Rm}[l,r-1] & & \\ \mathtt{C}_{Ro}[l,r-1] & + & \beta' \end{cases}$$

where $M(l,r)$ is the cost of matching the l^{th} pixel in the left scanline with the r^{th} pixel in the right scanline. In this section we are assuming given the matching costs $M(l,r)$ and focus on the DP algorithm only. Section 5 will describe how the cost function is computed.

The two other cost matrices $\mathtt{C}_{Ro}[l,r]$ and $\mathtt{C}_{Rm}[l,r]$ are defined by invoking symmetry on the definitions of $\mathtt{C}_{Lo}$ and $\mathtt{C}_{Lm}$ above. Note that there are 14 allowed state transitions, as illustrated in fig. 10. The cost structure defined by the four-state DP algorithm and the related state transitions can be represented compactly as a finite state machine as in figure 12.

In the forward pass, the computation of the four cumulative cost matrices proceeds from the corner $(l = 0, r = 0)$ in the left occluded state (L_o) and continues up to $(l = W-1, r = W-1)$ in the right occluded state (R_o), where W is the image width. At each iteration, as the cumulative costs matrices are built, backward pointers to the nodes with minimum cumulative cost are stored, similarly to conventional DP. In the backward pass, the minimum-cost path is recovered by following the fourteen different kinds of back-pointers from the $(l = W-1, r = W-1)$ corner on the left occluded state L_o to $(l = 0, r = 0)$ on the R_o state.

Setting the transition costs. The penalty parameters α, β, β' and γ are chosen as follows:

- The parameter α is set to $1/2$, a value chosen just sufficient to exceed the typical cost $M(l,r)$ ($0 \leq M(l,r) \leq 1$) of a good match.
- The penalty cost β is set to 1.0 – large enough to avoid erroneous labelling of weak true matches as occlusions, but not so large as to prevent the minimum cost path ever entering an occluded state.
- The parameter β' is set to 1.0 – large enough to avoid reset artefacts (leaving an occluded state too soon), but not so large as to prevent the minimum cost path ever entering a matched state.
- The cost γ is set to $1/4$ to bias *against* runs of transitions within the same *matched* state. Clearly we do not want to disallow these transitions as these are used to approximate slanted surfaces, but it is envisioned that in most cases, the minimum cost path will oscillate between the left and right matched states, approximating a roughly fronto-parallel surface in a stair-step fashion (fig. 13d).

It can be proven that the optimal path solution depends only on the sum $\beta + \beta'$. Therefore, we can set $\beta = \beta'$ without loss of generality, thus reducing the number of parameters to three. Optimal values for all these parameters may be learnt from input ground-truth data [KCB+05b]. Sensitivity of our algorithm with respect to its parameters is discussed in sec. 6.5.

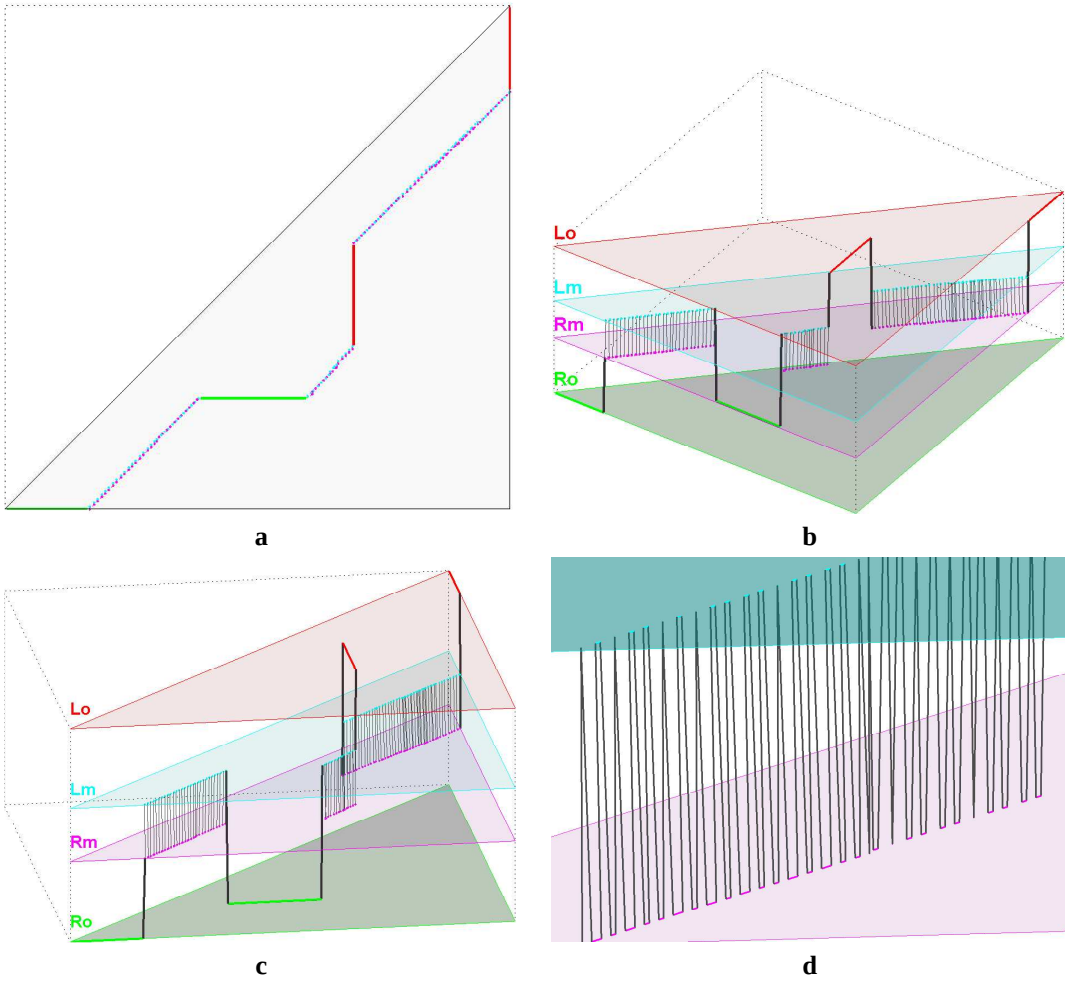


Fig. 13. **Minimum-cost 3D path in the four-state graph for DP.** (a,b,c) Different views of the 3D minimum-cost path estimated for a pair of input scanlines. The rapid oscillations between the two matched states (L_m and R_m) illustrate the way matched, slanted surfaces are represented as alternate horizontal and vertical *matched* moves. (d) A detail from (c) highlighting the matched oscillations.

Figure 13 shows an example of the recovered minimum-cost 3D path for a pair of corresponding scanlines extracted from real stereo images. The 3D minimum cost path resulting from the application of our DP algorithm weaves its way through the four states of the graph. Note the two large occlusions (red and green segments) lying on the corresponding occluded states. As expected, slanted surfaces are tracked as series of oscillations between the two matched states.

Next, we discuss the details of the cost function construction and cost aggregation.

5. Matching cost definition and aggregation

This section describes the computation of matching costs between pixels pairs and their aggregation to improve inter-scanline consistency.

Computation of matching costs. The use of neighbourhood windows in computing the cost of matching two pixels has already been shown to help reduce streaky artefacts [SS02]. The matching cost $M(l, r)$ we employ in this paper is calculated for every pair of pixels along corresponding epipolar lines with a windowed Normalised Sum of Squared Differences

(NSSD), defined as:

$$M(l,r) = \frac{1}{2}\frac{\sum_{\boldsymbol{\delta}\in\Omega}\left[(I^l_{\mathbf{p}_l+\boldsymbol{\delta}} - \overline{I^l}_{\mathbf{p}_l}) - (I^r_{\mathbf{p}_r+\boldsymbol{\delta}} - \overline{I^r}_{\mathbf{p}_r})\right]^2}{\sum_{\boldsymbol{\delta}\in\Omega}(I^l_{\mathbf{p}_l+\boldsymbol{\delta}} - \overline{I^l}_{\mathbf{p}_l})^2 + \sum_{\boldsymbol{\delta}\in\Omega}(I^r_{\mathbf{p}_r+\boldsymbol{\delta}} - \overline{I^r}_{\mathbf{p}_r})^2} \quad (4)$$

where Ω is an $n \times m$ generic template patch centred at the origin of the coordinate system; $\mathbf{p}_l$ and $\mathbf{p}_r$ are the pixels positions (2-vectors) in the left and right images, respectively; and $\boldsymbol{\delta}$ is a variable 2D displacement vector. The bar indicates the mean operator.

The mean subtraction and rescaling operations in (13) help deal with changes in the photometric settings of the two input cameras and possibly with limited non-Lambertian effects. Our experiments showed that for horizontally rectified images taller neighborhood windows (*e.g.* 7×3) help incorporate inter-scanline information better than square windows of similar area, with obvious advantages in terms of speed. Furthermore, the costs $M(l,r)$ can be computed efficiently using moving average techniques [SS02]. The normalization property of (13) ($0 \leq M(l,r) \leq 1 \ \forall l,r$) will turn out to be extremely convenient when setting the costs of the graph edges defined in the next sections.

In our experiments we have compared the Normalized SSD cost with the Normalized Cross-Correlation (NCC) matching cost defined as:

$$M_{ncc}(l,r) = \frac{1}{2} \times \left(1 - \frac{\sum_{\boldsymbol{\delta}\in\Omega}(I^l_{\mathbf{p}_l+\boldsymbol{\delta}} - \overline{I^l}_{\mathbf{p}_l})(I^r_{\mathbf{p}_r+\boldsymbol{\delta}} - \overline{I^r}_{\mathbf{p}_r})}{\sqrt{\sum_{\boldsymbol{\delta}\in\Omega}(I^l_{\mathbf{p}_l+\boldsymbol{\delta}} - \overline{I^l}_{\mathbf{p}_l})^2 \sum_{\boldsymbol{\delta}\in\Omega}(I^r_{\mathbf{p}_r+\boldsymbol{\delta}} - \overline{I^r}_{\mathbf{p}_r})^2}}\right) \quad (5)$$

We have found little difference, in terms of results, between the two implementations but NSSD is considerably faster than NCC (despite our efforts to improve the efficiency of the NCC code[5]). Interesting linearity and consistency properties of the NSSD cost function are discussed in [KCB+05b].

One of the biggest problems of dynamic programming, dense stereo algorithms is that scanlines are treated independently. This induces visible "streaky" artefacts in the output disparity maps and related synthesized images. This issue is addressed here by filtering the matching cost matrix across scanlines, over a three-dimensional cost space.

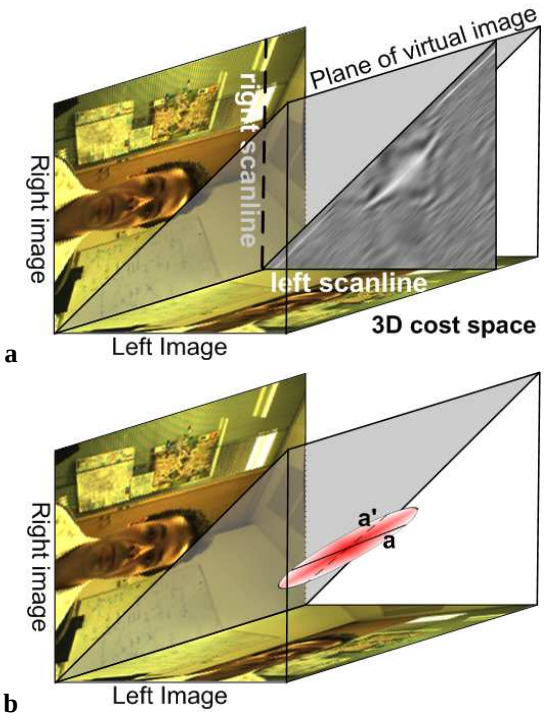


Fig. 14. **The 3D cost space for a pair of stereo images.** (a) Match cost space, as fig. 5a, now shown for full 3D volume. (b) In order to propagate cost information across scanlines a 2D Gaussian filter (represented by the red ellipse) parallel to the virtual image plane is applied to the 3D cost space.

5.1. Inter-scanline consistency and cost aggregation

A solution to the issue of inter-scanline consistency [OK85] is to propagate information across scanlines by detecting and matching vertical edges. This has two drawbacks however: (i) the robust matching of edges is an open issue, especially for occluding contours (precisely where we need most accuracy); (ii) edge detection and matching algorithms are slow. Our solution to the problem of encouraging the propagation of information across scanlines efficiently is to use small window neighborhoods in the cost computation step followed by a separate cost aggregation step.

The algorithm proceeds as follows: Firstly the cost matrices $M(l,r)$, associated with each pair of scanlines, are built and stacked to form a three-dimensional cost space as in fig. 14a. Secondly, a 2D Gaussian filter is applied with principal axes $\mathbf{a}$ parallel to the virtual image plane (fig. 14b). The axis $\mathbf{a}$ of the Gaussian kernel orthogonal to the left and right scanline axes is responsible for enforcing inter-scanline con-

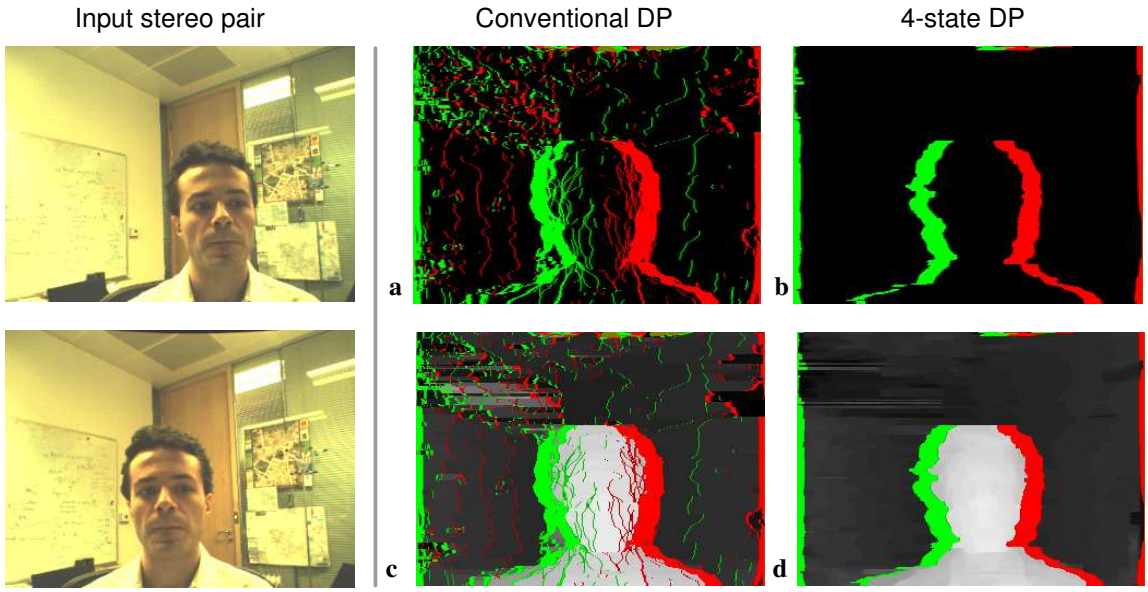


Fig. 15. **Comparing the different DP models.** (a,b) Occlusion maps obtained from the conventional DP and the 4-state DP algorithms, respectively. Red indicates left occlusions and right occlusions are green. (a) In conventional DP numerous matched pixels are incorrectly classified as occluded. True occlusions around the head show as broken up maps of ambiguous labels, and would cause "reset" artefacts in the synthesized cyclopean view. (b) In 4-state DP true occlusions around the head are correctly detected as unfragmented regions. Throughout, 7×3 window patches were used in the computation of matching costs; a value of $\sigma_a = 2.0$ has been used for the cost filtering step. (c,d) Disparities corresponding to (a,b), respectively. Removal of spurious occlusion labels helps also to clean up the disparity map. The disparity values have been scaled up 1.5 times for ease of visualization.

sistency of the costs; the other axis $\mathbf{a}'$ produces additional smoothing of sharp corners in the occlusion map by encouraging fronto-parallel surfaces [SS02]. Typical values for Gaussian smoothing parameters are: $\sigma_a = 3$ pixels along the $\mathbf{a}$ axis and $\sigma_{a'} = 2$ pixels along the $\mathbf{a}'$ axis.

Cost-filtering acts directly on the matching cost function rather than on the final matching path or the disparity map. In fact, cost aggregation precedes the optimal path finding step. The result is effective information propagation across scanlines with improved occlusion positioning without necessarily smoothing disparities (fig. 16). Furthermore, the cost filtering step, being a separable 2D convolution can be implemented efficiently by using two 1D convolutions.

Note that if we had used standard, un-normalized SSD in the cost computation step, then the use of large window neighborhoods (with Gaussian weighting) would have been equivalent to the cost aggregation performed in this section. Furthermore, we have found that normalized SSD costs on small windows works considerably better than standard SSD. Further research is necessary here to assess an optimal matching cost function.

The output of the cost aggregation process is the new set of $M(l, r)$ costs used, as input to the 4-state DP algorithm already described in section 4.

6. Evaluating the estimated disparity and occlusion maps

The goal of this section is two-fold: i) demonstrating the advantages of our new DP algorithm with respect to other techniques and, ii) defining measures of accuracy of occlusion detection for comparison with state of the art Graph-Cut techniques.

6.1. Advantages of four-state model for DP

Four-state vs conventional DP. Figure 15 demonstrates the effect of moving from the conventional DP algorithm to the four-state DP one. Comparing fig. 15a and fig. 15b one can see that the four-state model removes most of the incorrect occlusion events which occur in the background of fig. 15a (isolated

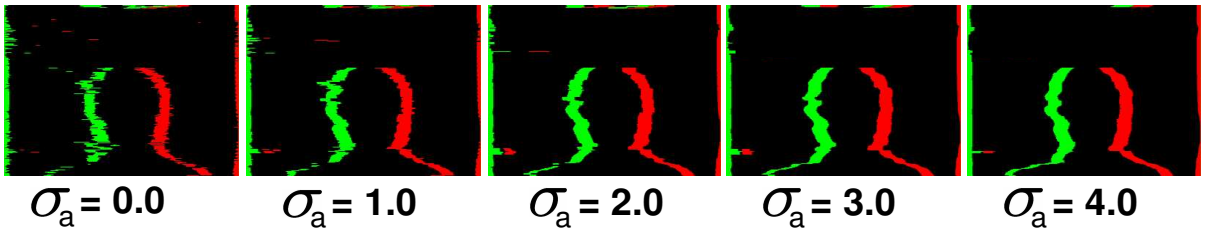


Fig. 16. **Inter-scanline consistency by cost filtering.** Occlusion maps obtained by 4-state DP for different values of σ_a (section 5.1). In this experiment the value of $\sigma_{a'}$ has been kept fixed at $\sigma_{a'} = 2\,\text{pixel}$. Cost filtering helps achieve more "compact" occlusion regions.

red and green points). The black pixels in fig. 15a,b correspond to matched moves. The four-state model correctly classifies small jumps in disparity levels as matched moves discretizing slanted surfaces (*e.g.* the face or the slanted walls in the background). Furthermore, this new graph structure is used to favour long runs of occlusions. In fig. 15b the occlusions are: (i) correctly located along the boundary of the foreground object, and (ii) detected as compact, solid regions. This, in turn, leads to better occlusion maps and more convincing synthesis, as will be demonstrated in section 9.

The effect of cost-space smoothing on inter-scanline consistency. Figure 16 shows the effect of varying the σ_a parameter for cost-space smoothing. As the value of the standard deviation σ_a of the Gaussian kernel increases the runs of occlusions become correctly aligned with the outline of the foreground head. Importantly, above a certain value of σ_a the results become quite stable.

6.2. Evaluating accuracy of disparities

Here we compare the computed disparities with those in the standard Middlebury database. Fig. 17 shows a snapshot of the stereo algorithm evaluation table in `http://www.middlebury.edu/stereo/`. The Middlebury error metrics show our algorithm amongst the most accurate of the efficient techniques (*e.g.* `Tree DP`, `Comp. win`, `Realtime`). Proper handling of occlusions vs slanted surfaces in large-disparity images is an advantage of 4-state DP. In this evaluation identical parameters have been used for all four tests. Due to its thin structures the "tsukuba" image pair presents most difficulties. This problem is typical of scanline dynamic programming techniques which impose the ordering constraint. However, note that thin structures do not arise often in video-conferencing kind of images.

While we cannot expect 4-state DP to beat the fully two-dimensional MRF techniques, its near real-time performance and accurate occlusion modelling present considerable advantages, especially for live view synthesis applications. Since realistic new view synthesis is the main objective of this paper the next sections will focus on evaluating the accuracy of the recovered occlusion maps.

6.3. Assessing the quality of occlusions.

Figure 18a,$\cdots$d illustrates the results of applying 4-state DP to four of the Middlebury test image pairs. Occlusions are recovered correctly only where they occur, while slanted surfaces are modelled by matched moves only. In all the above experiments the parameters were kept fixed for all image pairs and identical to those used in sec. 6.2. For comparison, ground-truth occlusion maps for the Middlebury datasets (fig. 18a',$\cdots$d') were computed by taking the left and the right disparity maps and cross-projecting them. The presence of *both* disparity maps enabled unambiguous detection of true occlusion while correctly modeling slanted surfaces.

The results of estimating (left) occlusion and disparity maps are quite convincing, even for non videoconferencing-like images. However, our algorithm has been designed to cope with situations involving much greater disparity ranges than the ones shown in [SS02]. For the next experiment we have created our own test stereo pair, characterized by a much larger disparity and occlusion range.

Algorithm	Tsukuba			Sawtooth			Venus			Map	
	all	untex.	disc.	all	untex.	disc.	all	untex.	disc.	all	disc.
Sym.BP+occl.	0.97 1	0.28 2	5.45 1	0.19 1	0.00 1	2.09 1	0.16 3	0.02 3	2.77 5	0.16 1	2.20 1
...	...	...	...	...	...	...	...	...	...	...	...
Multiw. cut	8.0836	6.5332	25.3337	0.6110	0.4625	4.6011	0.53 5	0.31 5	8.0617	0.26 5	3.27 5
Graph cuts	1.8618	1.0015	9.3515	0.42 8	0.1415	3.76 9	1.6923	2.3022	5.40 9	2.3934	9.3521
Tree DP **	1.7715	0.38 5	9.4816	1.4424	0.8428	6.8719	1.2114	1.4116	5.04 8	1.4526	13.0029
OUR METHOD **	4.7030	3.6827	21.0533	1.4323	0.1716	13.9330	1.1812	0.59 6	17.9828	0.30 7	4.2310
Comp. win. *	3.3625	3.5425	12.9125	1.6127	0.4524	7.8723	1.6722	2.1819	13.2422	0.3311	3.94 9
Realtime **	4.2529	4.4730	15.0529	1.3222	0.3522	9.2124	1.5319	1.8017	12.3320	0.8120	11.3526
Cooperative	3.4926	3.6526	14.7727	2.0328	2.2932	13.4129	2.5729	3.5229	26.3836	0.22 3	2.37 2
...	...	...	...	...	...	...	...	...	...	...	...

Fig. 17. **Evaluating 4-state DP with respect to the disparity error metrics in [SS02].** A snapshot of the evaluation table in `http://www.middlebury.edu/stereo/`. The red stars indicate different levels of reported algorithmic efficiency: double star for high efficiency ($>$ 1fps), single star for medium ($<$ 1fps) and no star for either low ($<<$ 1fps) or un-reported frame-rate. Four-state DP ranks amongst the best of the fast techniques. Additionally, proper handling of occlusions vs slanted surfaces (missing for example in `Tree DP`) is provided. However, being at the top of this table is not the main objective of this paper; while accurate new-view synthesis is.

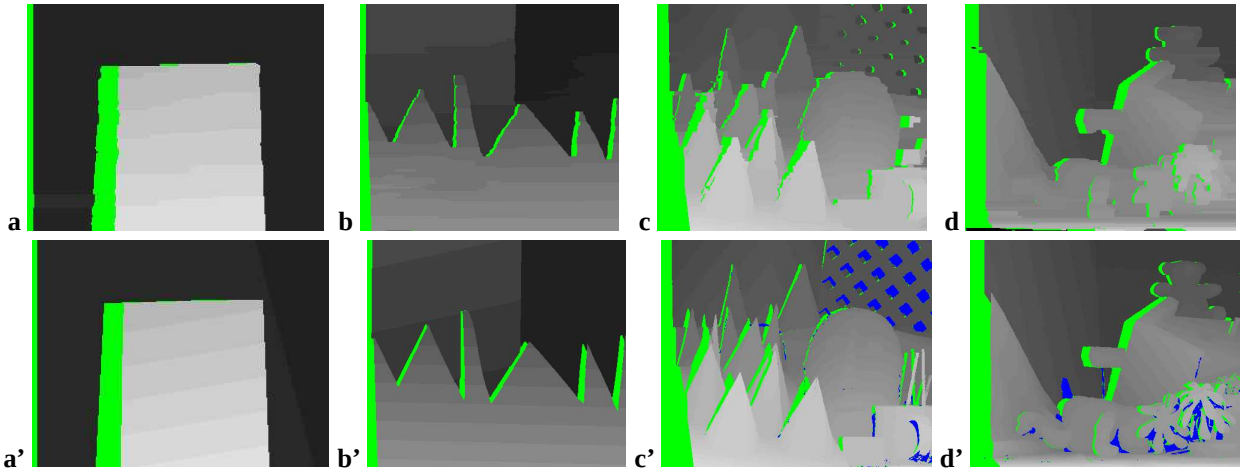


Fig. 18. **Accuracy of occlusion estimation on the Middlebury stereo pairs.** (a,$\cdots$,d) Estimated occlusions and disparities for the left view of `Map`, `Sawtooth`, `Cones` and `Teddy` stereo pairs, respectively. (a',$\cdots$,d') Corresponding ground-truth occlusions and disparities (blue pixels denote unlabelled pixels). Good correspondence between the ground-truth occlusions and those estimated by 4-state DP is evident. The maximum occlusion gap of 55 pixels (12% of image width) occurs at the edge of the `Cones` image. The maximum occlusion for `Map` is about 9% of the image width.

6.4. Quantitative assessment of occlusion errors.

Figure 19 compares our results with the ones obtained by three well known graph-cut techniques which estimate occlusions [BGCM02, HD98, KZ01] and the conventional 3-move DP [CHRM96]. Excepting the graph-cut method in [KZ01][6], the other algorithms are based on our own implementations.

Figure 19a,b are the two input images used in this experiment. The photographed scene is made of two background slanted planes and one foreground fronto-

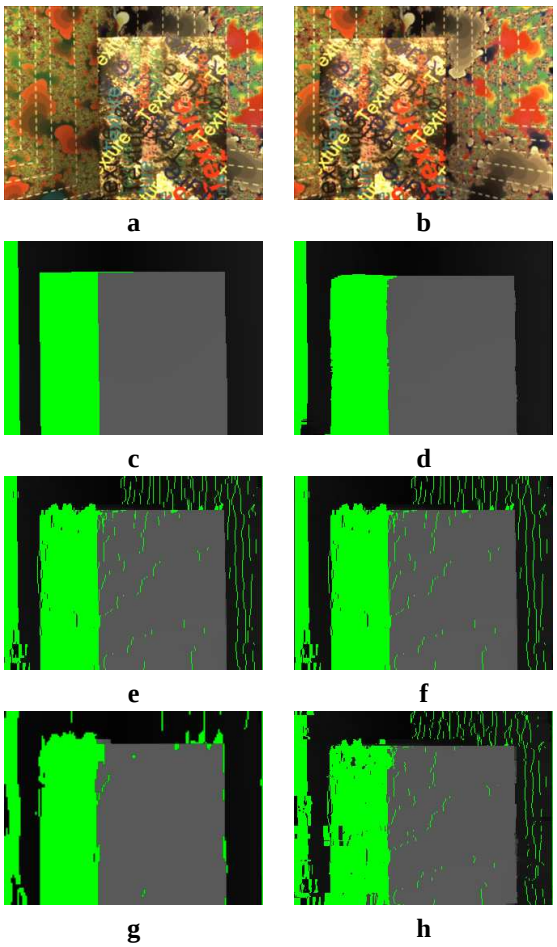


Fig. 19. **Comparing the occlusion maps returned by different algorithms**. (a,b) Input left and right images, respectively. (c) Ground-truth occlusion (and disparity) map with reference to the left camera. As usual, green indicates right-occlusion. (d-h) Left-referenced occlusion maps recovered by using: (d) 4-state DP; (e) Buehler *et al.* graph-cut algorithm [BGCM02]; (f) Kolmogorov *et al.* graph-cut algorithm [KZ01]; (g) Ishikawa *et al.* graph-cut algorithm [HD98]; (h) Cox *et al.* dynamic-programming algorithm [CHRM96].

parallel plane. The stereo pair is characterized by a maximum disparity range of 90 pixels and a maximum occlusion gap of 72 pixels which corresponds to 22.5% of the image width (image dimensions are 320×240); more than twice the occlusion gap of the `Map` stereo pair in fig. 18. The ground truth disparity and occlusion map (fig. 19c) was obtained by least-square fitting of the two planes in the background and the planar surface of the foreground object. The fitting process was initialised by dense matches produced by

Algorithm	*Misclass. rate*	*Runtime*
4-state DP	**2.61%**	**1.57s**
Buehler *et al.* [BGCM02]	6.45%	468 s
Kolmogorov *et al.* [KZ01]	6.57%	65 s
Ishikawa *et al.* [HD98]	6.61%	912 s
Cox *et al.* [CHRM96]	8.17%	**0.31 s**

Fig. 20. **Accuracy of occlusion detection**. Comparing accuracy and performance of different state-of-the-art dense stereo algorithm in estimating occlusion maps.

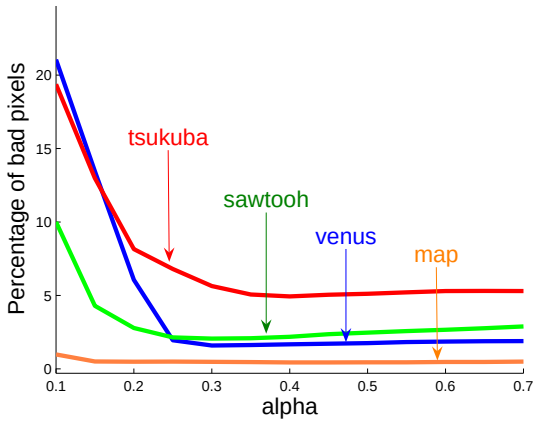
DP. The segmentation of the foreground object was perfomed manually and the correctness of the resulting ground truth was verified by manual inspection. Some graph-cut algorithms such as [KZ01] produce left and right occlusion maps and not the cyclopean map. Therefore, in order to reduce the possibility of error we have decided to compare the performance of the selected algorithms always with reference to the left camera. Fig. 19d-h show the results of computing the left-referenced occlusion and disparity maps via different algorithms.

In order to quantify the occlusion accuracy we define a new error measure, the *Misclassification rate*.

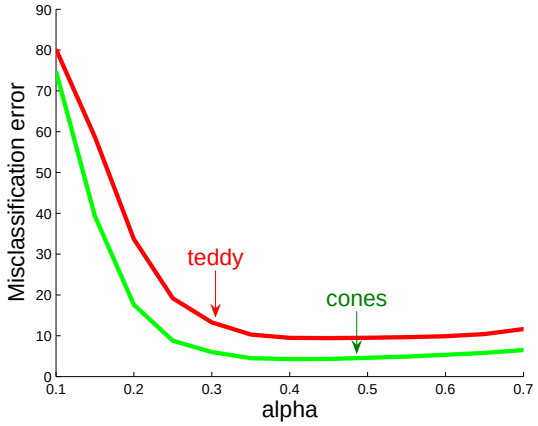
Misclassification rate is estimated by comparing the occlusion maps recovered by each algorithm with ground truth (fig. 19c) and counting the number N_m of misclassified pixels (both false positives and false negatives).

Comparative results. The misclassification rate has been measured for all the occlusion maps in fig. 19d-h and the results collected in the table in fig. 20.

The three graph-cut algorithms [BGCM02, HD98, KZ01] perform comparably and considerably better than conventional DP. The reduced misclassification error obtained by 4-state DP is due to the extended four-label pixel classification and the enforcement of occlusion-run constraints. While the GC framework in [KZ01] supports runs of occlusions, these are not correctly modeled in the sense that occluded moves are used to approximate slanted surfaces. Furthermore, right and left occlusions are not differentiated. An alternative GC algorithm [HD98] that does use explicit labels for occlusion produces poor results for lack of constraint enforcement. The effect of approximating slanted surfaces with occluded moves can be observed in figs. 19e,f,h; where the background is constellated by a large number of vertically aligned occluded pixels



a



b

Fig. 21. **Sensitivity to parameters**. (a) Sensitivity of the "percentage of bad pixels" error metric with respect to α. Note the flat behaviour for $\alpha > 0.35$. (b) Sensitivity of occlusion misclassification rate with respect to α. In all cases errors are measured on standard Middlebury stereo pairs. Occlusion errors are measured on the two stereo pairs with the largest occlusions. 4-state DP is not particularly sensitive to values of α varying within a reasonable range.

(marked in green). These results show that the combination of both an extended occlusion model for correct pixel classification and the enforcement of constraints on occluded areas achieves the best results. Figure 20 also shows our algorithm being the second fastest, immediately after the very efficient (but relatively poor quality) Cox DP.

Further notes on our experimental procedure. The different energy minimization algorithms analysed in this section have been applied to *exactly* the same cost space, which was computed only once[7]. This was done to eliminate variability due to different matching cost functions or cost smoothing parameters. Furthermore, for each algorithm we have selected the combination of parameters which has lead to the best results for that specific algorithm. In the case of the algorithm in [KZ01] the parameters were automatically selected by the original implementation. Finally, all algorithms were run on the same machine, a 3GHz, 1Gb RAM Pentium IV desktop computer.

Our results place 4-state DP amongst the most accurate efficient algorithms for shape *and occlusion* recovery from large-disparity image stereo pairs. Furthermore, we propose a novel error metric (for occlusions) which should be added to the set of metrics defined in [SS02].

6.5. Parameter sensitivity.

In order to assess the sensitivity of our algorithm with respect to its parameters we have measured the "percentage of bad pixels" [SS02] and misclassification rate for different values of α, β and γ. Fig. 21 shows some exemplary error plots illustrating sensitivity of α. Both disparity-based (fig. 21a) and occlusion-based (fig. 21b) error metrics show a flat behaviour in the range $\alpha \in [0.35, 0.65]$. Sensitivity with respect to the β and γ parameters has been found to be about ten times lower. Generally, good stability of the output errors has been found for relatively large ranges of parameter values and for both disparity-based and occlusion-based error metrics.

The previous sections have: i) illustrated 4-state DP for the reliable estimation of occlusion and disparity maps and ii) evaluated our algorithm against state of the art techniques. The next sections focus on the new-view rendering problem, how to best make use of the extracted geometric information for the purposes of efficient virtual-image generation.

7. Rendering occlusions

High-quality virtual image generation requires effective synthesis over occluded regions. We have investigated two strategies for occlusion filling: *static* filling, which applies to single pairs of stereo images and *temporal* filling which, instead, models what lies behind the occlusions from long sequences of stereo images.

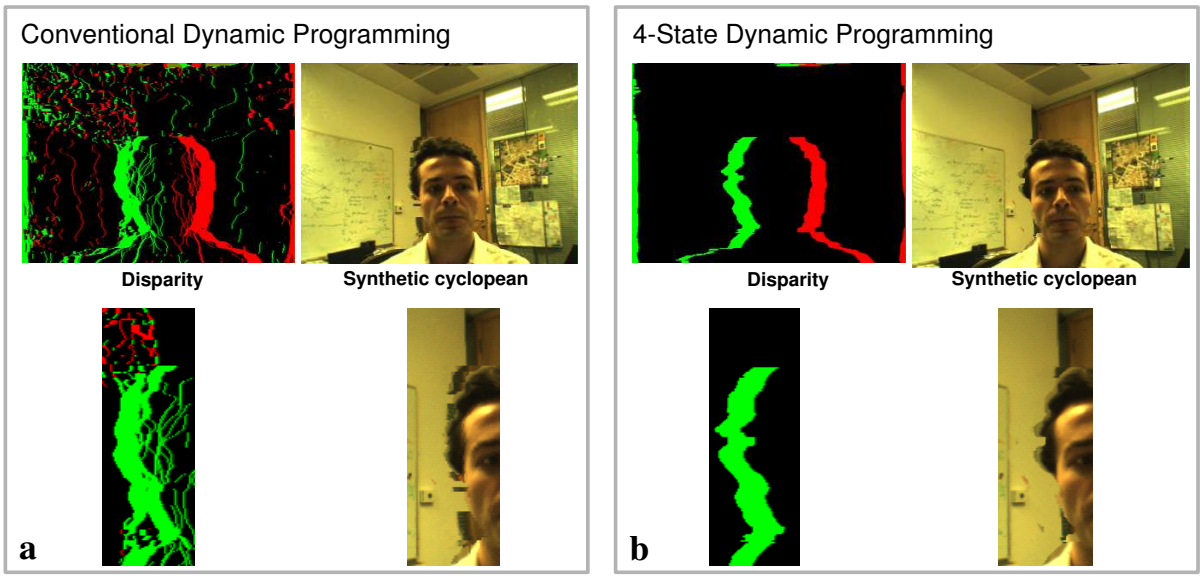


Fig. 22. **Eliminating the reset artefact.** (a) Occlusion map and reconstructed cyclopean view for conventional DP. Small islands of spurious matched pixels inside occlusion regions cause the reconstruction of the occluded areas to fail. Note also that many pixels on slanted surfaces have been incorrectly classified as occluded. (b) Occlusion map and reconstructed cyclopean view for the proposed four-state DP. The compactness of the recovered occlusion regions produces a more accurate cyclopean reconstruction: the background door frame is now straight and almost completely artefact-free.

Static occlusion filling. Given an input stereo pair of images 'fronto-parallel' synthesis of the occluded regions is done via the algorithm illustrated in fig. 7c. As discussed in section 2.2 effective filling of the occlusions is disrupted by inaccurate labelling. Figure 22b shows an example of realistic occlusion synthesis achieved by 4-state DP. Note that fig. 22b is free from any "halo" or "reset" artefacts.

Temporal occlusion filling. When the static filling algorithm is applied to long image sequences, temporal artefacts become visible in occluded regions. Moreover, because of the lack of pixel correspondence, stability of synthesis is a particular issue in the occluded areas. One solution to this problem is the construction and dynamic update of a model of the background used to fill in the regions of missing information. The algorithm is in two steps: the first step segments the foreground from the background at each time instance; the second step uses the newly uncovered (disoccluded) pixels of the background to improve the background model.

The segmentation step, performed at each time instance, proceeds as follows:

Given the estimated min-cost surface $\mathcal{S}$:

1. Along each scanline in the min-cost surface, for each run of occlusions, the disparity at the highest disparity end of the run is histogrammed (fig. 24).
2. The valley in the resulting bi-modal histogram determines the adaptive threshold disparity value $\hat{d}$ that is used for the background/foreground segmentation.

Figure 24b shows a typical histogram. The peak near the origin is due to the long and thin occlusion bands at the edges of the image, while the peak at higher disparity values is due to the foreground object and is the one we are mostly interested in. This kind of bi-modal histogram turns out to be characteristic of sequences of talking heads. This approach for automatic threshold detection works better than histogramming the whole set of estimated disparities. This is because the selected pixels (marked in white in fig. 24a) are more representative of the foreground object. The technique has been proven to work also in situations where part of the background are very close (in depth) to the talking head (*e.g.* a receding wall).

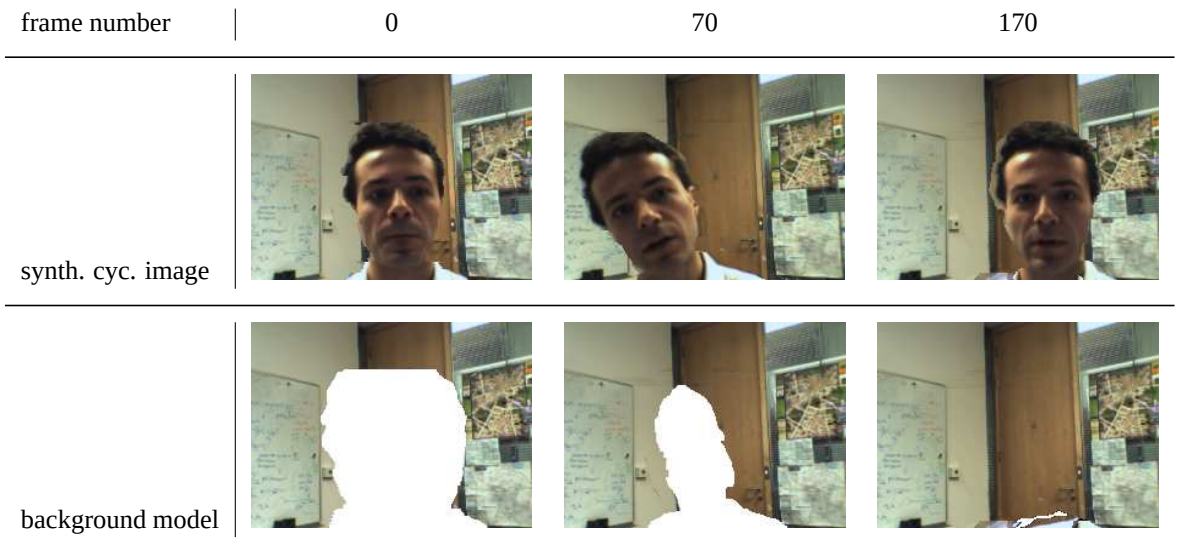


Fig. 23. **Temporal background generation.** (top row) Synthesized cyclopean views for different frames. More examples of synthesised cyclopean views are provided in the results section. (bottom row) Corresponding background models. As new regions of the background are discovered the background model is updated and the blank region (occlusion) progressively filled.

In the second step of the algorithm a background model is constructed and updated at each time instance. The background model is made of three elements: its disparity map D_B in cyclopean coordinates, and the corresponding left and right images I_B^l and I_B^r, respectively. At each time instance t the background model is updated by the following rule:

$$\begin{aligned} D_B^t(\mathbf{p}) &= \phi D_B^{t-1}(\mathbf{p}) + (1-\phi) D^t(\mathbf{p}) \\ I_B^{l^t}(\mathbf{p}_l) &= \phi I_B^{l^{t-1}}(\mathbf{p}_l) + (1-\phi) I^{l^t}(\mathbf{p}_l) \\ I_B^{r^t}(\mathbf{p}_r) &= \phi I_B^{r^{t-1}}(\mathbf{p}_r) + (1-\phi) I^{r^t}(\mathbf{p}_r) \end{aligned} \quad (6)$$

where $\mathbf{p}$ is a pixel whose disparity $D(\mathbf{p})$ falls below the automatically computed foreground threshold $\hat{d}$ (and thus belongs to the background). The points $\mathbf{p}_l$ and $\mathbf{p}_r$ are the corresponding positions on left and right input images, respectively. $D_B^t(\mathbf{p})$ is the disparity of the pixel $\mathbf{p}$ in the current background model at time t. The scalar factor ϕ represents a decay constant ($0 \leq \phi \leq 1$). The update rule in (6) applies to all the pixels which belong to the background and are visible and does not apply to occluded pixels. The use of the exponential memory parameter ϕ allows for a relaxation of the static background assumption. In our experiments $\phi = 0.9$ achieves a good balance between keeping the previous values of the background pixels and updating them in the case of dynamic events on the background.

Figure 23 illustrates the results of the temporal background filling algorithm. During the video-communication session the head moves and disoccludes portions of the background. The background model is updated and, after a few frames, if the head moves substantially, the background is completely reconstructed.

Advantages and disadvantages of static and temporal filling strategies. The static occlusion filling strategy is based on the assumption of a fronto-parallel background, which, although most of the time produces good results, may not make sense for scenes with very slanted surfaces. Furthermore, the static occlusion filling requires solid and accurate occlusion areas which are achieved by four-state DP but are not in conventional DP or graph-cut techniques. On the other hand, in temporal occlusion filling, the background model is learnt from the disocclusions of previous frames. This introduces the need for background/foreground segmentation and the assumption of a quasi-static background.

Temporal occlusion filling and background modeling is especially useful in the next section which introduces the three-dimensional motion of the virtual camera. In fact, as the virtual camera centre moves away from the baseline of the two input cameras less information is available from the current pair of stereo frames about the occluded regions, and temporally acquired background information becomes extremely

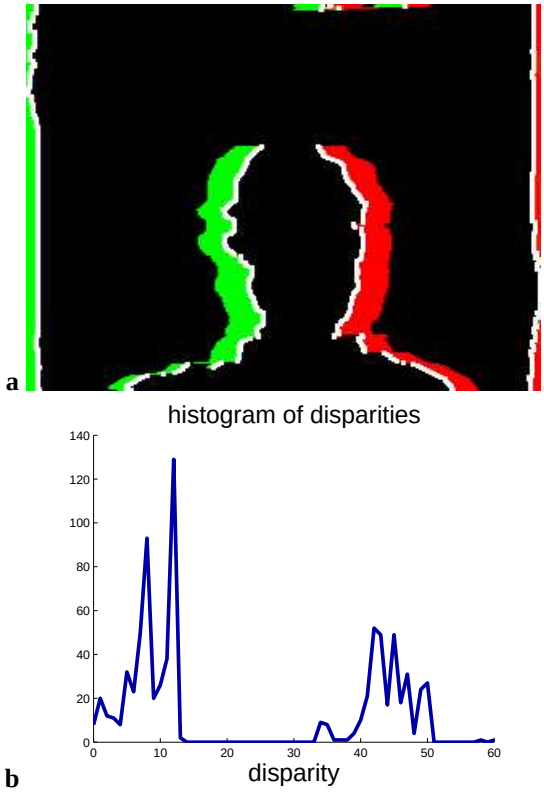


Fig. 24. **Foreground/background segmentation.** (a) The pixels corresponding to the higher-disparity end in each horizontal run of occlusion is marked in white. (b) The histogram corresponding to the disparities extracted in (a).

useful for reconstructing unseen regions. Overall, we have found that a combination of the two techniques works best: we use static filling in the half-occluded areas which have not yet been observed, and temporal filling in those occluded regions which have been disoccluded in previous frames.

8. Rendering from variable viewpoint

The ability to create virtual images from generic viewpoints is of considerable interest both for interactive video and teleconferencing applications. Conventionally, one way of generating novel views from virtual camera locations is by: (i) transforming the computed disparity map into a 3D surface (*e.g.* by means of a triangle mesh), (ii) texture mapping it with one of the two images, (iii) projecting the texture-mapped surface into the plane of the virtual camera. This section describes a novel, compact technique for rendering virtual views directly from the estimated disparity surface, thus overriding the need to construct an explicit 3D model of the scene.

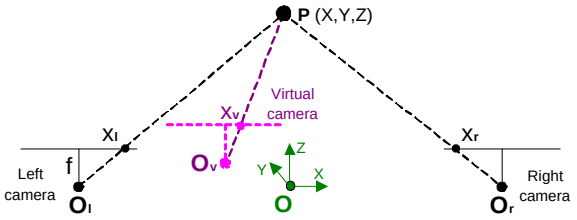


Fig. 25. **Notation for virtual image generation.** $\mathbf{O}_l$, $\mathbf{O}_r$ and $\mathbf{O}_v$ are the optical centres of left, right and virtual cameras respectively. The optical centre of the virtual camera can be placed anywhere in space and the corresponding virtual image is synthesized by our algorithm.

The geometry of the virtual camera. Figure 25 shows a plan view of the system with the optical centre of the virtual camera being placed in the generic location denoted $\mathbf{O}_v$. A 3D scene point $\mathbf{P}$ is projected on the left and right images into the points $\mathbf{p}_l = (x_l, y_l)^\top$ and $\mathbf{p}_r = (x_r, y_r)^\top$, respectively. Also, $\mathbf{P}$ is projected on the cyclopean camera (with optical centre in $\mathbf{O}_c = \mathbf{O}$) in the point $\mathbf{p}_c = (x_c, y_c)^\top$ (not shown in the figure) and on the virtual camera in the point $\mathbf{p}_v = (x_v, y_v)^\top$. The disparity between the corresponding left and right image points is easily computed as

$$d = x_l - x_r = f\frac{B}{Z} \quad . \tag{7}$$

In the cyclopean camera, by triangle similarity we can compute

$$x_c = f\frac{X}{Z} \quad . \tag{8}$$

For a virtual camera with optical center in $\mathbf{O}_v = (T_x, T_y, T_z)^\top$ we can write: $(X - T_x) : x_v = (Z - T_z) : f$, from which

$$x_v = f\frac{X - T_x}{Z - T_z} \quad . \tag{9}$$

By substituting (7) and (8) into (9) we obtain: $x_v = \frac{x_c - dT_x/B}{1 - dT_z/(fB)}$ which, together with the analogous equation for the y_v coordinate, can be rewritten in homogeneous coordinates as:

$$\begin{pmatrix} x_v \\ y_v \\ w \end{pmatrix} = \begin{bmatrix} 1 & 0 & -T_x/B & 0 \\ 0 & 1 & -T_y/B & 0 \\ 0 & 0 & -T_z/(fB) & 1 \end{bmatrix} \begin{pmatrix} x_c \\ y_c \\ d \\ 1 \end{pmatrix} \tag{10}$$

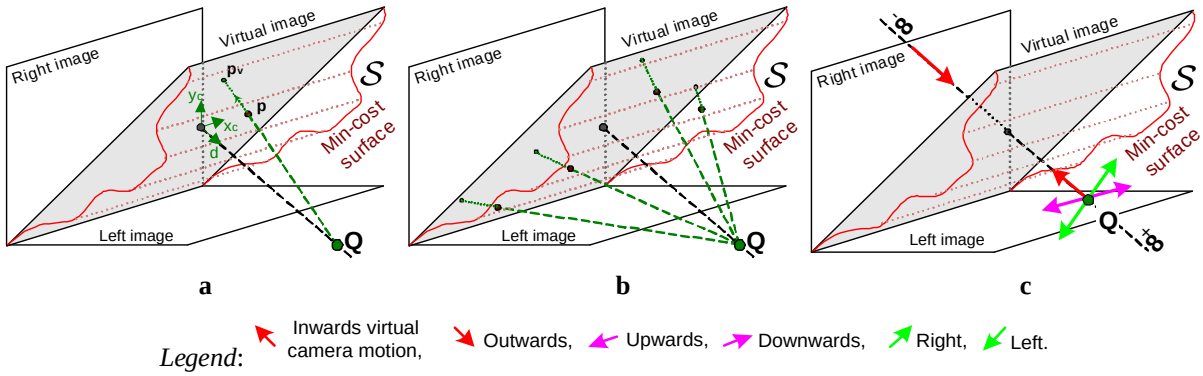


Fig. 26. **Virtual camera motion.** (a) The 3D motion of the virtual camera is achieved by direct projections of points on the minimum cost surface onto the virtual image plane. The reference coordinate system (x_c, y_c, d) has origin in the centre of the virtual image plane. (b) The virtual image is generated directly by projecting points from the minimum-cost surface $\mathcal{S}$ into the virtual image plane. (c) Moving the centre of projection $\mathbf{Q}$ corresponds to translating the virtual camera. The coloured arrows indicate the mapping between moving the centre of projection $\mathbf{Q}$ in our diagram and the corresponding translations of the virtual camera in the scene.

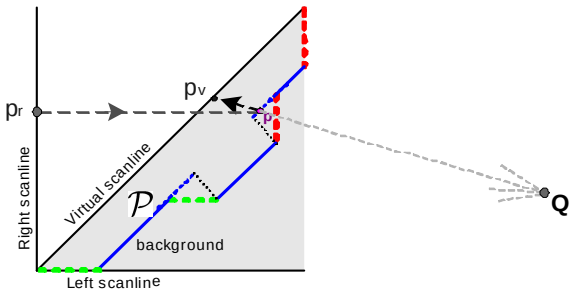


Fig. 27. **Occlusion filling for generic virtual camera placement**. The only difference with respect to the cyclopean occlusion filling illustrated in fig. 7c is that now the generic direction of projection is dictated by the position of the centre $\mathbf{Q}$.

Equation (10) represents a projection of 3D points into a plane [HZ00]. It can be proven that (10) corresponds to projecting points of the min-cost surface into the corresponding points on the plane of the virtual image (up to a scale, diagonal matrix) as illustrated in fig. 26a.

From (10) the centre of projection $\mathbf{Q}$ is readily computed as the null vector of the projection matrix, thus yielding: $\mathbf{Q} = \left(\frac{T_x}{B} \;\; \frac{T_y}{B} \;\; 1 \;\; \frac{T_z}{fB} \right)^\top$. Note that for $T_z = 0$ the transformation (10) is a *parallel* projection ($\mathbf{Q}$ is at infinity). This, in turn means that sidewise motion (in the X direction) and up/down motion (in the Y direction) of the virtual camera can be easily simulated by projecting points of the disparity surface $\mathcal{S}$ onto the virtual image plane via parallel rays. On the contrary, the inwards/outwards translation of the virtual camera ($T_z \neq 0$) is achieved by means of a *central* projection with *finite* centre of projection $\mathbf{Q}$. The simple mapping between the motion of the centre of projection $\mathbf{Q}$ and the corresponding translation of the virtual camera is illustrated in 26c,d. For instance, inwards camera translation (not zoom) is achieved by moving the centre $\mathbf{Q}$ from $+\infty$ towards the plane of the virtual image.

Note that for $\mathbf{Q} = (-1/2, 0, 1, 0)^\top$ (*i.e.* $\mathbf{O}_v = (-B/2, 0, 0)^\top$) the virtual image corresponds to the input left image, for $\mathbf{Q} = (1/2, 0, 1, 0)^\top$ (*i.e.* $\mathbf{O}_v = (B/2, 0, 0)^\top$) the virtual image corresponds to the input right image, and for $\mathbf{Q} = (0, 0, 1, 0)^\top$ (*i.e.* $\mathbf{O}_v = (0, 0, 0)^\top$) the virtual image corresponds to the cyclopean image.

Synthesizing virtual images from generic viewpoints. Given a point $\mathbf{p}$ on the minimum-cost surface and its corresponding virtual position $\mathbf{p}_v$ (fig. 26a and *cf.* fig. 7a), the corresponding pixel value (intensity or colour) is given by a combination of the pixel values of the corresponding pixels $\mathbf{p}_l$ and $\mathbf{p}_r$ in the input images according to the following equation[8]:

$$I^v(\mathbf{p}_v) = (1-\mu) I^l(\mathbf{p}_l) + \mu I^r(\mathbf{p}_r) \qquad (11)$$

with $\mu = \frac{|\mathbf{O}_x^v - \mathbf{O}_x^l|}{B}$; where the subscript indicates the x component of optical centres of the two input cameras.

Occlusion filling and rendering. The filling of occlusions for generic virtual view placement is very similar to the cyclopean case illustrated in fig. 7c. As

a **b**

Fig. 28. **Example of gaze correction.** (a) Cyclopean image synthesized via the algorithm in [COL93]. This image is identical to that in fig. 3b and is repeated here for clarity. The input left and right images are shown in fig. 3a,c, respectively. (b) The cyclopean, gaze-corrected view generated by our algorithm. The gaze has been corrected while eliminating the artefacts of (a).

a **b** **c**

Fig. 29. Another example of **gaze correction**. The central image, (b) has been generated from the two input views (a,c) and shows correct gaze (the person is looking at us). There are no significant "halo" effects or streaky artefacts.

shown in fig. 27 now the direction of projection is dictated by the position of the centre $\mathbf{Q}$, the cyclopean case being a special case of this general projection.

The rendering algorithm described here is an extension of the cyclopean rendering presented in section 2.2. By inspection of (11) one can see that in the cases where $\mathbf{O}_v = \mathbf{O}_l$ or $\mathbf{O}_v = \mathbf{O}_r$, the original left and right views are resynthesised *exactly*, and independently from the recovered disparities, as expected. Further advantages of our rendering technique are: (i) direct view-dependent texture rendering which negates the need for surface triangulation and (ii) effortless occlusion reconstruction by simple projection of the minimum-cost surface. High-quality output images are obtained by standard reverse mapping and bilinear interpolation techniques. Note that rotations of the virtual camera have not been considered here. Rotations may be achieved by homography-based image warping. However, virtual-camera rotation does not seem to be an important requirement in videoconferencing.

9. New view synthesis results

This section presents a number of synthesis results achieved on real input sequences. In particular, we demonstrate: gaze correction, cyclopean view generation, three-dimensional translation of the virtual camera, simple editing such as background substitution.

Gaze correction by cyclopean view synthesis. Figure 28 shows an example where the input left and right images of fig. 3 have been used to generate the cyclopean view via the proposed algorithm. Note that the spatial artefacts (streaks in fig. 3b) have been removed. In the output image (fig. 28) the gaze has been corrected. Another example of gaze correction from stereo images is illustrated in fig. 29.

3D translation of the virtual camera. Figure 30 shows an example of translating the virtual camera towards and away from the visualized scene. Note that this is different from simple zooming or cropping of the output image. Parallax effect may be noticed in the boundary between the head and the background, thus providing the correct three-dimensional feel.

Figure 31 shows an example of in-plane translation (with $\mathbf{O}_v$ on the XY plane) of the virtual camera. Notice the relative displacement of the head with respect to the background.

Cyclopean view generation in long sequences. Figure 32 and 33 demonstrate the effectiveness of the proposed algorithm for reconstructing cyclopean views of extended temporal sequences. It can be observed that most of the spatial artefacts (*e.g.* streaks, halo) and temporal artefacts (*e.g.* flickering) are attenuated. The background close to the foreground/background transitions is correctly synthesized. Exemplary synthesized videos are available at `http://research.microsoft.com/vision/cambridge/i2i/movies/4pdp.zip`

Basic 3D scene editing. The proposed algorithm generates novel, virtual views, but also, as a by-product, a 3D representation of the observed scene. The latter can be advantageous for 3D scene editing. As an example, fig. 34 demonstrates the possibility of replacing the original background with a different one, either taken from real photographs or artificially generated. This is made possible thanks to the foreground/background segmentation step described in section 7. Recent developments of

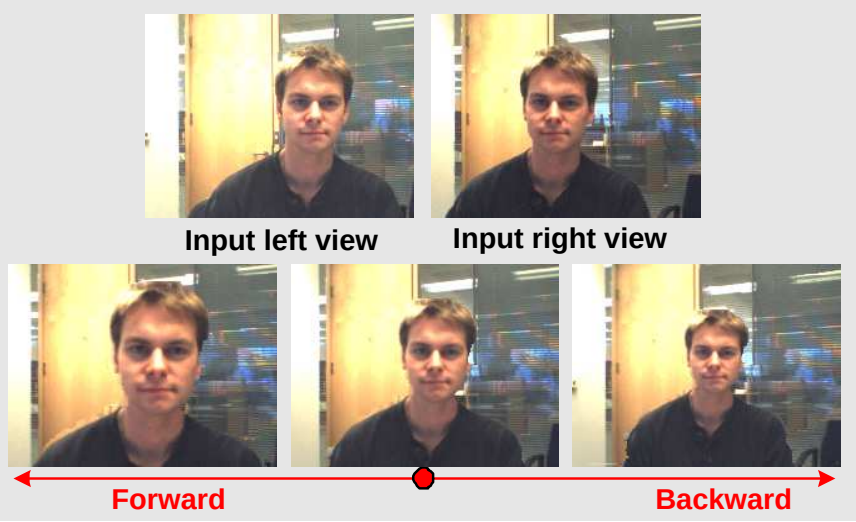


Fig. 30. **Forward/backward translation of virtual camera.** The bottom row shows the synthesized cyclopean views with (left) forward virtual camera translation, (centre) no virtual camera translation, (right) backward virtual camera translation. Notice the *parallax* effect around the head.

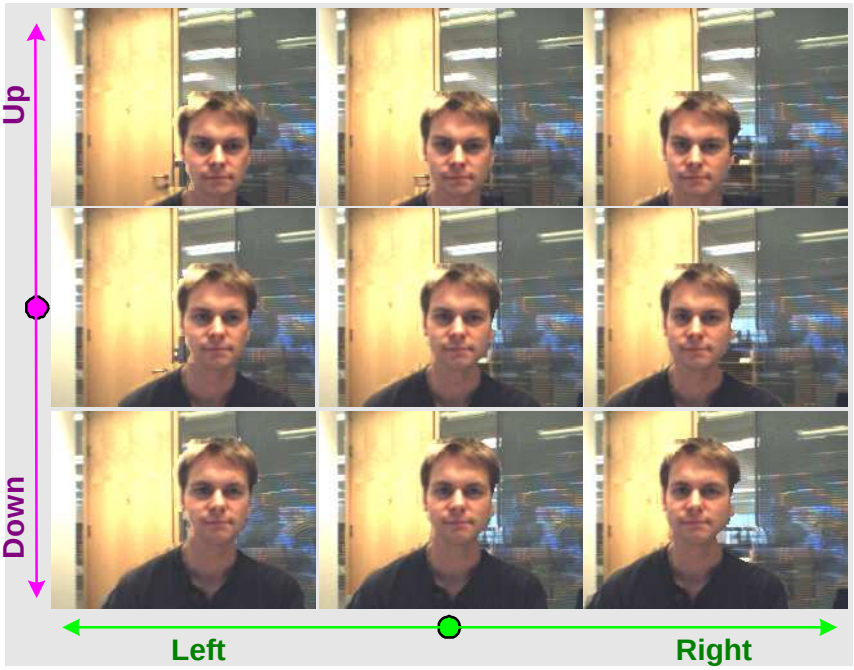


Fig. 31. **In-plane translation of virtual camera.** The left and right input images are the same as in fig. 30. This table shows the synthesized images corresponding to translation of the virtual camera along the x and y axes. Notice the *parallax* effect around the head. Also, the door frame is reconstructed nicely despite it being partially occluded in the right input view.

the background substitution technique may be found in [KCB+05a]. Sophisticated matting techniques for high-quality layer compositing are not the focus of this paper.

10. Conclusions and future work

This paper has described an efficient algorithm for the synthesis and geometric manipulation of high-quality

Fig. 32. **Cyclopean image synthesis for long sequences.** Some frames of a cyclopean video sequence synthesized by 4-state DP. The two input left and right sequences are not shown here.

Fig. 33. **Another example of virtual image synthesis in sequences.** Frames extracted from a reconstructed cyclopean sequence. The input images are not shown here. Notice the quality of the synthesized images.

virtual images generated from a pair of synchronized stereo sequences with large disparities. In this paper we have focused on one-to-one teleconferencing applications but the techniques are more general and can be employed in other fields requiring high-quality novel view generation and dense stereo.

The main contributions of the paper can be summarized as:

Fig. 34. **Background replacement.** Four-state DP allows, amongst other things, for the foreground to be segmented from the background. This, in turn, allows the real background to be replaced by alternative images, or videos.

- A new four-state DP algorithm for the correct detection and classification of occlusion events;
- A compact geometric technique for the rendering of novel views *directly* from the minimum-cost surface estimated by the DP algorithm.

The effectiveness of the new algorithmic components has been demonstrated in a number of examples where the artefacts typical of DP techniques have been eliminated while keeping quite a high frame rate. The current implementation exploits SSE2 instructions and produces virtual images at about 7 frames per second (on 320×240 images, on a $3.0Ghz$ Pentium IV with $1Gb$ RAM). The viability of the proposed algorithm has also been demonstrated by comparing the accuracy of the estimated occlusion maps with the ones generated by state of the art techniques amongst which three of the most recent graph-cut algorithms.

Despite recent progress, the depth maps obtained by 4-state DP still lack the level of accuracy necessary for seamless background substitution. Fusion of different cues, such as depth, motion, colour ans contrast seems very promising. Progress to date in this area is reported in [KCB$^+$05a, KCB$^+$05b].

Acknowledgements. The authors would like to thank G. Smyth, G. Cross, V. Kolmogorov, I. Cox, D. Scharstein, R. Szeliski, Y. Boykov, for their useful comments and inspiring discussions.

Notes

1. *e.g.* `messenger.msn.co.uk/, messenger.yahoo.com/, www.aol.co.uk/aim/`
2. We refer to *cyclopean view* as the image generated from a virtual camera located in the mid-point between the two input cameras.
3. The *minimum-cost surface* is defined to be the collection of all the minimum-cost paths estimated (independently) by the DP algorithm at each scanline.
4. We used the epipolar rectification technique described in [HZ00].
5. `www.idiom.com/~zilla/Work/nvisionInterface/`
6. Original algorithm available from `www.cs.cornell.edu/People/vnk/software.html`
7. Note that we had to adapt the source code in [KZ01] to read our filtered cost space as input. Then, graph-cut was used for energy minimization only.
8. Equation 11 is strictly valid only for matched pixels; while values of occluded pixels are taken only from the image where they are visible (fig. 7b).

References

[BGCM02] C. Buehler, S. Gortler, M. Cohen, and L. McMillan. Min surfaces for stereo. In *Proc. Europ. Conf. Computer Vision*, Copenhagen, Denmark, May 2002.

[BM92] P.N. Belhumeur and D. Mumford. A Bayesian treatment of the stereo correspondence problem using half-occluded regions. In *IEEE Comp. Soc. Conf. on Comp. Vision and Pattern Recognition*, pages 506–512, 1992.

[CHRM96] I. J. Cox, S. L. Hingorani, S. B. Rao, and B. M. Maggs. A maximum-likelihood stereo algorithm. *Computer Vision and Image Understanding*, 63(3):542–567, May 1996.

[COL93] I. Cox, M. Ott, and J.P. Lewis. Videoconference system using a virtual camera image. *US Patent*, 5,359,362, 1993.

[CSBT03] A. Criminisi, J. Shotton, A. Blake, and P. Torr. Gaze manipulation for one-to-one teleconferencing. In *Proc. International Conference on Computer Vision*, Nice, Oct 2003.

[CW93] E. Chen and L. Williams. View interpolation for image synthesis. In *SIGGRAPH*, pages 279–288, 1993.

[GTZ$^+$00] J. Gemmell, K. Toyama, C. Zitnick, T. Kang, and S. Seitz. Gaze awareness for video-conferencing: A software approach. *IEEE Multimedia*, 7(4), 2000.

[HD98] H.Ishikawa and D.Geiger. Occlusions, discontinuities, and epipolar lines in stereo. In *European Conference on Computer Vision*, pages 232–248, Freiburg, Germany, 1998.

[HZ00] R. I. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, ISBN: 0521623049, 2000.

[KCB$^+$05a] V. Kolmogorov, A. Criminisi, A. Blake, G. Cross, and C. Rother. Bi-layer segmentation of binocular stereo video. In *Computer Vision and Pattern Recognition (CVPR)*, San Diego, 2005.

[KCB$^+$05b] V. Kolmogorov, A. Criminisi, A. Blake, G. Cross, and C. Rother. Probabilistic fusion of stereo with color and contrast in bi-layer segmentation. Technical Report MSR-TR-2005-80, Microsoft Research Cambridge, UK, 7 J J Thomson Ave, 2005.

[KZ01] V. Kolmogorov and R. Zabih. Computing visual correspondence with occlusions using graph cuts. In *International Conference on Computer Vision*, pages II:508–515, Vancouver, Canada, 2001.

[KZ02] V. Kolmogorov and R. Zabih. Multi-camera scene reconstruction via graph cuts. In *Proc. Europ. Conf. Computer Vision*, pages 82–96, Copenhagen, Denmark, May 2002.

[OK85] Y. Ohta and T. Kanade. Stereo by intra- and inter-scanline search. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 7(2):139–154, 1985.

[Sch99] D. Scharstein. *View Synthesis Using Stereo Vision*, volume 1583 of *Lecture Notes in Computer Science (LNCS)*. Springer-Verlag, 1999.

[SS02] D. Scharstein and R. Szeliski. A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. *Int. J. Computer Vision*, 47(1–3):7–42, 2002.

[SSZ02] J. Sun, H. Y. Shum, and N. N. Zheng. Stereo matching using belief propagation. In *Proc. Europ. Conf. Computer Vision*, Copenhagen, Denmark, May 2002.

[Sze99] R. Szeliski. Prediction error as a quality metric for motion and stereo. In *Proc. Int. Conf. on Computer Vision*, pages 781–788, Kerkyra, Greece, 1999.

[Vet98] T. Vetter. Synthesis of novel views from a single face image. *Int. J. Computer Vision*, 28(2):103–116, 1998.

[YZ02] R. Yang and Z. Zhang. Eye gaze correction with stereovision for video tele-conferencing. In *Proc. Europ. Conf. Computer Vision*, volume 2, pages 479–494, Copenhagen, Denmark, May 2002.