Estimating Disparity and Occlusions in Stereo Video Sequences

Oliver Williams
Dept. of Engineering
University of Cambridge, UK

Michael Isard
Microsoft Research
Mountain View, CA

John MacCormick
Microsoft Research
Mountain View, CA

Abstract

We propose an algorithm for estimating disparity and occlusion in stereo video sequences. The algorithm defines a prior on sequences of disparity maps using a 3D Markov random field, and approximately computes the MAP estimate for the disparity sequence using loopy belief propagation. In contrast to previous work on temporal stereo, the algorithm (i) correctly models half-occlusions — scene points visible in one camera but not the other — and (ii) enforces the so-called "monotonicity constraint" on the boundary of half-occluded regions. The algorithm is also able to exploit temporal coherence more appropriately than many previous approaches to temporal stereo, by employing additional states in the Markov random field. These additional states permit rudimentary motion estimation to be performed as part of the belief propagation, thus improving the quality of temporal inference. Parameters of the algorithm are learned from the ground truth disparities of a real stereo sequence. Qualitative results are shown on real sequences, including comparisons with competing approaches, and the performance of the algorithm is assessed quantitatively using the ground truth data.

1. Introduction

The problem of reconstructing scene geometry from calibrated stereo image pairs has a 30-year history in computer vision[1]. The history of stereo reconstruction for *sequences* of stereo pairs is shorter (perhaps 10 years), yet still quite rich. However, it appears that previous approaches to temporal stereo have not attempted to model pixels that are "half-occluded" — that is, visible in one image and not the other. Furthermore, previous temporal stereo approaches ignore a very useful constraint, known as the *monotonicity constraint* (see Section 2), which precisely specifies the disparities permitted on the boundary of half-occluded regions. This paper presents an algorithm to infer disparity sequences with half-occlusions while enforcing the monotonicity constraint.

The algorithm is inspired by previous successful approaches to *static* stereo reconstruction using Markov random fields (MRFs). These approaches perform estimation by either graph cuts (e.g. [3]) or belief propagation (e.g. [21]). We concentrate here on belief propagation, which generalizes naturally to the temporal domain. A notable special case of the belief propagation approach to stereo is the 3-plane model (3PM) of Criminisi *et al.* [6], which builds on earlier ideas employing dynamic programming for stereo [4, 15, 10, 1]. We single out 3PM here for two reasons: (i) recent papers have demonstrated that 3PM is fast and accurate, especially when certain post-processing steps are applied [5]; and (ii) 3PM is the primary inspiration for our new algorithms. 3PM correctly models half-occlusions and the monotonicity constraint, and introduces a powerful new capability: modeling of non-fronto-parallel surfaces. Our algorithm retains these properties, while overcoming a potential disadvantage of 3PM, which treats all scanlines independently: this can result in poor accuracy unless care is taken with separate pre- and post-processing stages.

The paper's main contribution is a hierarchy of three MRF models with good performance for temporal stereo. First, we show how to remove the inter-scanline independence assumption from 3PM, resulting in a 2D MRF that retains the advantages of explicit occlusion labeling, non-fronto-parallel surfaces, and the monotonicity constraint. Second, we extend the MRF in a similar fashion by introducing dependencies between frames. This results in a 3D MRF which exploits both inter-scanline and temporal coherence. Third, we further extend the MRF by introducing a new binary state ("in motion" vs "stationary") at each pixel. This primitive motion estimation permits improved constraints on stationary regions which increase the benefits of temporal coherence. The temporally-coherent models are formulated and compared in both filtering and smoothing contexts, and their results compared with the best known competitors: graph cuts, and 3PM with pre-processing.

In addition to these theoretical developments, the paper makes an important practical contribution: as far as we are

[1] However, as Belhumeur [1] points out, biological evolution solved this problem millions of years ago, and humans have studied the concept of stereo vision since at least the time of Leonardo da Vinci.

aware, it provides the first ever hand labelled "ground truth" data for a real (i.e. non-synthesized) stereo sequence. We use this labelled data to provide hard numbers on our algorithms' performance. Our labelled data will be made available to the research community.

The final significant aspect of the paper is that all MRF parameters are determined by learning. We propose a learning methodology inspired by Freeman *et al.* [9], which encompasses transitions between left-, right- and un-occluded pixels in the temporal and spatial domains, in addition to the more familiar transitions between disparity levels.

1.1. Related work

Disparity estimation algorithms can be categorized according to the type of coherence they impose on their disparity maps (horizontal *i.e.* intra-scanline, vertical *i.e.* inter-scanline, or temporal *i.e.* inter-frame), and the extent to which they model occlusions (do they incorporate occlusions at all, and if so do they enforce the monotonicity constraint described in Section 2?). Figure 1 gives examples of early and/or significant work in these categories, and indicates the contributions we believe are made by this paper. To be specific, we believe this paper is the first to address both vertical and temporal coherence with the monotonicity constraint enforced correctly. Furthermore, this paper introduces motion modeling to assist with temporal coherence.

	Type of coherence		
	Horizontal	Horizontal & vertical	Horizontal, vertical, & temporal
No occlusions	Cox96 (DP)	Ohta85 (DP++), Sun02 (BP)	Leung04 (DP++)
Occlusions		Belhumeur-92 (DP++), Kolmogorov-01 (GC)	
Occlusions & monotonicity constraint	Geiger95 (DP), Criminisi03 (DP)	This paper (BP)	This paper (BP)

Figure 1. Previous Work.*Early and/or significant contributions in each category of disparity estimation are given, together with the chief algorithm used: dynamic programming (DP), dynamic programming with additional processing (DP++), graph cuts (GC), belief propagation (BP).*

There is, of course, much work on temporal stereo that defies the simple taxonomy of Figure 1. For example, Zhang *et al.*[25], Strecha and van Gool [19], and Davis *et al.*[7] are mostly suitable for use in static scenes and/or structured light. Other approaches rely on effective preprocessing to obtain good estimates of higher-level scene characteristics. For example, Vedula *et al.*[23] employ optic flow, while Shao [18] employs line and edge information.

2. Background

The aim of this work is to recover the 3D structure of a scene from stereo image sequences by finding the *disparity* of points in the scene. When a scene point is projected into a left and a right image, the disparity is defined as the distance between the projections. For this paper, the disparity is measured with reference to the *cyclopean* image. This is an image with an optical centre in the middle of the optical centres of the left and right input images. Also, all inputs are rectified to align scanlines [12]: a point will project to the same vertical position in the left, right and cyclopean images.

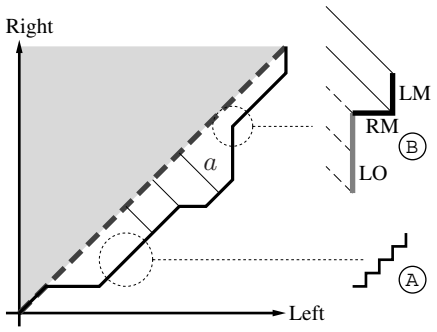


Figure 2. Scanline matching diagram.*This diagram shows the* matching surface *for a single scanline. (A) Parts of the image at constant depth are represented using alternating horizontal and vertical "moves". (B) Pixels that are half-occluded are explicitly labelled as such. The pixels that are not visible in the left scanline and are labelled as* left-occluded *(LO) before returning to* right-matched *(RM) and* left-matched *(LM) moves where the points are visible in both inputs. For the points marked as occluded, a* virtual disparity *can be measured (shown dashed).*

Figure 2 shows a diagram of the type introduced by [10]. Corresponding scanlines from the left and right images are used as axes on which a *matching surface* is drawn. Each point on the matching surface corresponds to a single point in the scene and its horizontal position in the two input images can be read off the axes. We refer to the 45° line (thick dashed) as the *cyclopean line* because the horizontal position of a point in the cyclopean image is found by diagonally projecting the matching surface onto this line. The disparity is proportional to the distance of this projection a.

Each scanline is composed of discrete pixels and the matching surface is therefore piecewise linear between points on a grid. We call these linear components *moves*.

After [5], we restrict the possible moves to horizontal and vertical. A planar object in the scene that is parallel to the image planes should correspond to a portion of the matching surface that is at 45° (constant disparity). By restricting the moves, such portions are approximately modelled as alternating horizontal and vertical moves (see Figure 2). Planar objects that are not parallel to the image planes will have matching surfaces at different angles comprising either more horizontal or more vertical steps. Another reason for a prolonged sequence of horizontal or vertical moves is a depth discontinuity in the scene. In such cases, background pixels appearing in one input image are occluded by foreground in the other. Again, we follow [5] and explicitly label such moves as "left-occluded" or "right-occluded". For such pixels, disparity is undefined, however we introduce the notion of *virtual disparity* which equals the true disparity for matched points, and the diagonal distance to the cyclopean line for occluded ones (Figure 2). Note that (provided the well-known ordering constraint holds), there is only one way to terminate a run of half-occluded pixels: a *matched* move perpendicular to the run. This exactly specifies the disparity value on the boundary of a half-occluded region. Because this fact is equivalent to the requirement that the matching surface is a monotonic function of both the left and right pixel locations, Geiger *et al.* [10] call this the "monotonicity constraint". 3PM-like algorithms exploit these powerful constraints along epipolar lines to achieve computational efficiency.

The cyclopean image is defined as having the same horizontal resolution as the inputs. As this corresponds to the 45° line in Figure 2, each cyclopean pixel will encompass two moves on the matching surface. Because of monotonicity, only three two-move possibilities are allowed: right-up, right-right and up-up. These correspond to changes in (virtual) disparity of 0, $+2$ and -2 along epipolar lines.

To establish the dense stereo task as an inference problem, we introduce hidden nodes at every pixel location in the cyclopean image. The problem is then a matter of labelling a node $i \in \mathcal{I}$ with a label x_i representing the (virtual) disparity at that point and whether it is visible in both inputs or just one. Hidden nodes are then "connected" to one another, encoding the spatial (and temporal) relationships between them. Hidden nodes x are also connected to visible ones, y which take measurements from the input data to support proposed labellings.

3. Problem formulation

Consider a single stereo pair of images with no temporal data. The spatial neighbourhood for a hidden node x_1 in the cyclopean image is illustrated in Figure 3a where it is modelled as dependent on its four neighbours only. In addition to this, each hidden node is singly connected to an observable node capturing measurements from real data. This results in a Markov random field [11] consisting solely of pairwise cliques and the joint distribution over the hidden nodes (given data y) may be written

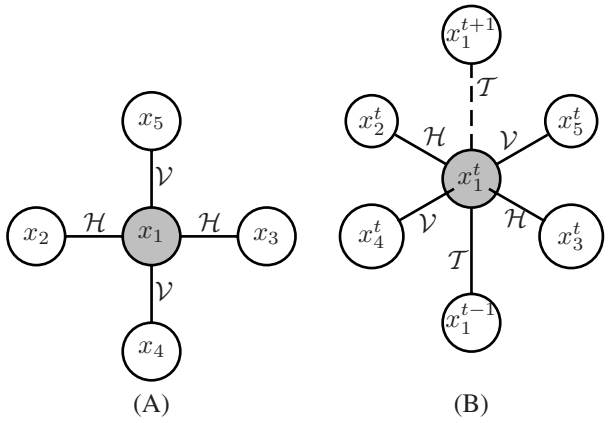


Figure 3. MRF neighbourhoods.*(A) Spatial neighbourhood: circles represent hidden nodes; lines show the cliques of which x_1 is a member. (B) Spatiotemporal neighbourhood.*

$$
\begin{aligned}
P(x|y) &\propto P(y|x)P(x) \\
&\propto \prod_{i\in\mathcal{I}} \Phi(x_i; y) \times \prod_{i,j\in\mathcal{H}} \Psi_h(x_i, x_j) \times \prod_{i,j\in\mathcal{V}} \Psi_v(x_i, x_j),
\end{aligned}
\tag{1}
$$

where Ψ_h is a *compatibility function* between two hidden nodes connected horizontally in the image, and Ψ_v is for those connected vertically. Φ is a term representing cliques between a hidden node and an observable node.

Now consider a sequence of images in which information is to be passed from nodes at one time step to adjacent time steps. Figure 3b shows this spatiotemporal neighbourhood of a node x_1^t. The state of this node is modelled as being dependent on four spatial neighbours (as it was in Figure 3a), and two temporal neighbours, one time-step into the future and the past. The temporal neighbours are in the same spatial location as x_1^t. As in Equation (1), we may write a joint distribution for all hidden nodes x, across all locations and all times, similarly factorized into data terms Φ and compatibility functions Ψ:

$$
\begin{aligned}
P(x|y, \mathcal{M}_s) &\propto P(y^t, \ldots, y^1|x^t, \ldots, x^1)P(x^t, \ldots, x^1) \\
= \prod_{\tau=1}^{t} \Bigg[\prod_{i\in\mathcal{I}} \Phi(x_i^\tau; y^\tau) \prod_{i,j\in\mathcal{H}} \Psi_h(x_i^\tau, x_j^\tau) \prod_{i,j\in\mathcal{V}} \Psi_v(x_i^\tau, x_j^\tau) \times \\
\prod_{i\in\mathcal{I}} \Psi_t(x_i^\tau, x_i^{\tau-1}) \Bigg],
\end{aligned}
\tag{2}
$$

where Ψ_t is the compatibility function for temporal cliques.

By marginalizing the joint distribution in Equation (2), an expression may be derived for the hidden variables in the last frame only:

$$P(x^t|y,\mathcal{M}_f) = \sum_{x^1,\ldots,x^{t-1}} P(x^t,\ldots,x^1|y^t,\ldots,y^1)$$
$$= \prod_{i\in\mathcal{I}} \Phi(x_i^t;y^t) \prod_{i,j\in\mathcal{H}} \Psi_h(x_i^t,x_j^t) \prod_{i,j\in\mathcal{V}} \Psi_v(x_i^t,x_j^t)\times$$
$$\prod_{i\in\mathcal{I}} \sum_{x_i^{t-1}} \Psi_t(x_i^t,x_i^{t-1}) P(x_i^{t-1}|y^{t-1},\ldots,y^1). \quad (3)$$

We call this the *filtering* model as it infers the state at time t under the assumption that nodes in the past are conditionally independent of nodes in the future. In contrast to this, we will refer to Equation (2) as the *smoothing* model where past nodes are not independent of future ones. $\mathcal{M}_s$ and $\mathcal{M}_f$ imply that the smoothing or filtering model is in use.

3.1. Specifying MRF parameters

In the cyclopean image, every location must be labelled with a disparity if it is visible in both input images, or as occluded (with virtual disparity). If the maximum anticipated disparity is Δ pixels, the space of possible labels $\mathcal{X}$ is

$$x_i \in \mathcal{X} \equiv \{d_0,\ldots,d_\Delta,l_0,\ldots,l_\Delta,r_0,\ldots,r_\Delta\} \quad (4)$$

where d_n indicates visibility in both images with a disparity of n pixels and l_n/r_n indicates occlusion in the left/right image with a virtual disparity of n.

Recall from Section 2 that our (virtual) disparities change by either 0 or ± 2 between neighbouring cyclopean pixels, and that there is only one possible change in disparity when transitioning from a matched to occluded state. If node i is the left neighbour of node j, these facts are encapsulated by the following parameterization of the (unnormalized) horizontal compatibility function:

$\Psi_h(x_i,x_j)$	x_i			
x_j		d_n	r_n	l_n
	d_n	1	0	0
	d_{n+1}	$e^{-\gamma}$	$e^{-\beta}$	0
	d_{n-1}	$e^{-\gamma}$	0	$e^{-\beta}$
	r_{n+1}	$e^{-\beta}$	$e^{-\alpha}$	0
	l_{n-1}	$e^{-\beta}$	0	$e^{-\alpha}$

. (5)

$\Psi_h(x_i,x_j) = 0$ for all other inputs not listed above. There are fewer constraints on the possible disparity values between vertical neighbours, and thus Ψ_v has a more general form:

$$\Psi_v(x_i,x_j) = \begin{cases} e^{-q_v(\|x_i-x_j\|)} & x_i,x_j \text{ both matched} \\ e^{-\kappa_v} & x_i,x_j \text{ both occluded} \\ e^{-\epsilon_v} & \text{one occ., one matched} \end{cases} \quad (6)$$

Ψ_t takes the same form as Ψ_v with function q_t and constants κ_t and ϵ_t.

Data is introduced via a 1D matching score between left and right pixels. For a proposed pair of matching points, the normalized sum of squared differences (NSSD) [17] is computed for a 3×3 region $\Omega \subset \mathbb{Z}^2$ around the points $p_l = [U(i)+D(x_i^t),V(i)]$ in the left image and $p_r = [U(i)-D(x_i^t),V(i)]$ in the right:

$$s(y_l^t,y_r^t;x_i^t) = \frac{\sum_{\delta\in\Omega}\Big([y_l(p_l+\delta)-\bar{y}_l(p_l)]-[y_r(p_r+\delta)-\bar{y}_r(p_r)]\Big)^2}{2\sum_{\delta\in\Omega}\Big([y_l(p_l+\delta)-\bar{y}_l(p_l)]^2+[y_r(p_r+\delta)-\bar{y}_r(p_r)]^2\Big)}, \quad (7)$$

where $\bar{y}_{l/r}(p_{l/r})$ is the mean intensity over the patch of size Ω centred at $p_{l/r}$. The data term Φ is then defined in terms of this score:

$$\Phi(x_i;y^t) = \begin{cases} e^{-f(s(y_l^t,y_r^t;x_i^t))} & x_i \text{ matched} \\ e^{-\rho} & \text{otherwise} \end{cases} \quad (8)$$

3.2. Approximate learning of MRF parameters

There are 7 constants and 3 functions defining the data term and compatibility functions. These parameters may be set by hand, but a more pragmatic approach is to learn them from data. We use the approximate algorithm described in [9] since exact learning in a loopy MRF is challenging. Section 5 describes a hand-labelled video sequence. By treating each clique independently, statistics were gathered from the labelled data and these were used to set the model parameters. Figure 4 shows the shape of the functions q_v and q_t. The unusual shapes can be attributed to the planar structure from which they were learnt: objects are mostly smooth but there are occasional large jumps in disparity. Given more training data we expect to observe a wider range of discontinuities in disparity and so the curve shown as a dotted line in Figure 4 was fitted which has a more intuitive form: as noted in Section 5 the inference is somewhat more reliable with this fitted form. In addition, this truncated linear form would allow significant computational savings and this is discussed in Section 6. The form of the data cost $f(s(y_l^t,y_r^t;x_i^t))$ is shown in Figure 5.

3.3. Allowing for motion

We extend the basic smoothing and filtering models to explicitly model moving objects by introducing a new set of binary hidden variables g, where $g_i = 0/1$ implies that the point corresponding to position i has/has not moved since the last time step. A version of Equation (2) may now be

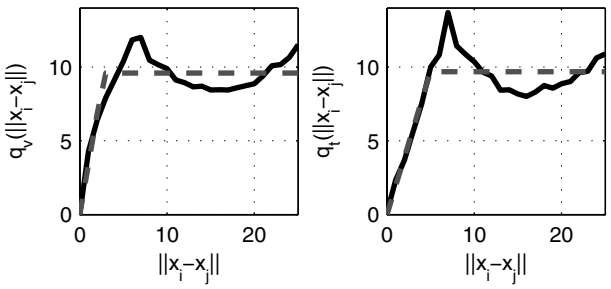


Figure 4. Learnt compatibility functions.*(left) function q_v defines the compatibility function when i and j are connected vertically; (right) q_t defines compatibility when i and j are temporal neighbours. These functions were learnt from the hand-labelled sequence described in Section 5. The dashed lines show least-squares fits of truncated linear models to these data.*

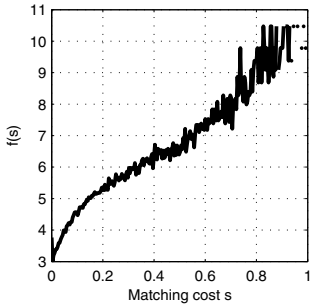


Figure 5. Data function f.

written taking g into account:

$$P(x|y,\mathcal{M}_{s+}) \propto \sum_{g^t,\ldots,g^1} P(y|x,g)P(x|g)P(g)$$
$$\propto \prod_{\tau=1}^{t}\left[\prod_{i,j\in\mathcal{H}}\Psi_h(x_i^\tau,x_j^\tau)\prod_{i,j\in\mathcal{V}}\Psi_v(x_i^\tau,x_j^\tau)\times\right.$$
$$\left.\prod_{i\in\mathcal{I}}\sum_{g_i^\tau}P(g_i^\tau)P(y^\tau|y^{\tau-1},x_i^\tau,g_i^\tau)\Psi_t'(x_i^\tau,x_i^{\tau-1};g_i^\tau)\right]. \quad (9)$$

The spatial compatibility functions, Ψ_v and Ψ_h are the same as previously, however the data term and temporal compatibility function now depend on the motion flag, and there is a prior over the motion labels $P(g)$, which are treated as mutually independent. This allows the match between the image data at adjacent frames to directly influence the amount of temporal smoothing.

The data term is now modelled as:

$$P(y^\tau|y^{\tau-1},x_i^\tau,g_i^\tau) \propto \Phi(x_i^\tau;y^\tau)\Gamma(g_i^\tau;y^\tau,y^{\tau-1}) \quad (10)$$

where Φ only considers how well the data at time τ correspond to the given disparity. If the node is labelled as stationary ($g_i^\tau = 0$) then the pixels indicated by x_i^τ should be similar in appearance to the same spatial locations in the previous time step, and this is captured by a new function Γ. The temporal compatibility is likewise extended to capture the fact that if a point has not moved, it will have exactly the same disparity as previously:

$$\Psi_t'(x_i^\tau,x_i^{\tau-1};g_i^\tau) = \begin{cases} \Psi_t(x_i^\tau,x_i^{\tau-1}) & \text{if } g_i=1 \\ \delta(x_i^\tau - x_i^{\tau-1}) & \text{if } g_i=0 \end{cases} \quad (11)$$

where δ is the Dirac delta function.

As before, Equation (9) can be marginalized according to the filtering model to give:

$$P(x^t|y,\mathcal{M}_{f+}) = \prod_{i,j\in\mathcal{H}}\Psi_h(x_i^t,x_j^t)\prod_{i,j\in\mathcal{V}}\Psi_v(x_i^t,x_j^t)\times$$
$$\prod_{i\in\mathcal{I}}\Phi(x_i^t;y^t)\left((1-\Gamma_i^\tau)P(x_i^{t-1}=x_i^t|y^{t-1},\ldots,y^1)P(g_i)+\right.$$
$$\left.\Gamma_i^\tau P(\bar{g}_i)\sum_{x_i^{t-1}}\Psi_t(x_i^t,x_i^{t-1})P(x_i^{t-1}|y^{t-1},\ldots,y^1)\right). \quad (12)$$

where $\Gamma_i^\tau = \Gamma(g_i^\tau = 1;y^\tau,y^{\tau-1})$. $\mathcal{M}_{s+}$ and $\mathcal{M}_{f+}$ denote the smoothing and filtering models with motion detection.

4. Loopy belief propagation

Having defined a Markov network in Section 3, it is necessary to perform inference on it and thereby estimate the underlying scene structure given some data. For networks without loops, message-passing rules can be used to compute MAP and MMSE estimates at each node [16, 24]. Even though the MRF defined above *does* contain loops, it has been shown (e.g. in [24, 8]) that satisfactory approximate results are still achieved if the loops are ignored and messages are passed as if the loops were not there. We refer to this algorithm as *loopy belief propagation* and restrict ourselves to the MAP formulation.

The algorithm works by computing messages from a node i to one of its neighbours (node j) using the following formula:

$$m_{ij}(x_j) = \max_{x_i}\left(Q_2(x_i,x_j)Q_1(x_j)\prod_{k\in N(i)\setminus j}m_{ki}(x_i)\right). \quad (13)$$

where $N(i)$ is the set of all nodes in the neighbourhood of i. Q_1 is the composition of all single node terms appearing in the joint distribution; likewise Q_2 is for two-node terms. Once the messages have converged, we can compute the *belief* at each node:

$$B(x_i) = Q_1(x_i)\prod_{k\in N(i)}m_{ki}(x_i). \quad (14)$$

A point estimate for the state of a node i can be found as the value of x_i that maximizes $B(x_i)$.

The forms of Q_1 and Q_2 depend on whether inference is taking place under the smoothing or filtering temporal model and whether motion detection is being used. In any case $Q_2(x_i, x_j) = \Psi_h, \Psi_v$ or Ψ_t depending on the relative orientation of i and j[2]. The one-node term for the smoothing models is simply the data term, $Q_1(x_i; \mathcal{M}_s) = \Phi(x_i; y)$. In the filtering model, Q_1 includes the prediction from the previous time-step:

$$Q_1(x_i^t; \mathcal{M}_f) \approx \Phi(x_i^t; y^t) \sum_{x_i^{t-1}} \Psi_t(x_i, x_i^{t-1}) B(x_i^{t-1}) \quad (15)$$

$$Q_1(x_i^t; \mathcal{M}_{f+}) \approx \Phi(x_i^t; y^t) \Big((1 - \Gamma_i^t) B(x_i^t) + \Gamma_i^t \sum_{x_i^{t-1}} \Psi_t(x_i, x_i^{t-1}) B(x_i^{t-1}) \Big) \quad (16)$$

Notice that the belief $B(x_i^{t-1})$ is being used in place of the marginal reverse-time posterior $P(x_i^{t-1} | y^{t-1}, \ldots, y_1)$. The belief does not have a simple mathematical interpretation in terms of the marginal posterior, however we use it as a convenient proxy.

In this paper, messages are computed according to the "accelerated" update schedule proposed in [22] for which all messages are updated in-place and are put to use immediately in subsequent evaluations of (13). A complete iteration involves passing all right-going messages, traversing the grid from the leftmost to the rightmost nodes and then repeating for leftward, upward and downward messages. If the smoothing model is in place, messages are similarly passed in the future and past directions where (abusing notation) the i and j nodes in Equation (13) are temporal neighbours. Iterations are then repeated until convergence (in this paper a fixed number of 10 iterations is used). As discussed in Section 6, by judicious choice of MRF parameters we expect in future to be able to take advantage of significant computational speedups [8].

5. Results

To evaluate the performance of the new stereo algorithms, a simple "ground truth" sequence was captured using a stereo camera. Two stills from the sequence are shown in the first row of Figure 6. As the scene is entirely composed of (approximately) planar surfaces, it was possible to estimate the ground-truth disparities and occlusions by hand-labelling the corners of the constituent planes in both right and left input images. The outer regions of the images, which contain no interesting texture, are deliberately left undefined. Two renderings of the hand-labelled disparities from the cyclopean viewpoint are shown as the second row of Figure 6. The sequence consists of 51 frames at a resolution of 320×240 pixels.

[2] In the case of smoothing with motion detection ($\mathcal{M}_{s+}$), $\Psi_t(x_i^\tau, x_i^{\tau-1})$ is replaced by the term $(1 - \Gamma_i^\tau)\delta(x_i^\tau - x_i^{\tau-1}) + \Gamma_i^\tau \Psi_t(x_i^\tau, x_i^{\tau-1})$

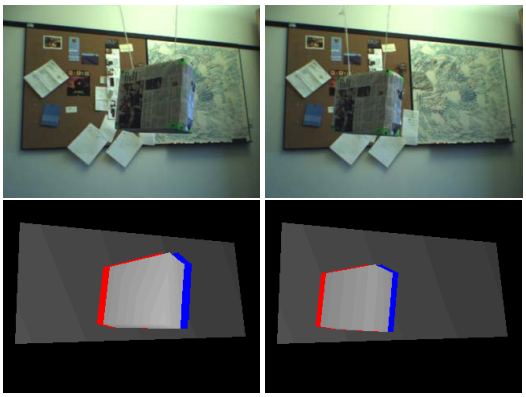


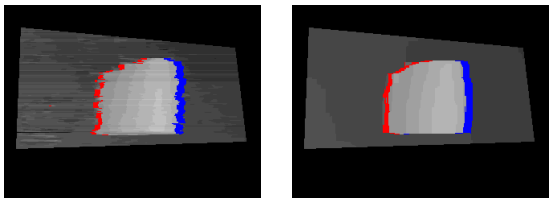
Figure 6. Ground-truth sequence.*Gray pixels are matched in both inputs (lighter indicates greater disparity). Left and right occluded points are shown as blue and red respectively.*

To compare the relative improvements afforded by the new models, Figure 7 shows the mean error in disparity estimates when tested on the ground-truth sequence, normalized such that the 3PM algorithm has an error of 1. For a single frame, there will be locations labelled as occluded by either the hand-labelling or the algorithm being tested and there are also undefined portions of the hand-labelled sequence. The error is therefore the mean absolute error in disparity estimates over all points in an image for which a disparity is available for both the algorithm output and the labelled sequence. As described in Section 3, the MRF model parameters were learnt from the labelled sequence, and for this the first 12 and last 12 frames of the sequence were used. Errors are therefore reported as the mean over the middle 27 frames.

Algorithm	Normalized error
Horizontal coherence (3PM)	1.0000
Hoz. & vert. coherence	0.0163
Filtering	
no motion detection ($\mathcal{M}_f$)	0.9220
motion detection ($\mathcal{M}_{f+}$)	0.0132
Smoothing ($\mathcal{M}_s$)	0.0127

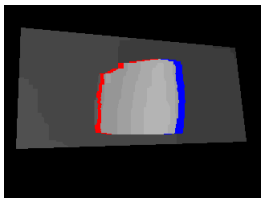
Figure 7. Normalized accuracy.*Five algorithms are compared. They are (i) the 3PM, which uses horizontal coherence only, (ii) a 2D MRF incorporating vertical coherence (Figure 3a), (iii,iv) The filtering algorithm with and without the motion flag and (v) the smoothing model. Errors are normalized so that the 3PM has an error of 1.*

The output of some of the algorithms in Figure 7 are

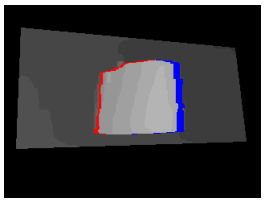


Horizontal coherence

Hoz. & vert. coherence



Filtering & motion



Smoothing

Figure 8. Output of four algorithms on ground-truth sequence.

shown as stills in Figure 8. While the hand-labelled data was invaluable for providing data to approximately learn the MRF parameters, it is "too easy" in that many of the algorithms perform equally well, with a very low error rate. We therefore also ran the algorithms on a more challenging sequence showing a ballet dancer (without ground-truth disparities) and stills comparing the output of the various algorithms are shown in Figure 9. The MPEG clips submitted as supplementary material give side-by-side comparisons of these algorithms. The addition of vertical coherence in the 2D MRF model affords the most dramatic improvement over the output of 3PM. Adding temporal predictions through filtering ($\mathcal{M}_f$) qualitatively improves the performance over the stationary parts of a sequence, but performs worse over the moving parts. This is because the future-going predictions rapidly tend to point estimates causing objects to "stick" to the background. By using simple motion detection ($\mathcal{M}_{f+}$), the moving parts are un-stuck leading to considerable improvements in the output. Solving the 3D MRF as a batch, smoothing problem ($\mathcal{M}_s$) gives the best overall performance as information is propagated in both future and past directions. However, adding motion detection to the smoothing algorithm ($\mathcal{M}_{s+}$) is less beneficial than in the filtering case; the reasons for this are currently under investigation.

The supplementary material also shows comparisons with two of the most successful approaches in previous work: (i) 3PM with additional preprocessing termed "cost-space smoothing" [6], and (ii) the graph cut approach of Kolmogorov and Zabih [13]. None of the approaches is a clear winner, and fair comparisons are difficult here. However, our temporal algorithms demonstrate a pleasing level of detail and accuracy, particularly on the stationary backgrounds where their stability is superior to 3PM-with-preprocessing, while capturing more detail than graph cuts.

Figure 10 contains a table of the time taken per frame by each of the algorithms tested. All experiments were conducted on a 3.2GHz Intel Pentium IV PC with 3GB of RAM, except the smoothing experiments which were carried out on a 64-bit machine with 16GB of RAM and a 2.2 GHz AMD Opteron processor.

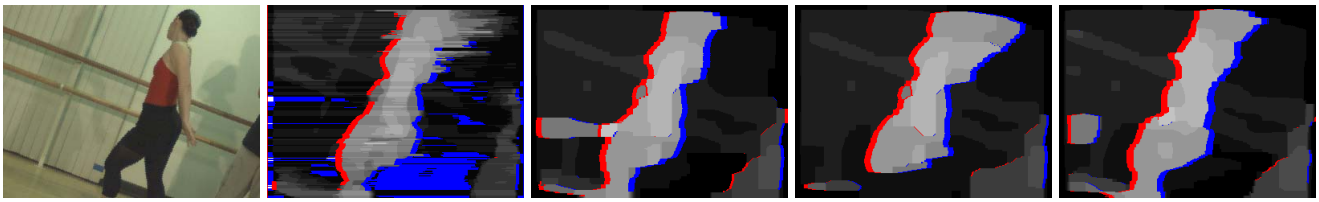
Algorithm	Time per 320×240 frame (s)
Horizontal coherence	1.9
Hoz. & vert. coherence	246.7
Filtering	
no motion detection ($\mathcal{M}_f$)	298.1
motion detection ($\mathcal{M}_{f+}$)	351.3
Smoothing	
no motion flag ($\mathcal{M}_s$)	*686.1
motion flag ($\mathcal{M}_{s+}$)	*947.5
Graph cuts	49.1

Figure 10. Algorithm run times.*Smoothing experiments were conducted on a different machine to the rest.*

6. Discussion and conclusions

We present a hierarchy of MRF models for temporal stereo including explicit occlusion labellings and the monotonicity constraint. We also present a preliminary step toward making full principled use of motion information in the computation of dense stereo. We investigate the simplest possible temporal model which assumes temporal smoothness everywhere and show that better results can be obtained by adding a binary "in motion"/"stationary" flag. In future work we anticipate that by explicitly modelling motion and disparity in the same MRF it should be possible to combine optic flow with dense stereo to the advantage of both. The difficult areas for optic flow come at the boundaries of moving objects, but in many cases where one camera sees a occluding boundary the other will see an unobstructed full patch of background due to parallax. A joint estimation of both cameras' flow fields and the disparity between them may provide significant gains in performance.

The running times presented in Figure 10 are somewhat slow compared with the state of the art [5, 8]. This is because our algorithm is general enough to use the learned MRF parameters shown in Figure 4. However, we demonstrate our best results using the fitted curve from that figure, and have chosen the analytic form of that curve so that it is amenable to the distance transform methods pioneered in vision by Felzenszwalb and Huttenlocher [8]. By combining this with the coarse-to-fine evaluation mechanism described in [8] the authors report several orders of magnitude speedup on a problem very similar to ours, and future work will verify that this is indeed attainable.



Input (left) Horizontal coherence Hoz. & vert. coherence Filtering & motion Smoothing

Figure 9. Output of four algorithms on ballet sequence.

Finally, we have used approximate learning methods from a very limited training set. We are actively seeking to gather more extensive and challenging ground-truth sequences, and plan to investigate more accurate learning approximations as discussed in [9].

References

[1] P. Belhumeur. A Bayesian approach to binocular stereopsis. *Int. J. Computer Vision*, 19(3):237–260, 1996.

[2] P. N. Belhumeur and D. Mumford. A Bayesian treatment of the stereo correspondence problem using half-occluded regions. In *Proc. Conf. Computer Vision and Pattern Recognition*, 1992.

[3] S. Birchfield and C. Tomasi. Multiway cut for stereo and motion with slanted surfaces. In *Proc. of the 7th Int. Conf. on Computer Vision*, pages 489–495, Sept. 1999.

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

[5] A. Criminisi and A. Blake. The SPS algorithm: Patching figural continuity and transparency by split-patch search. In *Proc. Conf. Computer Vision and Pattern Recognition*, 2004.

[6] A. Criminisi, J. Shotton, A. Blake, and P. Torr. Gaze manipulation for one-to-one teleconferencing. In *Proc. Int. Conf. on Computer Vision*, 2003.

[7] J. Davis, R. Ramamoorthi, and S. Rusinkiewicz. Spacetime stereo: A unifying framework for depth from triangulation. In *Proc. Conf. Computer Vision and Pattern Recognition*, volume 2, pages 359–366, 2003.

[8] P. Felzenszwalb and D. Huttenlocher. Efficient belief propagation for early vision. In *Proc. Conf. Computer Vision and Pattern Recognition*, 2004.

[9] W. Freeman, E. C. Pasztor, and O. T. Carmichael. Learning low-level vision. *Int. J. Computer Vision*, 40(1):25–47, 2000.

[10] D. Geiger, B. Ladendorf, and A. Yuille. Occlusions and binocular stereo. *Int. J. Computer Vision*, 14:211–226, 1995.

[11] S. Geman and D. Geman. Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 6(6):721–741, 1984.

[12] R. Hartley and A. Zisserman. *Multiple View Geometry*. Cambridge University Press, 2000.

[13] V. Kolmogorov and R. Zabih. Computing visual correspondence with occlusions using graph cuts. In *Proc. Int. Conf. on Computer Vision*, 2001.

[14] C. Leung, B. Appleton, B. C. Lovell, and C. Sun. An energy minimisation approach to stereo-temporal dense reconstruction. In *Proc. Int. Conf. on Pattern Recognition*, 2004.

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

[16] J. Pearl. *Probabilistic Reasoning in Intelligent Systems*. Morgan Kaufmann, Palo Alto, 1988.

[17] D. Scharstein and R. Szeliski. A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. *Int. J. Computer Vision*, 2002.

[18] J. Shao. Generation of temporally consistent multiple virtual camera views from stereoscopic image sequences. *Int. J. Computer Vision*, 47(1-3):171–180, 2002.

[19] C. Strecha and L. V. Gool. Motion-stereo integration for depth estimation. In *Proc. European Conf. on Computer Vision*, volume 2, pages 170–185, 2002.

[20] J. Sun, H.-Y. Shum, and N.-N. Zheng. Stereo matching using belief propagation. In *Proc. European Conf. on Computer Vision*, pages 510–524, 2002.

[21] J. Sun, N.-N. Zheng, and H.-Y. Shum. Stereo matching using belief propagation. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 25:787 – 800, July 2003.

[22] M. Tappen and W. Freeman. Comparison of graph cuts with belief propagation for stereo, using identical MRF parameters. In *Proc. Int. Conf. on Computer Vision*, 2003.

[23] S. Vedula, S. Baker, P. Rander, R. Collins, and T. Kanade. Three-dimensional scene flow. In *Proc. Int. Conf. on Computer Vision*, volume 2, pages 722–729, 1999.

[24] J. Yedidia, W. Freeman, and Y. Weiss. *Exploring Artificial Intelligence in the New Millennium*, chapter Understanding Belief Propagation and its Generalizations. Elsevier Science, 2003.

[25] L. Zhang, B. Curless, and S. M. Seitz. Spacetime stereo: Shape recovery for dynamic scenes. In *Proc. Conf. Computer Vision and Pattern Recognition*, volume 2, pages 367–374, 2003.