Dense Stereo Using Pivoted Dynamic Programming

P. H. S. Torr and A. Criminisi
Microsoft Research Ltd,
St George House, 1 Guildhall St,
Cambridge CB2 3NH, UK
philtorr@microsoft.com

Technical Report
MSR-TR-2002-51

Microsoft Research
Microsoft Corporation
One Microsoft Way
Redmond, WA 98052
http://www.research.microsoft.com

Abstract

This paper describes an improvement to the dynamic programming approach for dense stereo. Traditionally dense stereo algorithms proceed independently for each pair of epipolar lines, and then a further step is used to smooth the estimated disparities between the epipolar lines. This typically results in a streaky disparity map along depth discontinuities. In order to overcome this problem the information from corner and edge matching algorithms are exploited. Indeed we present a unified dynamic programming/statistical framework that allows the incorporation of any partial knowledge about disparities, such as matched features and known surfaces within the scene. The result is a fully automatic dense stereo system with a faster run time and greater accuracy than the standard dynamic programming method.

1 Introduction

Automatically generating 3D models from images is an on going topic of research. A successful image to model system has been developed by the research groups at Oxford [2] and Leuven [11]. The method proceeds as a set of independent modules: first features are extracted and matched, second projection matrices and calibration recovered, third a dense stereo algorithm based on dynamic programming is used to extract depths and finally a three dimensional model is constructed. It can be seen that this process involves two representations, one sparse and feature based, the other a dense depth map yielded by dynamic programming. Within this paper the dynamic programming algorithm for recovering the dense depth map is discussed and various improvements to its speed and accuracy are advocated, which utilize the results already obtained by the sparse feature matcher. It will be seen that this unique synthesis of the sparse and dense matching techniques leads to improved results and run times.

The problem of obtaining dense correspondence along pairs of corresponding epipolar lines may be solved using dynamic programming as an optimal path finding problem on a 2D plane. However there is no corresponding efficient method to impose a smoothness constraint between the epipolar lines. The solution along consecutive epipolar lines can vary significantly, creating artifacts across depth discontinuities and in homogeneous patches of intensity. An example of this is shown in figure 1. It can be seen that indeed the estimated disparity along the depth discontinuities is poor, despite being estimated by a method that attempts to enforce inter epipolar line consistency [5] (the epipolar lines in this case run roughly horizontally).

There have been a variety of proposals to solve the "streaky" artifact. Henderson [9] was the first to use dynamic programming to solve the structure from motion problem. He proposed a sequential approach to impose inter epipolar line constraints, starting at the top of the image and proceeding down through the epipolar lines using the result of the previous as a guide for the next. This method however suffers from an avalanche effect in the errors, in that small errors made early on can be magnified as the algorithm progresses.

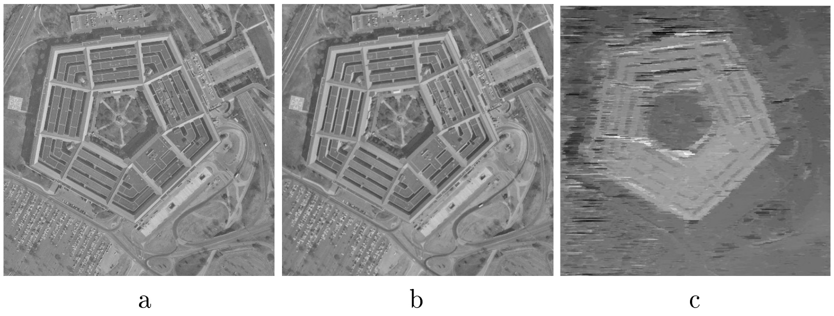

Figure 1: **(a)(b)** *The left and right images of the Pentagon standard test stereo pair obtained from the CMU VASC database (*`http://www.vasc.ri.cmu.edu/idb/`*).* **(c)** *the disparity map generated by the Cox MLMH+V algorithm, typical of the dynamic programming approach [5] (lighter shades indicate larger disparity). What this disparity map does not show is those pixels that are unmatched (occlusion), this will be discussed in more detail in Section 5. Note the lack of consistency between epipolar lines, which run horizontally.*

Baker and Binford [1] popularized dynamic programming for stereo, explaining it in terms of the Viterbi algorithm [6]. First using Viterbi to match edges, and then a second round of Viterbi to match pixels between edges. However their method does not adequately deal with occlusion and cannot recover if the edges are mismatched. A slightly different idea is put forward for edgel matching by Ohta and Kanade [13] who propose performing dynamic programming to solve a path planning problem in the product space of the epipolar lines. To impose consistency between epipolar lines they extend the dynamic programming to 3D. This furnished only a sparse representation in terms of edges however.

A problem with the above methods is that they do not deal with occlusion very well, and have no mechanism for detecting occluded pixels. A series of methods [3, 5, 7] that were developed roughly concurrently are all characterized by the modelling of an occlusion process together with a shift to estimate a per pixel disparity (as opposed to matching edges first as in [1, 13]). Cox *et al* [5] propose a purely maximum likelihood approach, whereas Belhumeur [3] and Geiger *et al* [7] prescribe a Bayesian philosophy. The other difference is algorithmic, all the methods can be cast as one of finding the best path through a graph, but the structure of the graphs are different. However, the methods all operate on each pair of epipolar lines independently and rely on further iterative steps to enforce smoothness constraints between epipolar lines.

Recently a new class of methods based on maximum flow/minimum cut algorithms has promised to generalize the dynamic programming approach to incorporate inter epipolar line constraints. Roy and Cox [14] introduced a maximum flow algorithm on an undirected graph for stereo, however as pointed out in [10] their method does not really generalize dynamic programming and does not model discontinuities and occlusions as all pixels are forced to have a match. Ishikawa and Geiger [10] propose a max flow algorithm on a directed graph with the aim of imposing constraints between epipolar lines, this approach seems interesting but there is still evidence of the streaky effect at depth discontinuities suggesting that the algorithm does not adequately enforce inter epipolar line constraints. These algorithms are also very computationally intensive when compared with the pair-wise epipolar line solutions.

None of these papers exploit external information that might already have been gathered. Within this paper we explore a class of methods that combines the sparse but accurate representations yielded by feature detectors and matchers with the dense representation yielded by dynamic programming.

This paper is laid out as follows: Section 2 explains the dynamic programming approach to dense stereo. Section 3 briefly describes the feature extraction and matching algorithms. It also suggests an improvement on the edge matching by making use of the results of the corner matching. Section 4 shows how the corner and edge information can be readily incorporated into the dynamic programming framework. A comparison of the disparity map computations for our methods and for that of Cox *et al* [5] is given in Section 5.

2 Dynamic Programming to solve stereo correspondence

Within this section the dynamic programming approach is described in detail, and it is observed that the difference between the various methods that have been previously proposed lies in the structure of the graph on which the path planning is performed. We then lay out a statistical framework so that additional constraints are easily incorporated (in Section 4).

All of the dynamic programming approaches to dense stereo described in the introduction can be thought of in terms of a path planning problem on a graph. There are two types of graph that are typically constructed differing in their connectivity. The first (espoused in [5, 7, 13]) is a graph formed on the product space of the two epipolar lines as shown in figure 2, the second (advocated in [1, 3, 9]) is a graph formed on the product space of one of the epipolar lines and the set of putative disparities. The two approaches lead to similar results and the differences are small, in this paper we explore the first in some detail and show how it can be improved, but the improvements are generic to both methods. The path defines a mapping between the two epipolar lines for each corresponding pairs of points e.g. points A and B as shown in figure 2a. Uniqueness and the ordering constraint (evoked by Cox) enforces the gradient of this path to be greater than or equal to zero, as does the monotonicity constraint of Geiger *et al* [7].

In the Cox *et al* formulation horizontal or vertical parts of the path correspond to occluded regions that are seen in one image but not the other (sometimes referred to as *half-occlusions*). However only diagonal moves can be made (fig. 2b) meaning that there is no effective way to represent expansion or contraction of the image. Geiger *et al* [7] allow for a wider range of changes in disparity between consecutive pixels together with a prior on the changes (fig. 2c). For the most part this is only important to represent sub-pixel disparity changes, however we found in our experiments that the increased computational time for the sub-pixel calculation was not worth the slight increase in the quality of the result.

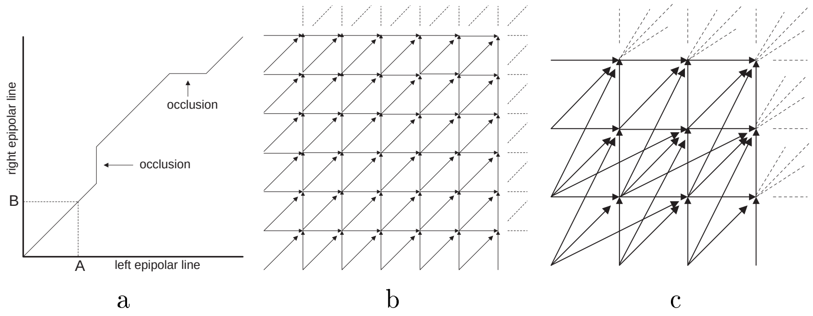

Figure 2: *Dense epipolar matching.* **(a)** *the path defines a matching between the two epipolar lines, e.g. point A is matched to point B. Horizontal and vertical arcs represent occlusions.* **(b)** *the connectivity of the graph of the Cox et al method, each vertex represents a putative matching of a left and right pixel and has three input arcs and three output allowing horizontal, vertical and diagonal moves.* **(c)** *the connectivity of the Geiger et al method, here more complex movements can be made.*

Matching Cost Cox *et al* use the squared difference of pixel intensity for his matching cost between two pixels, Koch [11] *et al* proposes using the cross correlation of a neighbourhood around the pixel as matching cost. Geiger *et al* proposes splitting the cross correlation window into two, one to the left of the pixel on the epipolar line one to the right, to take into account half occlusions. The effects of these different choices will be explored in the results section 5. Typically there is a constant cost for unmatched pixels, this can be explained in statistical terms as follows.

Statistical Formulation For ease of exposition it is first assumed that the two images have been rectified so that the epipolar lines are horizontal and have the same length. Given a pair of corresponding epipolar lines discretized into m pixels, feature vectors $\mathbf{l}$ and $\mathbf{r}$ are extracted for left and right epipolar line respectively, with the result indexed by the pixel: $l_i, r_i, i = 1 \ldots m$. Following [12] a matching process is defined δ_{ij}, such that $\delta_{ij} = 1$ if the ith pixel in the left view matches the jth pixel in the right, and $\delta_{ij} = 0$ otherwise. The matching process can be represented by a $m \times m$ matrix Δ with ijth element given by δ_{ij}. To ensure that each pixel has only one match each row or column of Δ must sum to 0 (no match) or 1 (match). The likelihood of generating the feature vectors $\mathbf{l}, \mathbf{r}$ given a matching Δ is defined to be

$$\Pr(\mathbf{l}, \mathbf{r} | \Delta) = \exp - \frac{\sum_{ij} \left(\delta_{ij} c(l_i, r_j) + (1 - \delta_{ij}) c_0 \right)}{\mu} \qquad (1)$$

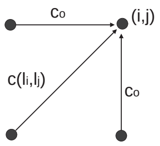

Figure 3: *The input to a node for the maximum likelihood model shown in figure 2b. Horizontal and vertical moves mean that a point is unmatched and are assigned cost c_0, the diagonal move means that the ith left pixel is matched to the jth right pixel with cost $c(l_i, r_j)$.*

where μ is a normalization constant, c_0 is the cost for occlusion [5], and $c()$ is the matching cost function (e.g. difference of pixel intensities or sum of squared differences). Different forms for $c()$ arise naturally from different assumptions about the statistical form of the errors, for instance maximizing normalized cross correlation for matching yields the maximum likelihood match if the scaling of intensities between images is unknown.

The match matrix Δ that maximizes (1) and satisfies the uniqueness and ordering constraints is the maximum likelihood matching. The optimal matching Δ can be found by finding the minimum cost path through the Cox *et al* graph shown in figure 2b, the costs on the arcs are assigned as in figure 3. By inspection it can be seen that the total cost of the minimal path is the negative log likelihood:

$$\text{Path Cost} = \sum_{ij} (\delta_{ij} c(l_i, r_j) + (1 - \delta_{ij}) c_0) + \text{constant} \tag{2}$$

In this formulation it appears that there is no smoothness prior as opposed to the Geiger *et al* formulation in which the smoothness of disparities along the epipolar line is made explicit as a Markov Random Field. However examination of the structure of the graph reveals that this is not the case. First disparity is defined. If $\delta_{ij} = 1$ then the disparity is defined to be on the left epipolar line $\gamma(i) = j - i$ and on the right epipolar line $\rho(j) = i - j$. Assume that the $(i-1)$th pixel in the left image is matched to the $(j-1)$th in the right, thus $\gamma(i-1) = j - i$; a diagonal move in the graph from $(i-1, j-1)$ to (i, j) corresponds to the consecutive pixels represented by the nodes having the same disparity $\gamma(i-1) = \gamma(i) = j - i$, with cost $c(l_i, r_j)$. For the next pixel to have one greater disparity this involves a move from $(i-1, j-1)$ to $(i-1, j+1)$, which can only be achieved by a vertical move followed by a diagonal move with cost $c(l_i, r_j) + c_0$ thus the change in disparity is penalized by c_0. Therefore, a change of k pixels in the disparity leads to a penalty of kc_0 which effectively encodes a smoothness constraint.

The second formulation of Henderson/Binford/Belhemeur lends itself to the Viterbi algorithm and has a slightly different structure. One image is considered as a principal image and the disparity is hypothesized for each pixel (or part thereof) along each epipolar line. There are a finite number of discrete disparities and also the possibility of the void disparity representing occlusion of that pixel in the next image. The structure of this graph is shown in figure 4.

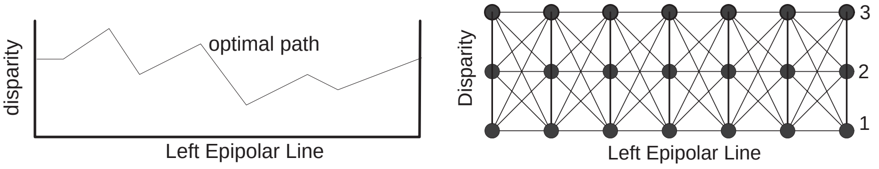

Figure 4: *Epipolar matching: Viterbi formulation. Left: a path is selected that gives a disparity per pixel. Right: the connectivity of the graph, each vertex gives a disparity for the corresponding pixel and has $k + 1$ inputs and outputs, where k is the number of disparities hypothesized (+1 to take into account the occlusion hypothesis).*

However all the dynamic programming algorithms that operate only on epipolar line pairs give similar sorts of results, failing to adequately enforce inter epipolar line constraints. In the next sections it will be shown how to use the output of the feature matchers to (a) enforce inter epipolar line constraints, (b) improve the accuracy of the depth map and (c) speed up the algorithm.

3 Feature extraction and matching

Within this section first it will be shown how to match features and then how the matched features can be used to improve the dense stereo. There are two types of features that are used in this paper: corners (points) and edges. Corners are extracted using the Harris corner detector [8] and then matched using cross-correlation, from this the fundamental matrix is estimated and the matches refined using the type of robust methods (RANSAC based) described in [17, 18, 19], this process can be extended to multiple views [2]. The corner matches for the pentagon stereo pair are shown in figure 5d.

Once the epipolar geometry is recovered Canny edges are extracted [4] in each image. Next the recovered epipolar geometry is used to match the Canny edges based on the curve matching algorithm of Schmid and Zisserman [16, 15]. This algorithm scores two curves that are putatively matched by cross-correlation of image intensities. The point-to-point correspondence between the curves is determined by the intersection of epipolar lines with the curves (figure 5a). For each edge all the edges within a search region in the next image are scored as candidate matches. The score is simply the sum of the correlation scores between points in correspondence along the curve divided by the length of the curve. The best correlating curve is taken as being matched if its score lies above a threshold.

3.1 Improvements to the edge matching

Within this section a heuristic is described that speeds up and improves the accuracy of the edge matching algorithm. The algorithm described in the previous section typically achieves a high number of correctly matched edges, but also many mismatches. In fact it fails when the image presents repeated structure

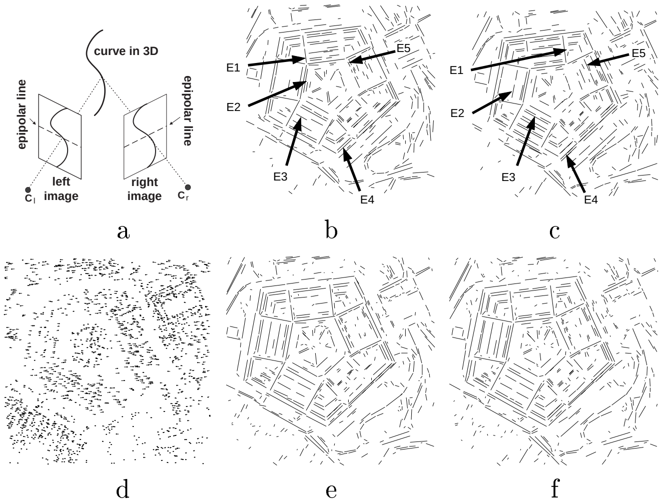

Figure 5: *Corner and edge matching in pairs of images.* **(a)** *A matching score is established for curves by correlating corresponding points on the curves, determined by the intersection of edge and corresponding epipolar lines. C_l and C_r are the camera centres.* **(b), (c)** *Erroneous matches caused by repeated structure using the algorithm of Schmid and Zisserman.* **(d)** *The Harris corner matches shown as disparity vectors (the matches are used to guide the edge matching and dense stereo, see text).* **(e), (f)** *The corresponding edges between the two images of the Pentagon, correctly computed by our guided matching algorithm. The matched edges are used to guide out dense matching algorithm, see text.*

(as in the pentagon case). The edge matches originated for the pentagon stereo pair by this algorithm are shown in figures 5b,c. Some of the mismatched edges are marked in the figure. This matching algorithm can also be unnecessarily slow if the motion between images is large, as there is a combinatorial explosion in the number of putative edge matchings that must be considered. As suggested in [16] the epipolar geometry can be used to reduce the search space. As shown in figure 6a the two end points $\mathbf{x}, \mathbf{y}$ of the curve, in the left image, generate two epipolar lines $\mathbf{l}_x, \mathbf{l}_y$ in the next image. These two lines define a region called an *epipolar beam* within which the corresponding edge must lie. Assuming known calibration so that the homography of the plane at infinity $\mathbf{H}_\infty$ can be computed then a search window in the shape of a parallelogram (the shaded region in figure 6a) can be defined with vertices given by $\mathbf{H}_\infty\mathbf{x}$, $\mathbf{H}_\infty\mathbf{y}$ and two other vertices lying on $\mathbf{l}_x$, $\mathbf{l}_y$ and defining the minimum distance of lines, from the cameras, that can be matched. However this still leaves a large number of putative matches to be correlated in some images, as the range of motions can be potentially great and any edge that is even partially contained in the beam must be considered as a candidate match.

As mentioned, in the Oxford/Leuven system and most other SFM systems, the output of each module is estimated separately, thus corner detection and matching has no direct bearing on edge detection and dense stereo. In keeping with our philosophy of integrating all the disparate parts of the algorithm we

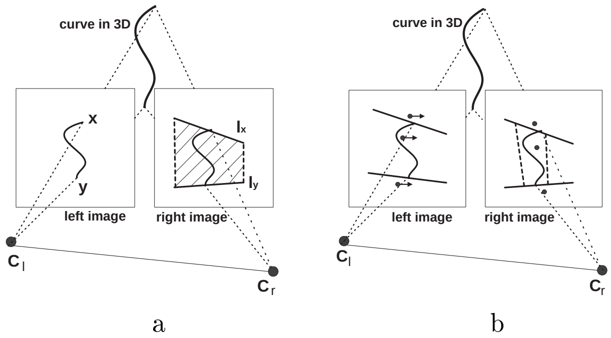

Figure 6: *Edge matching.* **(a)** *The search window (epipolar beam) suggested by Schmid and Zisserman [16] is shown in the left image as a shaded region.* **(b)** *The reduced search window determined using feature correspondences.*

propose that the corner matching algorithm should be used to aid the curve matching.

This has a two-fold effect: increasing the speed and the accuracy of the curve matching. The algorithm is as follows: the set of n (a user defined constant) matched corners nearest the curve are found either side of the curve; the disparity of each of these matches is examined, let the maximum and minimum value of the disparity be $\delta_{\min}$ and $\delta_{\max}$, and the distance of the corresponding features from the edge be $d_{\min}$ and $d_{\max}$. The distances $d_{\min}$ and $d_{\max}$ are recorded, as the further the matched corner is from the edge the less effect it should have on the size of the search region. The left image of figure 6b shows three such corner matches whose motion can be used to guide the search for the nearby edge. The maximum and minimum disparity can be used to reduce the area of the epipolar beam that needs to be included in the search as shown in the right image of figure 6b. The maximum disparity to be searched equals $\delta_{\max} + d_{\max}$ and the minimum $\delta_{\min} + d_{\min}$. Given a large number of matching features within the image this heuristic can often reduce the search region by in excess of 95% with a commensurate increase in speed of the edge matching process.

The reduced search range, estimated from the corner matches, helps reduce ambiguous edge matches. An example of this is shown in fig. 5e,f. Here far more correct edge matches are found than in the basic algorithm (fig. 5b,c).

4 Matched features to guide dense matching

The feature matching algorithms described in section 3 yield a sparse set of disparities for matched corner features and slightly less so for matched curves. This can be thought of as a prior distribution for those points and it can be combined with the likelihoods developed in section 2. In order to do this some new notation is needed. First the likelihood function (1) is rewritten in terms of disparities. Let Γ be the set of disparities $\gamma_1 \ldots \gamma_m$ for the left epipolar line, as some pixels are occluded the null disparity is defined: $\gamma_i = \emptyset$ if the ith pixel is unmatched ($\emptyset$ being adopted as the symbol for no match). Again, the disparities must conform to the uniqueness and ordering constraints so that finding an optimal Γ is equivalent to finding the optimal Δ i.e. $\Pr(\mathbf{l}, \mathbf{r}|\Delta) = \Pr(\mathbf{l}, \mathbf{r}|\Gamma)$. Thus (1) becomes

$$\Pr(\mathbf{l}, \mathbf{r}|\Gamma) = \exp - \frac{\sum_i \left(c(l_i, r_{\gamma(i)})\right)}{\nu} \tag{3}$$

with $c(l_i, r_{\gamma(i)}) = c_o$, if $\gamma(i) = \emptyset$, and ν being the normalization constant.

In the maximum likelihood formulation it is implicitly assumed that there is a uniform distribution on Γ, if we already have some indication of likely disparities from the feature matchers then this is not the case. A Bayesian estimate of Γ would be

$$\max_{\Gamma} \Pr(\Gamma|\mathbf{l}, \mathbf{r}) = \max_{\Gamma} \Pr(\mathbf{l}, \mathbf{r}|\Gamma) \Pr(\Gamma) \tag{4}$$

assuming that the γ_i are all independent, leading to

$$\Pr(\boldsymbol{\Gamma}|\mathbf{l}, \mathbf{r}) = \prod_i \exp - \frac{\left(c(l_i, r_{\gamma(i)})\right)}{\nu} \Pr(\gamma(i)) \ . \tag{5}$$

Modelling the $\gamma(i)$ as independent may not be correct as it ignores the smoothness constraint, however it has already been shown how the uniqueness and ordering constraints implicitly encode smoothness, so it may not be a bad model.

Next the prior distribution $\Pr(\gamma(i))$ is formulated. On the epipolar line if the ith pixel is by the feature matchers unmatched then there is prior knowledge about $\gamma(i)$ and a natural choice of prior is a uniform distribution on $\Pr(\gamma(i))$, i.e. if there are m possible disparities $\Pr(\gamma(i) = k) = (1 - \epsilon)\frac{1}{m}, k = 1 \ldots m$, and $\Pr(\gamma(i) = \emptyset) = \epsilon$, ϵ is the prior probability of there being no match for a pixel due to occlusion. Note that in the framework we shall propose any other prior could be used, for instance one that favours smaller disparities.

If one of the feature matchers has matched the ith pixel then this gives some information about its the disparity. How much information depends upon how accurate we believe the feature matching to be. This entails learning the error rate of the feature matcher, i.e. the probability that any given match generated by the algorithm is, in fact, incorrect. Suppose that the error rate for the corner matcher is λ_c and suppose that the corner matcher indicates that the ith pixel has disparity $\gamma(i) = p$; then the probability that the ith pixel is in fact correctly matched is $1 - \lambda_c$. This means that $\Pr(\gamma(i) = p) = 1 - \lambda_c$ and $\Pr(\gamma(i) \neq p) = \lambda_c$ the remaining probability is distributed uniformly amongst the other disparities: $\Pr(\gamma(i) = k) = (1 - \epsilon)\frac{\lambda_c}{m}, k = 1 \ldots m, k \neq p$, and $\Pr(\gamma(i) = \emptyset) = \frac{\epsilon \lambda_c}{m}$.

Taking account of this prior, the new cost for matching the ith pixel to jth pixel becomes $c_p(l_i, r_j) = c(l_i, r_j) - \log \Pr(\gamma(i) = j - i)$. Given this new matching cost dynamic programming can again be used to find the optimal matching path. In effect the cost for points that have already been matched

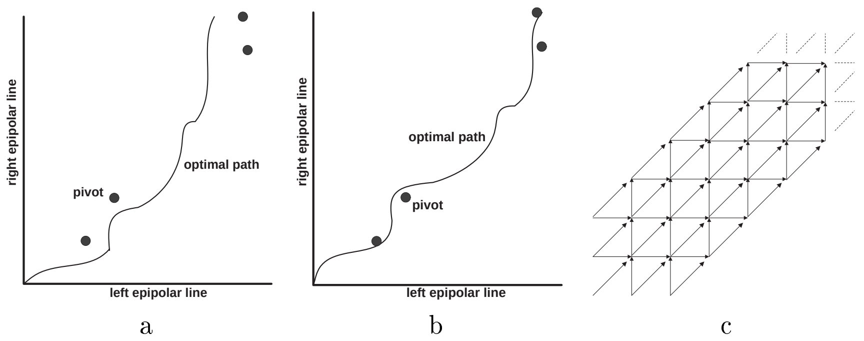

Figure 7: *The optimal path can be pivoted by the addition of some known corner and edge matches (*pivots*),* **(a)** *Optimal path before pivoting.* **(b)** *Optimal path after pivoting. The optimal path has been attracted towards the pivots, thanks to their lower matching costs. The optimal matching path does not necessarily go through the pivots.* **(c)** *The reduced graph computed for speed (cf. fig. 2b).*

by the corner or curve matcher is drastically reduced; this forces the optimal matching path to be attracted towards those matches.

We call this idea *pivoting* as it allows the matched corners and edges to pivot the paths estimated in adjacent epipolar lines into alignment. The matched corners and edges are referred to as *pivots* (for them the cost c_p is lower).

Consider, for example, figure 7a,b. Fig 7a shows a matching path computed for a pair of corresponding epipolar lines by standard dynamic programming techniques and also some pivots in the matching space. In fig. 7b our pivoting strategy is employed: additional weight (lower matching cost c_p) accrues at the pivot locations and these, in turn, attract the path towards them. However, the path is not forced to go through the pivots. This still leaves the chance for recovery from bad edge or corner matching if the rest of the pixels on the epipolar line indicate that a different matching is more probable.

4.1 Reducing the disparity search range

In order to speed up the path planning algorithm several heuristics could be used such as the A* algorithm. One heuristic commonly employed is to reduce the range of disparities that the graph is calculated on. This reduces the number of evaluations of the matching cost function ($c(l_i, r_j)$) necessary, which is the most expensive part of the algorithm. As a heuristic the search range on disparities is reduced to ±30 disparities of the nearest matched point. This produces a graph such as that shown in figure 7c (*cf.* fig. 2b).

However just as nearby corners can be used to reduce the search for matching edges so matched edges and corners can be used to reduce the number of disparities that need to be searched in the dense stereo algorithm. This has the effect of guiding the search path by changing its shape with a considerable increase in speed and accuracy. This will be shown in the example fig. 13.

5 Results

Within this section results are presented that demonstrate the palpable improvement yielded by the pivoting approach, both in the quality and number of dense matches produced. The results are illustrated on the pentagon stereo pair and on a stereo pair provided by Tsukuba University with known ground truth.

Figure 8 shows the result of our implementation of the standard Cox *et al* dynamic programming method on the pentagon pair; with the same parameters used as their paper [5]. Figure 8a gives the disparity map. The lighter the pixel the larger the disparity. The image has been histogram equalized but the disparity range is about 20 pixels. Note that there is little consistency between the epipolar lines, the white "streaks" correspond to particularly bad matching, such that the whole line is outlying relative to its neighbours. Figure 8b shows (in black) unmatched pixels which are liberally sprinkled across the image. Note that in the original paper of Cox *et al* the unmatched pixels are not shown separately.

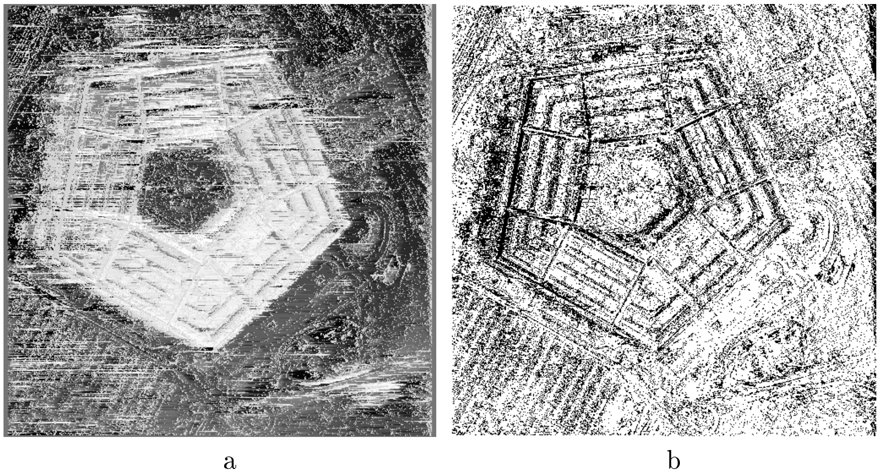

Figure 8: *Standard Cox et al method (a) Disparity Map, very streaky (b) Occlusion Map.*

Figure 9 gives the results of the Cox algorithm shown in figure 8, but this time incorporating the effects of our pivoting strategy. Here the pivots are the matched Harris corners shown in figure 5d and matched Canny edges shown in figures 5e and f. Note that the disparity map is far smoother than in the previous case with far fewer errant epipolar lines. The depth discontinuities are much sharper than in the previous example. Furthermore the occlusions around the depth discontinuities are better detected (fig. 9b).

Next, the effect of increasing the correlation window is observed. It is well known that greater correlation window sizes can increase the accuracy of the disparity, but if the window is made too large the map again degrades due to over smoothing. We found that a 5×5 window produced good results across

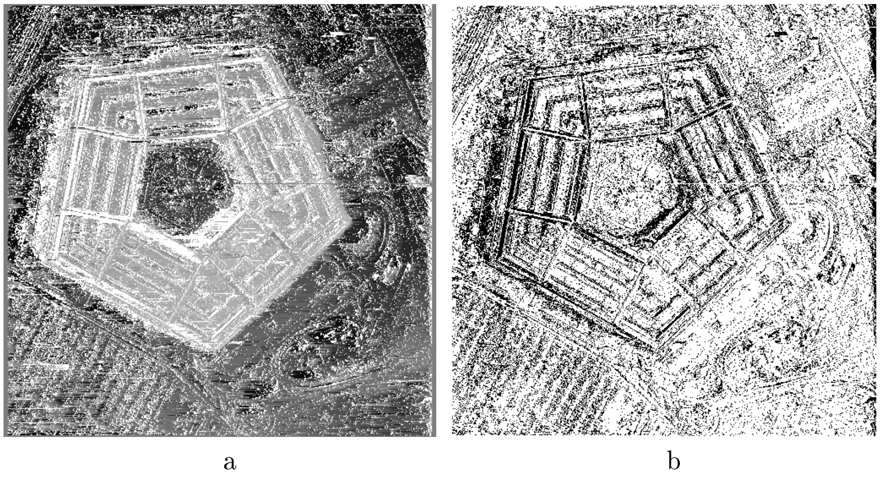

Figure 9: *Cox method with pivoting (a) Disparity Map. (b) Occlusion Map.*

the whole image as seen in fig. 10, which are an improvement over the first example (shown in figure 8). However note the “streaking” effect in the depth map (marked with white boxes in fig. 10a) where whole segments of the epipolar lines in the upper middle part have incorrectly estimated disparities (they are too large). No pivoting has been employed in this example.

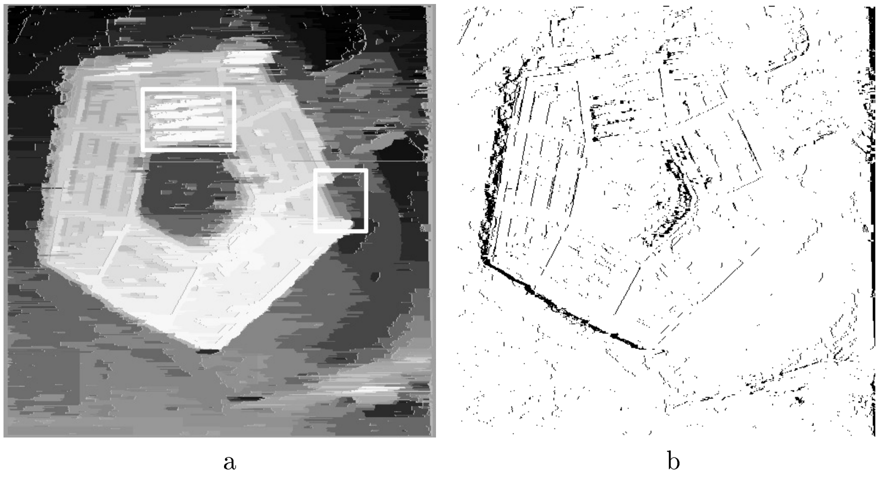

Figure 10: *Cox method using* 5×5 *window for normalized cross-correlation (no pivoting) (a) Disparity Map with some errors marked. (b) Occlusion Map.*

Next, the combination of larger correlation windows and pivoting is examined, the disparity maps and unmatched pixels for which are shown in Figure 11. Once again the results are improved, with the streaking removed. The occluding edges, especially the ones internal to the pentagon, are better picked out. Another key improvement of our method is that it dramatically increases the number of correctly matched pixels.

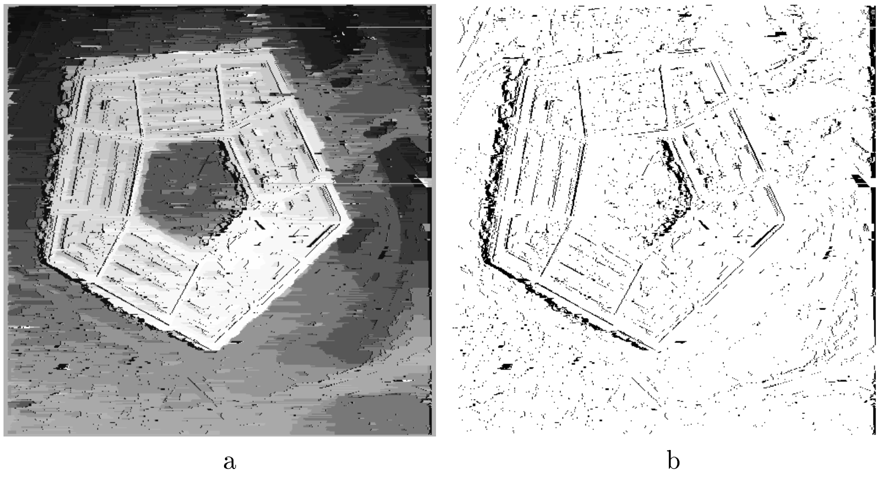

Figure 11: *Cox method with pivoting using* 5×5 *window for normalized cross-correlation. (a) Disparity Map. (b) Occlusion Map.*

The results for the pentagon pair could be further improved by incorporating other prior information on shape such as the detected planes in the image. This is an easy constraint to incorporate in the pivoting paradigm.

The next example will illustrate the effect of pivoting on a different stereo pair. Figure 12a,b shows a stereo pair taken at the University of Tsukuba. The scene is a simple composition of a slanted rectangular textured background plane in front of a textured background plane. Note that the apparent motion is horizontal. The computed corner and edges matches are shown in fig. 12d-f. The disparity ground truth has been constructed by hand (fig. 12c), but notice that the ground truth does not take occlusions into account. However, comparison with the results of our dense matching algorithm (resulting disparity map in fig. 12i) shows a close agreement. The computed occluded pixels are shown, for the left and right images, in fig. 12g and fig. 12h respectively.

A manually selected corresponding pair of epipolar lines is shown on fig. 13a,b. This pair is now used to illustrate the effect of pivoting in detail. Fig. 13c shows the matrix of matching costs, the x axis is the left epipolar line and the y axis is the right epipolar line. The x and y axes give the intensity values along the left and right epipolar lines. Each point (x, y) shows the cost of matching the pixel in the x position on the left epipolar line with the one in the y position on the right epipolar line. Rather than form the whole graph, one heuristic to speed the algorithm is to set a maximum disparity (here 90 pixels) for each pixel, this produces a cost matrix in the form of a diagonal band. Here the matching cost is the difference in pixel values. Notice the checkerboard effect caused by repeated structures making the matching quite difficult. In fig. 13d

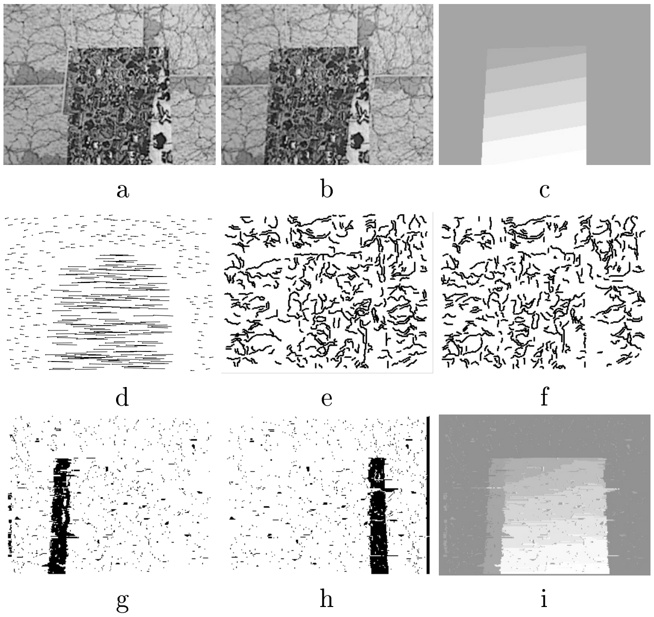

Figure 12: **(a)(b)** *The original Tsukuba stereo pair; left and right images.* **(c)** *The hand labeled disparity "ground truth".* **(d)** *Computed corner matches,* **(e) (f)** *Edge matches computed by the matching algorithm described in sect. 3.1. Note that the boundaries of the foreground slanted rectangle are not detected by the Canny edge detector.* **(g)(h)** *The computed occluded pixels (in black) for the two images.* **(f)** *The disparities computed by our improved dense stereo algorithm using a 5x5 correlation window and pivoting.*

the same cost matrix is represented but this time the pivoting strategy has been employed. Notice that the reduced search range, guided by the corner matching, is no longer straight but it follows the movements of the matched corners. Thanks to this guidance the width of the search range can be reduced from 90 pixel to 30 pixel with a commensurate further improvement in the speed of the algorithm. Fig. 13e shows the cost matrix for normalized cross-correlation and pivoting. In fig. 13f-h the computed optimal path is shown for the same three situations: first for the standard Cox algorithm (matching costs are difference of pixel intensities), second with the pivoted version of the Cox algorithm and third with normalized cross-correlation (5x5 correlation window) and pivoting. Notice that the estimated path is quite wiggly for the vanilla Cox (fig. 13f), the actual disparities should be constant across the background and foreground thus the path should be straight with gradient one, without these wiggles. Pivoting (fig. 13g) improves the straightness of the line, however there are still some wiggles. The best result is obtained by both pivoting and increasing the correlation windows to 5×5 (fig. 13h).

Finally in fig. 14 we show an application of our dense stereo reconstruction technique. From a pair of stereo images of an everyday scene (a kitchen scene, see fig. 14a,b) we generate a three-dimensional animation. The matched edges of the objects (coffee machine, bread stick, fruit bowl...) in the scene have been used for pivoting. Only two frames of the generated three-dimensional sequence (20 frames) are shown in fig. 14c,d.

Finally we have uploaded two AVI animations for the reviewers of the final result of our reconstruction algorithm. The animations depict the 3D model created from stereo images of two indoor scenes. Note the sharpness of the edges at the depth discontinuities.

6 Conclusion

Within this paper a new integrated methodology has been put forward. Rather than considering the corner matching, curve matching and dense stereo matching parts of the SFM process in isolation, we propose that they should be strongly entwined. We consider the output of each stage as a prior for the next stage, i.e. the corner matching helps the edge matching, the corner and edge matching help the dense stereo. To do this the method of pivoting is introduced which involves modifying the cost function in the dynamic programming method of estimating dense stereo. The new pivoting method helps to enforce constraints between epipolar lines and helps to reduce ambiguity within an epipolar line for disparity estimation. Furthermore it is shown how the corner matching can speed up the edge matching and the corner and edge matching can speed up the dense matching. This increase in speed can be by as much as a factor four. These improvements have been demonstrated on standard test sequences. Finally, we have uploaded results of a simple application, based on our novel pivoted dense stereo algorithm, for generating 3D animations from a single stereo pair.

References

[1] H.H. Baker and T.O. Binford. Depth from edge and intensity based stereo. In *Proc. Int. Joint Conf. Artificial Intelligence*, pages 631–636, 1981.

[2] P. Beardsley, P. H. S. Torr, and A. Zisserman. 3D model acquisition from extended image sequences. In B. Buxton and Cipolla R., editors, *Proc. 4th European Conference on Computer Vision, LNCS 1065, Cambridge*, pages 683–695. Springer–Verlag, 1996.

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

[4] J. Canny. A computational approach to edge detection. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 8(6):679–698, 1986.

[5] I.J. Cox, S.L. Hingorani, and S.B. Rao. A maximum likelihood stereo algorithm. *Computer vision and image understanding*, 63(3):542–567, 1996.

[6] G Forney. The viterbi algorithm. *Proc IEEE*, 61(3), 1973.

[7] D Geiger, B. Ladendorf, and A. Yuille. Occlusions and binocular stereo. *IJCV*, 14:211–226, 1985.

[8] C. Harris and M. Stephens. A combined corner and edge detector. In *4th Alvey Vision Conference*, pages 147–151, 1988.

[9] R. L. Henderson, W. Miller, and C. Grosch. Stereo reconstruction of man made targets. In *SPIE Vol 1*, pages 240–248, 1979.

[10] H. Ishikawa and D. Geiger. Occlusions, discontinuities, and epipolar lines in stereo. In H. Burkharddt and B. Neumann, editors, *ECCV98 Vol 1*, pages 232–248. Springer, 1998.

[11] R. Koch, M. Pollefreys, and L. Van Gool. Multi viewpoint stereo from uncalibrated video sequences. In *Proc. 5th European Conf. Computer Vision*, volume 1, pages 55–71, 1998.

[12] D. Marr and T. Poggio. Cooperative computation of stereo disparity. *Science*, 194:283–287, 1976.

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

[14] S. Roy and I. Cox. A maximum-flow formulation of the n-camera stereo correspondence problem. In U Desai, editor, *ICCV6*, pages 492–499. Narosa Publishing House, 1998.

[15] C. Schmid and A. Zisserman. Automatic line matching across views. In *Proc. Computer Vision and Pattern Recognition*, pages 666–671, 1997.

[16] C. Schmid and A. Zisserman. The geometry and matching of curves in multiple views. In H. Burkharddt and B. Neumann, editors, *ECCV98 Vol 1*, pages 394–409. Springer, 1998.

[17] P. H. S. Torr and D. W. Murray. Outlier detection and motion segmentation. In P. S. Schenker, editor, *Sensor Fusion VI*, pages 432–443. SPIE volume 2059, 1993. Boston.

[18] P. H. S. Torr and D. W. Murray. The development and comparison of robust methods for estimating the fundamental matrix. *Int Journal of Computer Vision*, 24(3):271–300, 1997.

[19] Z. Zhang, R. Deriche, O. Faugeras, and Q. T. Luong. A robust technique for matching two uncalibrated images through the recovery of the unknown epipolar geometry. *AI Journal*, vol.78:87–119, 1994.

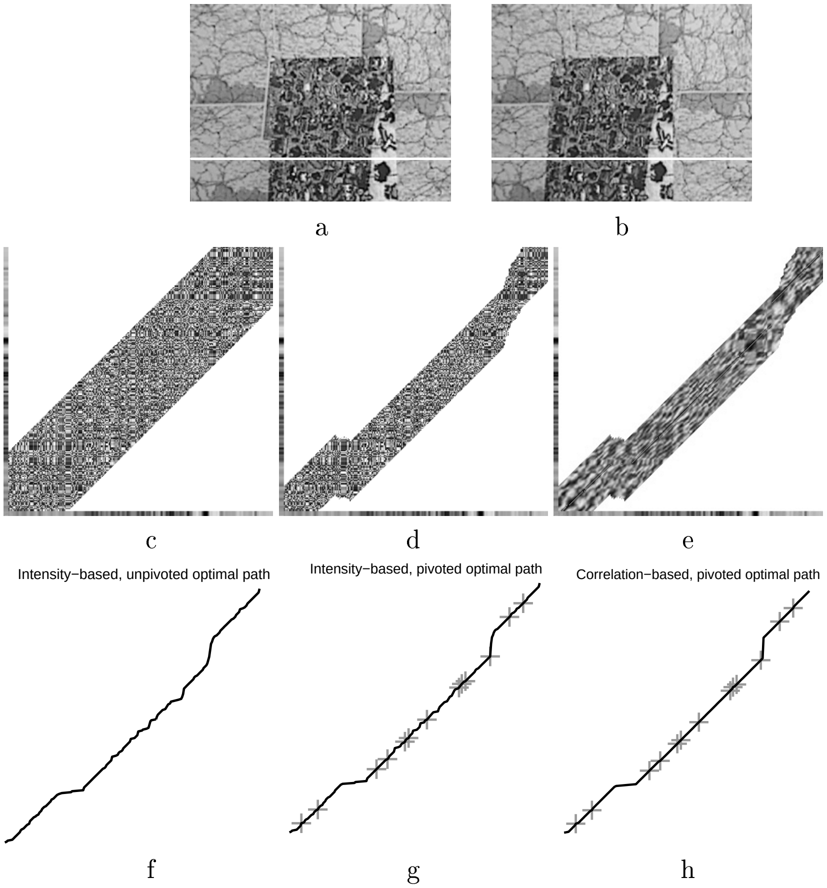

Figure 13: **(a)(b)** *Left and right images of the Tsukuba image pair with a selected pair of corresponding epipolar lines superimposed (in white).* **(c)(d)(e)** *Cost matrices for matching (lighter shades mean higher matching cost), see text.* **(f)(g)(h)** *Change in path due to pivoting and increase of correlation window; the dark curve is the estimated optimal path and the gray crosses are the pivots, see text.*

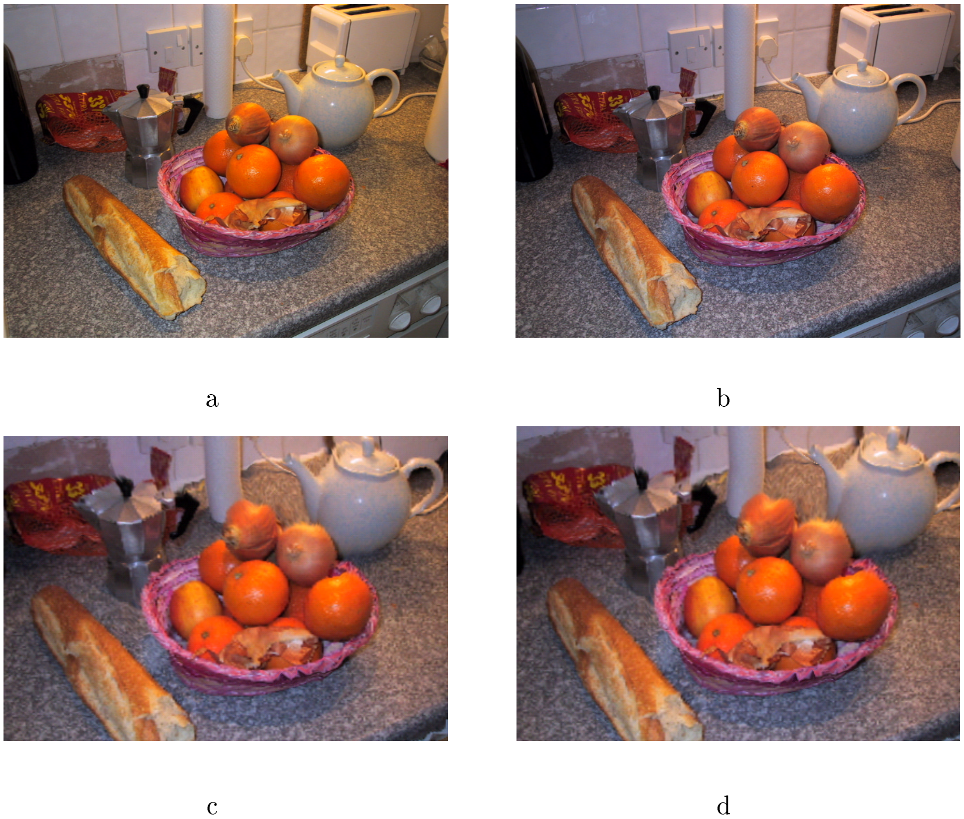

Figure 14: **(a),(b)** *The left and right images of the kitchen stereo pair.* **(c),(d)** *Two frames of the generated three-dimensional sequence (the scene is viewed slightly from above).*