5. The disparity space image path

Because disparity can be modelled as one dimensional it is possible for a single scan-line to visualise the match errors for the various possible disparities in the so called disparity space image which is described in Section 5.1. A description of what can be seen in the disparity space image is given in Section 5.2. One of the things which can be found in the disparity space image is the path of true disparity. Some characteristics of this path are discussed in Section 5.3.

The relationship of the disparity space image with disparity estimation is dealt with in Section 5.4. The technique by which a disparity path can be found in a disparity space image using dynamic programming is described in Section 5.5.

A disparity path finding technique based upon the genetic algorithm is presented in Section 5.6. Experimental results are presented in Section 5.7.

5.1 The disparity space image

The disparity space image is defined for a corresponding pair of lines from a stereoscopic picture pair. The picture has dimensions of H_{pix} rows by W_{pix} columns. The intensities for the W_{pix} pixels of the ith line of the left image are given by:

$$s_L(x,i) \quad \text{where } 0 \le x \le W_{pix} \quad \text{and} \quad 0 \le i \le H_{pix}. \tag{5.1}$$

Analogously the intensities for the W_{pix} pixels of the ith line of the right image are given by:

$$s_R(x,i) \quad \text{where } 0 \le x \le W_{pix} \quad \text{and} \quad 0 \le i \le H_{pix}. \tag{5.2}$$

The basic form of the DSI, the left DSI, is defined by:

$$DSI_i^L(x,d) = \begin{cases} |s_L(x,i) - s_R(x+d,i)| & 0 \le (x+d) < W_{pix} \\ \text{not defined} & \text{elsewhere} \end{cases}. \tag{5.3}$$

The x values of this DSI can take on the same values as the x values in the original pictures while the d values are usually bounded by the minimum and maximum disparities under consideration.

The right DSI is defined by:

$$DSI_i^R(x,d) = \begin{cases} |s_R(x,i) - s_L(x-d,i)| & 0 \le (x-d) < W_{pix} \\ \text{not defined} & \text{elsewhere} \end{cases}. \tag{5.4}$$

Comparison of the formulae for DSI_i^L and DSI_i^R reveals that DSI_i^L is contains the same information as DSI_i^R except that it is skewed along the x=-d axis:

$$DSI_i^L(x,d) = |s_L(x,i) - s_R(x+d,i)| = |s_R(x+d,i) - s_L(x,i)| \\ = DSI_i^R(x+d,d) \tag{5.5}$$

The geometrical mapping of DSI_i^L onto DSI_i^R is illustrated in Figure 5.1. The shaded areas are illegal in DSI_i^L and DSI_i^R respectively, areas where *x+d* or *x-d* respectively is out of range. The actual DSI information in the two DSIs is contained in the clear, non-shaded, areas. Note that those areas which are illegal in DSI_i^L fall outside of the $0 \le x \le N-1$ range in DSI_i^R and vice versa.

As DSI_i^R contains no additional information to that contained in DSI_i^L only DSI_i^L will be used henceforth unless the use of DSI_i^R is specifically required.

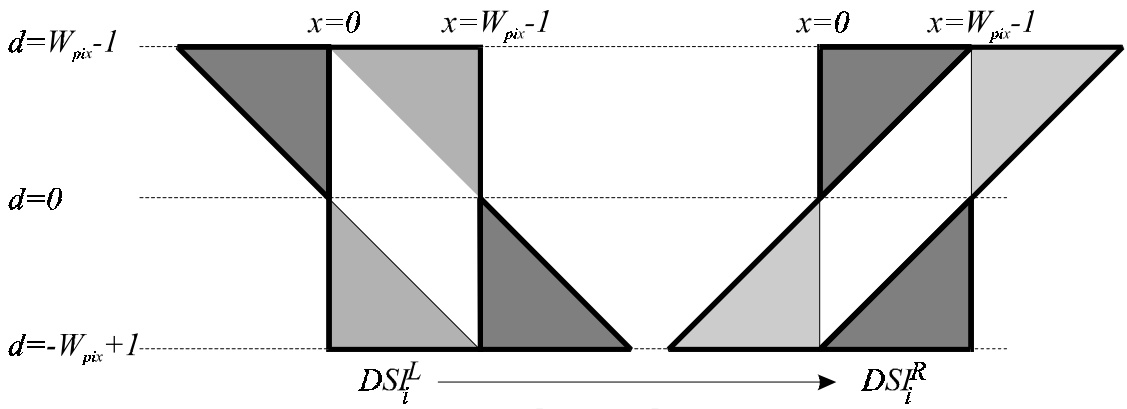

Figure 5.1: Geometric mapping of DSI_i^L to DSI_i^R.

Because practical digital imagery is affected by noise it is often necessary to apply prefiltering: calculating differences averaged over a window instead of for single pixels. For a filter window to be symmetrically centred around a pixel position its height, N_y, and width, N_x, must have odd integer values greater than one. The integer values of n_x and n_y are chosen such that:

$$N_x = 2n_x + 1, \quad n_x \ge 1 \tag{5.6}$$

and

$$N_y = 2n_y + 1, \quad n_y \ge 1. \tag{5.7}$$

Now the expressions for the filtered disparity space image become:

$$DSI_i^L(x,d,n_x,n_y) = \frac{\sum_{x_c=x-n_x}^{x+n_x} \sum_{i_c=i-n_y}^{i+n_y} \left(s_L(x_c,i_c) - s_R(x_c+d,i_c)\right)}{N_x \cdot N_y} \tag{5.8}$$

and

$$DSI_i^R(x,d,n_x,n_y) = \frac{\sum_{x_c=x-n_x}^{x+n_x} \sum_{i_c=i-n_y}^{i+n_y} \left(s_R(x_c,i_c) - s_L(x_c-d,i_c)\right)}{N_x \cdot N_y} \tag{5.9}$$

Choosing $n_x = n_y = 0$ (so that $N_x = N_y = 1$) results in the original DSI formulae.

5.2 Interpretation of the disparity space image

Ideally where d corresponds to the true disparity for pixel x of line i the following equality should hold:

$$DSI_i^L(x,d) = 0. \tag{5.10}$$

Finding the true disparity d for each x co-ordinate is not as easy as finding the d for which $DSI_i^L(x,d) = 0$ because of several reasons:

(a) There are matches $DSI_i^L(x,d) = 0$ for values of d which do not correspond to the true disparity.

(b) Because of occlusion (discussed later) there is sometimes no true match for the pixel at a particular x co-ordinate.

(c) In practice stereoscopic images are not noiseless which means that it is not possible to rely upon DSI_i^L equalling zero for a true match or even on a DSI_i^L value of zero implying any match at all.

A sample disparity space image is given in Figure 5.2 which represents DSI_{228}^L for the "aqua" test scene with disparity cropped to +/- 128 pixels. Inspection of this

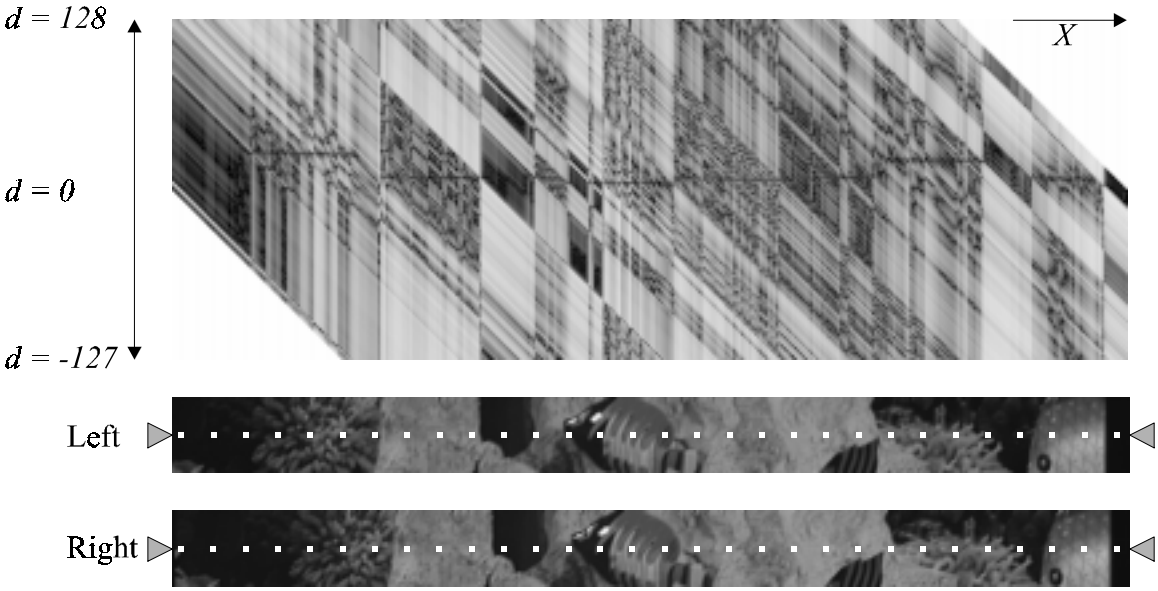

Figure 5.2: DSI_{228}^L for the first frame of the aqua sequence.

figure reveals a number of properties typical for disparity space image. These properties will be reviewed in the following sections.

For the purposes of visual inspection it helps to logarithmically scale the grey scale values of the pixels in the disparity space image. In the disparity space images shown on these pages the differences were first mapped to the [0,255] interval, logarithmically scaled and re-normalised to the [0,255] interval using the logarithmic display scaling function:

$$lds(x) = 255 \cdot \frac{\log(x+1)}{\log(256)} \ , \quad x \in [0,255]. \tag{5.11}$$

Interpreting the effects which can occur in the disparity space image is important because the disparity space image is in fact the "space" in which disparity estimators work.

5.2.1 The disparity path

True matches of pixels to their disparity shifted counterparts result in (near) zero disparity space image values. These true matches are visible as horizontal dark lines at appropriate disparity levels in the DSI. There are discontinuities in this horizontal pattern. Close inspection of Figure 5.2 (or another DSI_i^L) reveals that the nature of this discontinuity is different when going from a low disparity to a high one when compared to the discontinuity when going from a high disparity to a low one. This will be discussed in more detail in Section 5.3.

5.2.2 Diagonal clipping

In finite digital images pixels near the left edge (low x) of the left image cannot have DSI_i^L values for low (negative) disparities as this would imply calculating differences with pixels with negative x co-ordinates in the right image. These pixels are simply not there to calculate a difference with. Similarly pixels near the right edge (high x) of the left image cannot have DSI_i^L values for high (positive) disparities because this would mean having to calculate differences with pixels to the right of W_{pix}-1 in the right image. Hence the $0 \leq x+d < W_{pix}$ condition in (5.3). A similar argument holds for low disparities at the left edge of the right image and high disparities at the right edge of the right image in DSI_i^R.

5.2.3 Vertical edges

The vertical edges in the DSI_i^L correspond to edges in scan-line i of the left image. In DSI_i^L an edge in the left scan-line would have constant x, hence the fact that vertical line effects are introduced into DSI_i^L. The values along a vertical line correspond to the matches which are found at the various disparity values for the pixel at x.

5.2.4 Diagonal edges

The diagonal edges in the DSI_i^L can be equated to edges in scan-line i of the right image. A constant x co-ordinate in the right scan-line translates to a constant $x+d$ on the left scan-line. This results in the top-left to bottom-right sloping diagonals in the DSI_i^L.

5.2.5 Parallelogram patterns

The parallelograms which are visible in the DSI_i^L are bounded by a pair of vertical lines and a pair of diagonal lines as described in Sections 5.2.3 and 5.2.4. The vertical lines which bound the parallelograms on the left and right correspond to edges on the scan line from the left picture. Assuming that these edges belong to "objects" each such object in the left image produces a pair of vertical edges in the DSI_i^L. These edges bound one or more parallelograms. Dark parallelograms (low DSI_i^L values) indicate matches of pixels from the right scan-line to the left scan-line.

Uniformly dark parallelograms correspond to homogenous objects, objects for which the I value is almost constant so that a pixel from the object in the right-scan-line matches well to a number of pixels in that same object in the left scan-line. For these parallelograms the height from top vertex to bottom vertex is equal to twice the width from left vertex to right vertex. This proportionality of vertex lengths does not hold for matches of pixels from an object to pixels of similar I value in other objects which also produce parallelogram patterns.

Textured parallelogram patterns correspond to objects of non homogenous intensity where pixel match values van vary within an object.

5.2.6 Diagonally crossing stripe patterns

Where pixels from an object can find a match only at true disparity a light (high DSI_i^L values) diagonal cross pattern will be formed. At the intersection of the vertical and diagonal there will be a dark spot which is part of horizontal dark disparity path line except when the object in question is occluded in the alternate picture and there is absolutely no match for it. An example of such a diagonal cross is the instance caused by the light reflection on the middle fish in Figure 5.2 where the high I values are unique for scan-line 228 so that no matches other than the true disparity match may be found.

5.2.7 Disparity space image degeneration

When the left and right view images come from a camera which has a larger base line and/or convergence angle degeneration of the disparity space images occurs as illustrated in Figure 5.3.

The left and right view images in the figure were shot with a camera with a relatively large base line. It will be readily apparent that the differences between left and right view images now exceed pure horizontal shift. The two images contain different aspect views of the same face.

The corresponding disparity space images lack the clearly discernible dark disparity path described in Section 5.2.1. Also the parallelogram patterns, which are bounded by straight lines in good disparity space images, start to show degenerate curved edges.

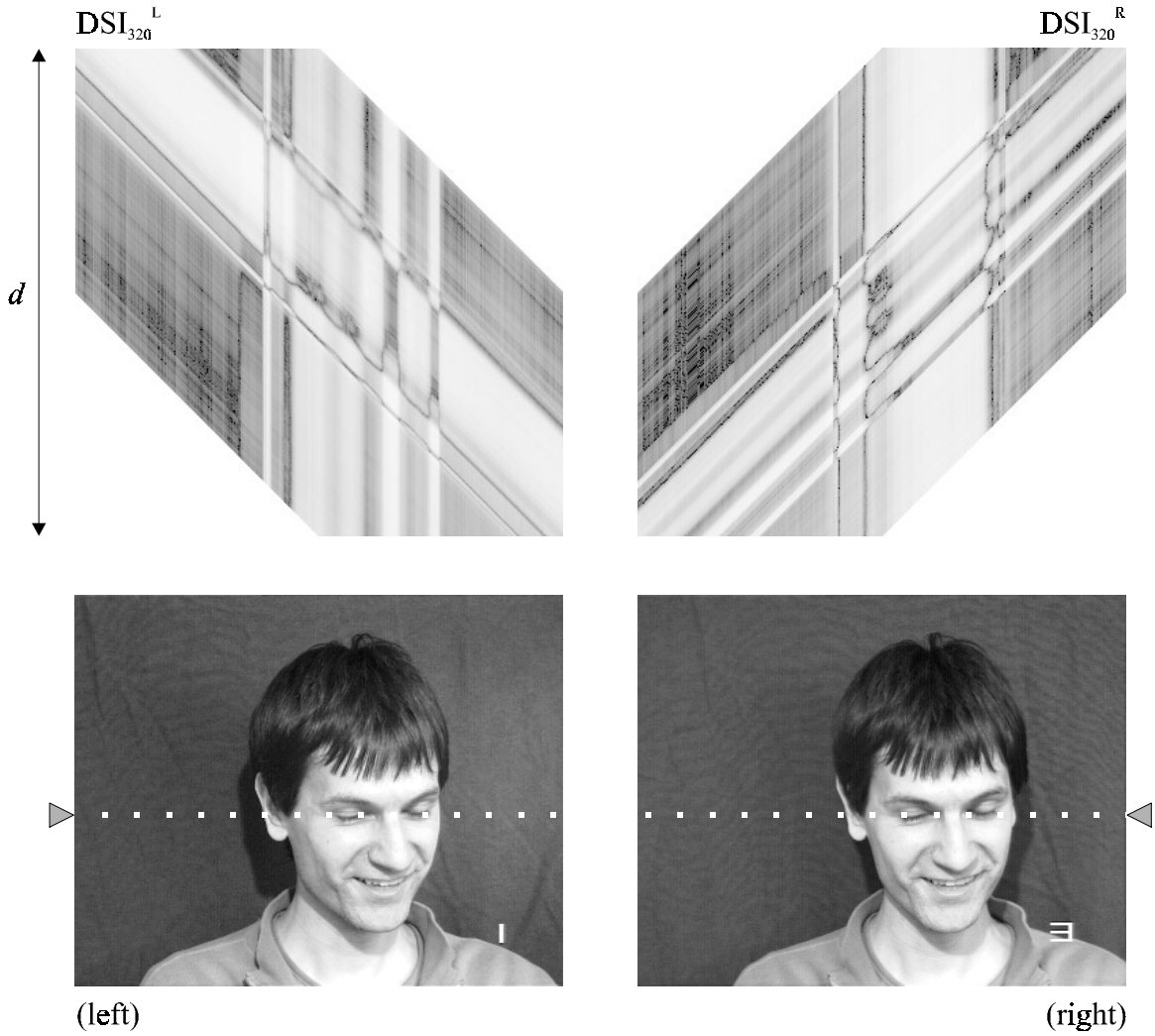

Figure 5.3: Example of degenerate disparity space images.

5.3 Rules for the disparity path

In Figure 5.4 a scene is depicted which is invariant in the *Y* component sense which means that we can think of either as flat or as mirror symmetric with regard to any plane which is parallel to the *X*-*Z* axis. Either way all scan-lines of this scene will be identical and it the disparity path along one of these scan-lines which is discussed here.

The scene contains three objects *A*, *B*, and *C*. *A* forms the background of the scene while *C* is nearest to the observer and *B* lies at a depth somewhere in between the two. In analogy to the thought behind DSI_i^L the left eye will be considered as scanning the scene from left (low *x*) to right (high *x*) while it a disparity match is sought in the material observed by the right eye. The corresponding plot of *x* versus *d* is referred to as the left disparity path for this scan line.

Initially, at the extreme left of the scene, the points on *A* observed by the left eye can be matched by the same points as observed by the right eye. As *A* is the furthest object in the scene this disparity match is made at the highest disparity present in the scene, d_A. At a point x_1 it is no longer possible to find a match for *A* because the right eye's view of it is occluded by *B*. As the scanning goes on the left eye encounters the left edge of *B* at point x_2. As the right eye can also see this point the disparity continues after x_2 at a level of d_B. Between x_1 and x_2 (the piece which corresponded to the part of *A* which is occluded to the right eye) the disparity is indeterminate as indicated by the dotted line in Figure 5.4.

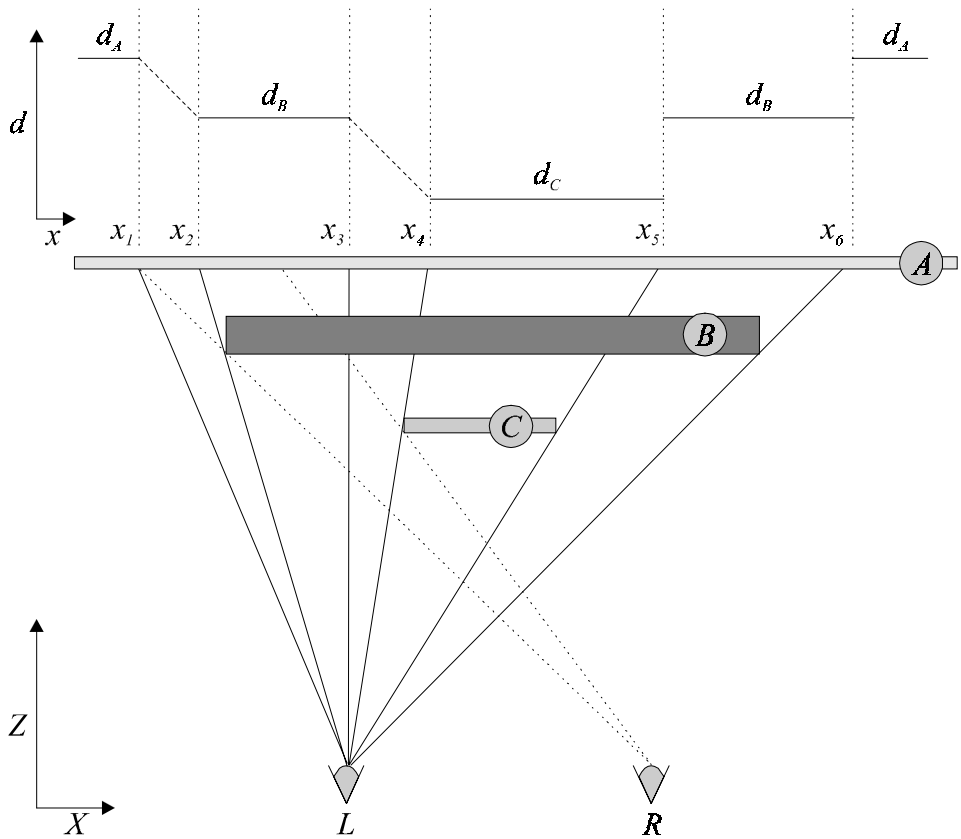

Figure 5.4: Example scene with corresponding left disparity path.

The same effect of disparity transition occurs where the left edge of *C* occludes part of *B* to the right eye this time with the occlusion between x_3 and x_4.

Just after having passed x_4 the left eye sees *C* which is also visible along its complete length to the right eye so that it may be matched at a disparity of d_C upto x_5. After x_5 *B* again becomes visible to the left eye. The first point on *B* that

becomes visible to the left eye is also visible to the right eye so that a match at a disparity of d_B can immediately be made from x_5 onwards. Similarly A becomes visible to both eyes at x_6 so that a disparity match at d_A can be made from x_6 onwards.

It will be clear that there are two types of disparity discontinuity in this left disparity path. The transition from high to low disparity means that the left eye encounters the left edge of (relative) foreground object while scanning along the line. A part of the relative background has been occluded to the right eye. The size of this piece of background which is occluded is directly proportional to the disparity difference.

Because this occlusion had been caused by the left edge of an occluding object this kind of occlusion is referred to as a left occlusion.

The transition from low to eye disparity means that the left eye encounters a background object after a foreground object. Nothing will have been occluded from the right eye and thus the disparity transition takes the form of a clean jump.

The situation with the right disparity path where the right eye scans from left to right and a match is made with what is perceived by the left eye in analogy of DSI_i^R is sketched in Figure 5.5.

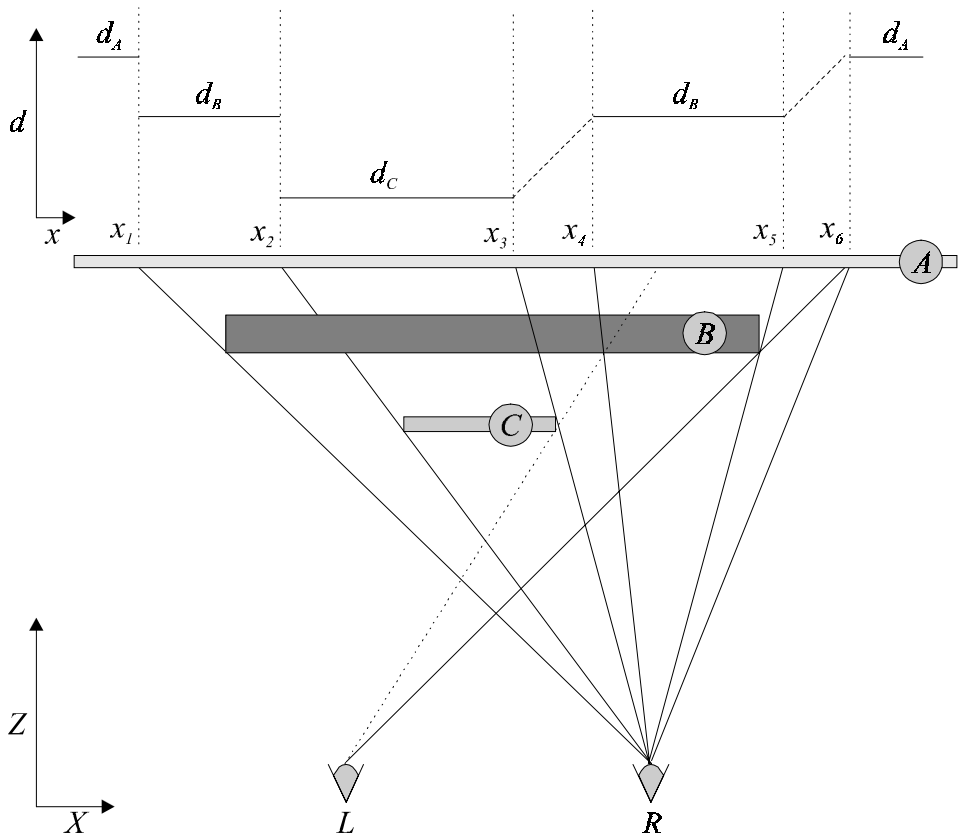

Figure 5.5: Example scene with corresponding right disparity path.

The main difference is that the high to low disparity transition is now a clean jump while the low to high disparity transition corresponds to an occlusion region. As this

effect is now caused by the right edge of an occluding object it is referred to as a right occlusion.

In this work the sign of disparity is chosen the same in the left and right disparity space image and in the left and right disparity path. In high disparity implies “far” and low disparity means “near”. Sometimes disparity is also taken to have opposite signs in left and right disparity space image and disparity path. This has the advantage of the rules by which disparity jumps are made are identical in both left and right disparity paths. It does however have the disadvantage that disparity now gets assigned different physical meaning in left and right data.

It should be noted that there is one pathological situation in which the disparity path rules given here won’t hold. That is when the width of the occluding object (in pixels) is smaller that amount by which its disparity is lower than that of the object it occludes. This is illustrated for a left disparity path in Figure 5.6.

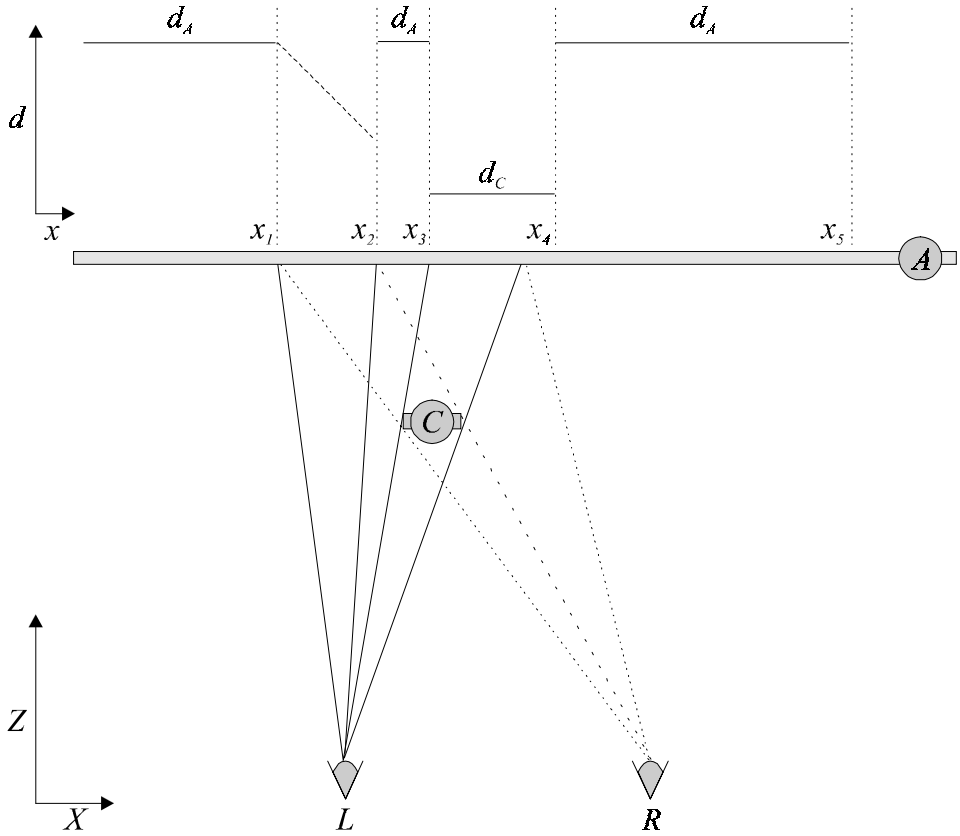

Figure 5.6: Example of ordering constraint violation.

As the left eye scans from left to right can no longer match parts of A from x_1 onwards and a region of left occlusion sets in. However C is very small so that the right eye can actually see round it and it is possible to find a disparity match at d_A for the parts of A between x_2 and x_3. From x_3 onwards things again proceed as for a normal left disparity path.

This problem is called ordering constraint violation because it effectively means that there is a part of A which the left eye perceives as being to the left of C while the

right eye perceives that part of *A* as lying to the right of *C*. The relative order of objects changes from the perception of the left eye to the perception of the right eye.

If ordering constraint occlusions can occur then the rules derived previously for the disparity path are (partially void). Whether ordering constraint violations occur and form a practical problem depends on the material being used, the scene in question. Particularly scenes shot at close range with a large base line will tend to present problems. Most "TV" type material shot at a longer distance with a larger base line will not present any problems.

5.4 Disparity space images and disparity estimation

Disparity estimators which estimate horizontal disparity examine an *x-d* match error space which, in essence, *is* the disparity space image. Therefore the disparity space image can bring insights into the working of existing disparity estimators and it can be of help in the designing of new disparity estimation techniques.

Blockmatchers do not estimate *d* for all possible *x* but only for a subset of all possible *x* which corresponds to (the centres) of the blocks used. Also the blocks are not pixel sized as can be reflected in the n_x and n_y parameters of the disparity space image formulae.

A full search blockmatcher used as disparity estimator searches the same range of *d* for every *x* for which it has to estimate a vector. The *d* which corresponds to the lowest error is selected and output as the estimated *d*. This corresponds to looking at a column in the disparity space image and selecting the "blackest" *d*. Often this *d* will be quite correct but in the presence of non-negligible noise it can also be wrong as can be seen on inspection of full search blockmatcher results.

The recursive blockmatcher takes the *d* estimated for the previous *x* examined and searches a limited disparity range around this value. If the previous *d* was on the correct disparity path then this will reduce the chances of a *d* being chosen which corresponds to a match error minimum which is not on the disparity path. There will be problems with occlusion jumps, especially when these are too large in magnitude. Large parts of the disparity space image will never even be examined by the recursive blockmatcher. It may well be that parts of the true disparity path are to be found in these areas.

The effect of hierarchical blockmatchers can be likened to a full searcher operating on a disparity space image calculated with large n_x and n_y. As inspection of **{???}** shows disparity space images calculated with large n_x and n_y have a disparity path which can be safely found with a full search blockmatcher. In later hierarchical levels refinements are made to this initial path using disparity space images calculated with smaller n_x and n_y. As with the recursive blockmatcher the disparity

space image on the final hierarchical level will have been examined in a limited area.

The genetic vector field estimator of the previous chapter is less simple to describe. It operates on vector fields and not distinct vectors. This means that a disparity path (a row of vectors) can be influenced by disparity paths above and below. Just looking in the horizontal sense the hybrid vector field mutation operator takes a single point on the disparity path, tries to update the disparity and then propagate it to the left and right. The vector field crossover operator chooses a spot along the path. It then determines the (constant disparity) part of the path to which this spot belongs in the two parent paths and exchanges the two path sub parts.

The two things which the genetic vector field estimator does not know about are occlusion and the rules which the ordering constraint imposes on the disparity path.

5.5 Disparity path finding with dynamic programming

Finding a good disparity path which corresponds to good matches and which obeys the strictures which the ordering constraint imposes on a disparity path is a problem which is closely related to classic combinatorial problems from the field of operations research like the travelling salesman problem and the stagecoach problem where a minimum cost "path" of some kind is the objective.

A standing technique for finding disparity paths in disparity space images uses dynamic programming [Intill93]. The term dynamic programming covers a wide range of techniques [Hill89]. The basic idea is that dynamic programming starts by examining a small part of the problem and finding an optimal solution for this part of the problem. It then extends the problem a little bit and extends the previously found optimal sub-problem solution to a solution which covers the current, slightly extended, view of the problem. This extension/solution cycle is repeated until the problem is solved in its entirety. Particularly for larger problems the computational savings when compared to full enumeration can be enormous.

The dynamic programming technique for disparity path finding will be explained here by illustrating it in an example. Left and right scan-line data for this example are given in Table 5.1:

Table 5.1: Example scan-line data.

x	0	1	2	3	4	5	6	7	8	9	10	11
I_L	5	10	15	20	25	100	100	100	75	80	85	90
I_R	0	5	10	100	100	100	44	69	70	75	80	85

The actual disparity in the example scene corresponds to a background at a disparity level of +1 and a foreground object three pixels wide at a disparity level of -2. This foreground object has an intensity value of 100 and extends from $x = 5$ to $x = 7$ on the left scan-line and because it has a disparity of -2 it extends from

$x = 3$ to $x = 5$ on the right scan-line. The cells corresponding to the object location on the left and right scan-lines have been shaded in Table 5.1. Data for a left disparity space image is calculated for these two scan-lines across a disparity range of -3…3 and is listed in Table 5.2. In this table those locations in the bottom left and top right which correspond to $x+d$ going out of bounds have artificially large values inserted (in this case 999). Also those locations in the disparity space image data which correspond to the true disparity path (which is known because this is an artificially constructed example) have been shaded.

Table 5.2: Disparity space image values for example data.

d	x=0	x=1	x=2	x=3	x=4	x=5	x=6	x=7	x=8	x=9	x=10	x=11
3	95	90	85	24	44	30	25	20	10	999	999	999
2	5	90	85	80	19	31	30	25	5	5	999	999
1	0	0	85	80	75	56	31	30	0	0	0	999
0	5	5	5	80	75	0	56	31	5	5	5	5
-1	999	10	10	10	75	0	0	56	6	10	10	10
-2	999	999	15	15	15	0	0	0	31	11	15	15
-3	999	999	999	20	20	90	0	0	25	36	16	20

On the disparity path it was not possible to choose a disparity of +1 for $x = 11$ because this would imply matching the intensity value at $x = 11$ on the left scan-line to the intensity value at $x = 12$ on the right-scan line. As this data is simply not available the next best thing which can be done in this case is to match at a disparity of 0.

Now to start finding the disparity path using dynamic programming the problem is first limited to finding a disparity path from $x = 11$ onwards, out of the disparity space image. What costs are involved in this? Simply the relevant matching error values from the last column disparity space image. These errors are entered into the cost table which is given in Table 5.3. The entries of this table are referred to as $C_i^L(x,d)$. The L superscript denotes that this table is belongs with a left disparity

Table 5.3: Cost matrix for example data.

d	x=0	x=1	x=2	x=3	x=4	x=5	x=6	x=7	x=8	x=9	x=10	x=11
	999	999	999	999	999	999	999	999	999	999	999	999
3	20	20	20	20	17	14	11	11	11	1000	1000	999
2	18	18	18	18	18	15	12	9	9	9	1001	999
1	16	16	16	16	16	16	13	10	7	7	7	999
0	17	17	17	14	14	14	14	11	8	8	8	5
-1	18	18	18	15	12	12	12	12	9	9	9	10
-2	19	19	19	16	13	10	10	10	10	10	10	15
-3	20	20	20	17	14	11	11	11	11	11	11	20
	999	999	999	999	999	999	999	999	999	999	999	999

space image and disparity path while the i subscript indexes the scan-line, x denotes the horizontal co-ordinate and d stands for disparity.

For x = 11 the optimal d is found to be 0 because no match can be made at d = 1.

Next the problem is extended to finding a disparity path (out of the disparity space image) from x = 10 onwards, i.e. the costs for the x = 10 column of the cost matrix have to be determined. There are three possibilities for each cell entry. The idea is to choose the minimum of these three.

This first possibility is that the current cell at (x,d) is part of a horizontal piece of the disparity path. The costs for getting out of the disparity space image from this cell are equal to the appropriate disparity space image value for this cell plus the minimal path completion costs for the cell directly to the right. Thus the value of the first candidate cost $cc_i^{L,1}(x,d)$ is given by:

$$cc_i^{L,1}(x,d) = DSI_i^L(x,d) + C_i^L(x+1,d)\,. \tag{5.12}$$

A second possibility is one where the current cell is part of a diagonal occlusion transition in the disparity path. In this case the minimal cost for getting out would be equal to the cost of getting out from the cell directly below and to the right plus a penalty value p_d for making the diagonal jump. Using a disparity space image value would not be appropriate here because disparity space image values are match errors and if an occlusion is assumed then that logically excludes a match. The penalty value is required to stop the dynamic programming algorithm from gratuitously deciding on diagonal transitions. So the second candidate cost now becomes:

$$cc_i^{L,2}(x,d) = p_d + C_i^L(x+1,d-1)\,. \tag{5.13}$$

The third and last possibility is the one where the current cell is part of a sheer vertical disparity rise in the disparity path. In this case the cost of getting out from the current cell onwards is equal to a penalty value p_v plus the cost of finishing the path from the cell above onwards:

$$cc_i^{L,3}(x,d) = p_v + C_i^L(x,d+1)\,. \tag{5.14}$$

As long as $p_d > 0$ it is in principle possible to choose $p_v = 0$ because gratuitous vertical jumps in the path would be penalised by subsequent diagonal descents.

When the three candidate costs have been determined the cost for the current cell is simply set to the minimum of the three:

$$C_i^L(x,d) = \min\{cc_i^{L,1}, cc_i^{L,2}, cc_i^{L,3}\}\,. \tag{5.15}$$

Because the costs for a cell depend on those for the cell above the costs should be calculated from top to bottom in a column. The order in which the columns are

evaluated is from right to left in keeping with the idea of extending the problem in a step by step fashion.

The values in Table 5.3 were calculated as described above using the values in Table 5.2 and $p_d = 2$ and $p_v = 1$. When the whole of C_i^L has been filled the disparity path can be found by tracing the path of minimum cost back through C_i^L, from left to right, jump for jump. determining the d which corresponds to the minimum cost value in each column.

The three possible candidate costs for a cell in $C_i^L(x,d)$ are illustrated graphically in Figure 5.7. The figure also illustrates the three basic directions involved in the right to left disparity path tracing along a small example disparity path.

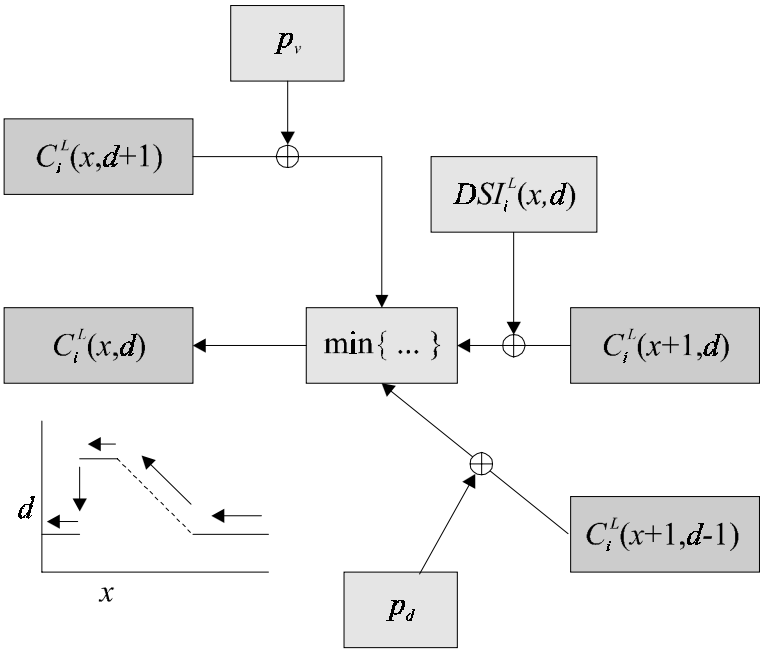

Figure 5.7: Candidate costs for $C_i^L(x,d)$.

The situation for the right disparity space image and disparity path is entirely analogous except that the vertical direction of the diagonal and discontinuous jumps is reversed:

$$cc_i^{R,1}(x,d) = DSI_i^R(x,d) + C_i^R(x+1,d), \tag{5.16}$$

$$cc_i^{R,2}(x,d) = p_d + C_i^R(x+1,d+1), \tag{5.17}$$

$$cc_i^{L,3}(x,d) = p_v + C_i^R(x,d-1), \tag{5.18}$$

and

$$C_i^R(x,d) = \min\{cc_i^{R,1}, cc_i^{R,2}, cc_i^{R,3}\} \tag{5.19}$$

as illustrated in Figure 5.8.

The order in which C_i^R is constructed is again right to left but now each column in constructed bottom (low d) to top (high d).

In fact the right disparity path can be found using exactly the same algorithm and formulae as the left disparity path if the left and right input scan-lines are exchanged and the sign of the *d* on the calculated path is reversed.

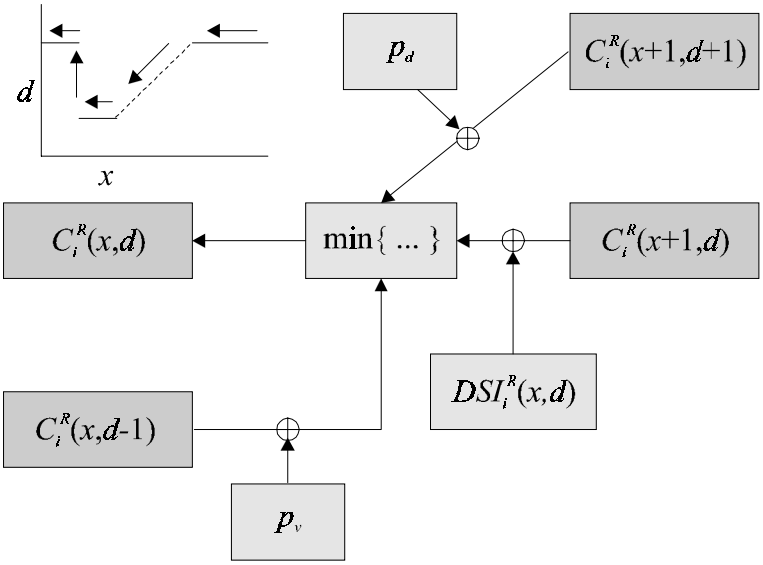

Figure 5.8: Candidate costs for $C_i^R(x,d)$.

Some disparity paths which have been calculated by applying the dynamic programming technique are shown in Figure 5.9 along with the disparity space image in which they were calculated.

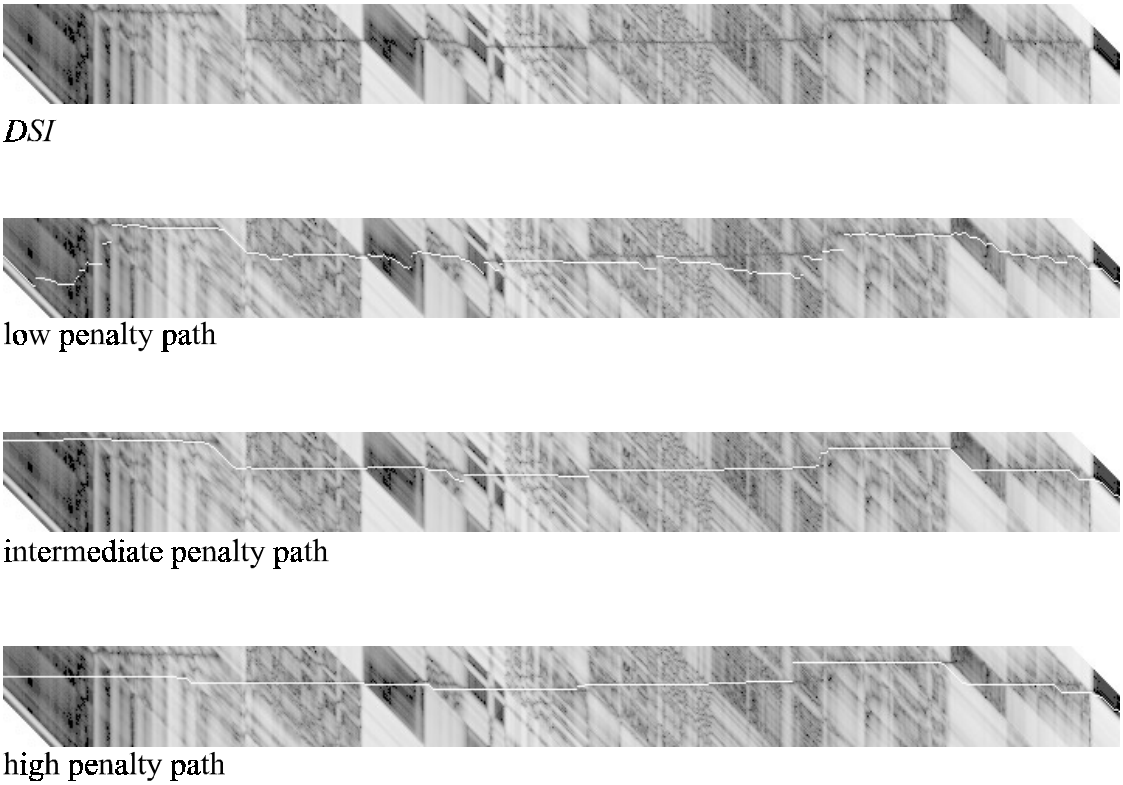

Figure 5.9: Influence of occlusion penalty on disparity path.

As can be seen from the figure the choice of p_d is a question of compromise. Choosing p_d too large brings with it the danger of larger occlusion jumps not being correctly identified because p_d multiplied by the size of the occlusion jump is a value larger than the disparity space image value penalty which will be incurred along the alternative non-occluded path.

Also an occlusion jump of one pixel only from one non-occluded part of the path to another should not really be penalised at all because it merely reflects the fact that a single object can have a disparity gradient along a scan line. The central rock in the "aqua" test-scene is an example.

On the other hand when p_d is chosen too small "wiggles" (Figure 5.10) will start to appear in the horizontal, non-occluded parts of the disparity path as a result of small occlusion jumps being made which are advantageous in terms of path cost but which are based on noise.

Also there is the danger of the completely degenerate solution appearing where a whole path is made up of occlusion jumps. In fact with a $p_d = 0$ and $p_v = 0$ this degenerate solution is guaranteed to be found because it corresponds to a path cost of zero!

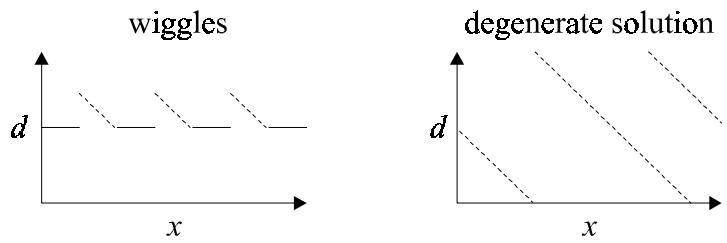

Figure 5.10: Problems with small occlusion penalty.

In order to avoid these problems a path cost function is required which is capable of distinguishing between occlusion jumps and changes in disparity because of the disparity gradient in a object. Occlusion jumps must have some kind of penalty associated with them because if they do not then the degenerate solution will be optimal. This penalty must not penalise larger occlusion jumps more than small occlusion jumps because in principle both are equally valid.

Such a cost function will need to have more global information about the path available to it then has the dynamic programming algorithm for disparity path finding which, in keeping with the Markovian nature of dynamic programming in general, calculates cost incrementally on a very local basis, independently of the part of the path which has yet to be determined.

Along with an improved cost function an optimisation algorithm will be required which is capable of finding an optimal path in terms of this cost function.

5.6 A genetic disparity path finding algorithm

The genetic algorithms described in the previous Chapter are a suitable choice of optimisation algorithm for finding a good disparity path according to a cost function. As with the genetic vector field estimation the design of a genetic algorithm for finding a path in the disparity space image again comes down to reflecting knowledge of the problem at hand onto the chromosome encoding technique, the fitness function and the reproduction operators.

The discussion of the genetic disparity path finding algorithm will be given for left disparity paths and disparity space images but subject to the appropriate modifications it is off course also applicable to right disparity paths/disparity space images like just as with the dynamic programming algorithm.

5.6.1 Path encoding

As the solution which is required is a disparity path stretching from the left of the disparity space image to the right of the disparity space image a chromosome should describe such a path.

One way of encoding a disparity path would be to store the locations of all key points (where a piece of path starts and how far it extends) in a list of nodes. As a typical disparity path has no more than 25 or so of such nodes such an encoding strategy would result in very compactly encoded chromosomes.

However, these chromosomes would also be of variable length. Variable length chromosomes introduce the problem of how to perform crossover. With variable length chromosomes it is no longer possible to choose a cross site in one parent chromosome and to use that same cross site in the second parent chromosome. In variable length chromosomes there is no genotypic correspondence between identical positions in two different lists.

Crossover with variable length chromosomes is possible but the way to do it is to perform it on a phenotypic as opposed to a genotypic basis [David89]. Supposing a node is selected for crossover in the first parent chromosome. This node corresponds to a location at (x_1,d_1). In order to find a suitable node for crossover in the second parent chromosome its list of nodes is searched for a node with (x_2,d_2) as near as possible to (x_1,d_1) according to some distance measure.

A second possibility of encoding a disparity path is to use a fixed length encoding and store a single d for each x thus resulting in a chromosome containing W_{pix} disparity values. This is conceptually less elegant and also less compact than the variable length encoding based on nodes. On the other hand genotypic crossover is now possible.

The choice between the two encoding strategies cannot really be made on functional merit because when it comes to choosing between the two there is the

fact that there is nothing which can be done using the variable length encoding which can't be done using the fixed length encoding and vice versa. Even genotypic crossover can be performed after a fashion using the variable length encoding if extra (redundant) nodes are inserted when required.

In this work the fixed length encoding was chosen because of its similarity to genetic vector field estimation.

The notation used for the disparity path element corresponding to the co-ordinate x is:

$$DP(x), \quad \text{where } 0 \le x < W_{pix}$$

5.6.2 Path fitness

For the fitness function the mechanism of linear normalisation will be used, i.e. the paths in the population will be sorted according to increasing cost and decreasing fitness.

The main component of the cost function will again be sum of the disparity space image values along the non-occluded (horizontal) path.

Parts disparity transitions of +/- 1 are not considered as occlusions and thus are not penalised by added cost: this does however mean that crossover and mutation operators must not create such jumps gratuitously.

Each (diagonal) occlusion transition (with a disparity jump greater than 1) is assigned a cost penalty p_g.

So the total cost for a path becomes equal to the sum of disparity space image values along the non occluded path parts plus p_g times the number of occlusion jumps of magnitude greater than one.

5.6.3 Path mutation

One essential feature which can be incorporated into the path mutation operator is "best match" searching behaviour. This should be coupled to the fact that spurious noise based best matches for a single pixel are not desirable.

Also the result of the mutation must be a legal disparity path which obeys the strictures imposed by the ordering constraint (the input disparity path must be legal too).

This leads to the following description for the path mutation operator:

1. A random location x_r is chosen along the path. x_r is generated according to a uniform distribution over [0, W_{pix}>.

2. The disparity space image column x_r is searched for the d_m which corresponds to the lowest value.
3. The d found in step (2) is recursively propagated to the left and right of x_r for as far it continues to give lower or equal disparity space image values. I.e. $DP(x_r\text{-}1)$ is replaced by d_m if $DSI_i^L(xr\text{-}1,d)>DSI_i^L(x_r,d_m)$ and so forth.
4. If recursive propagation was not possible and only $DP(x_r)$ was changed then this change is undone and the operator is finished.
5. To the left and right of the x range that has been modified the path is corrected in order to make it obey the rules of the ordering constraint.

This path mutation operator is illustrated in Figure 5.11.

The x range across which d_m could be propagated is shaded. On the mutated path the disparity is set to d_m along this range. To the right of this range no correction was necessary because the discontinuous transition to a higher disparity from left to right is legal as legal transition according to ordering constraint rules. To the left of the range corrections did have to be made because the right to left transition to a higher disparity must go along a step by step diagonal. Both initial and mutated disparity paths are legal.

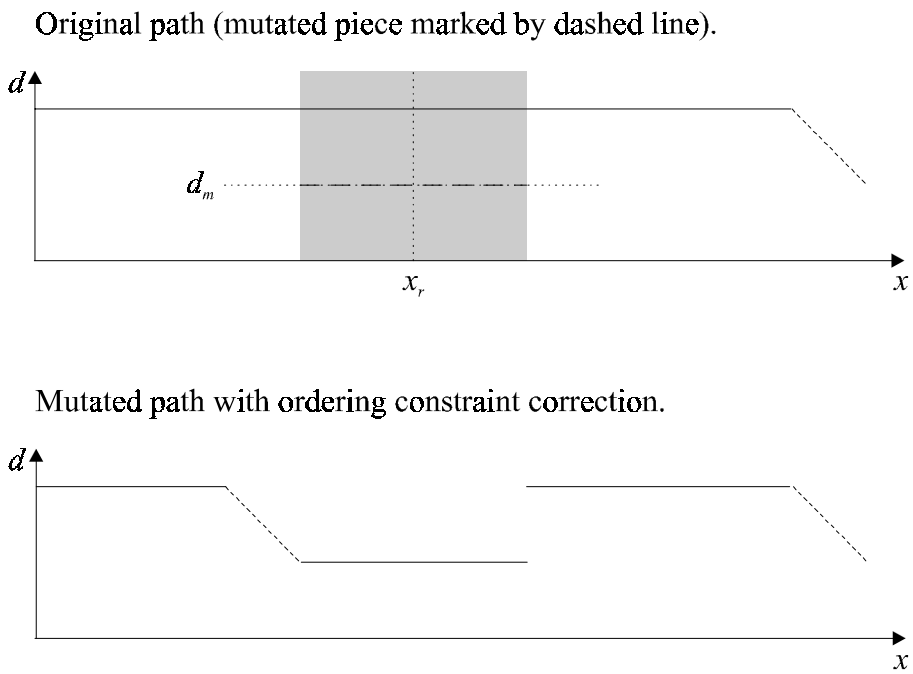

Figure 5.11: Disparity path mutation operator.

5.6.4 Path crossover

The vector field crossover operator works by identifying areas of constant disparity and exchanging these between two parent paths, performing legalisation on the

resulting child path if and where necessary. The path crossover operator for two disparity path DP_1 and DP_2 can be described as follows:

1. A single random location x_r is chosen for both of the parent paths. x_r is generated according to a uniform distribution over [0, W_{pix}>.
2. In DP_1 the contiguous x range containing x_r which shares $DP_1(x_r)$ is determined. Similarly the contiguous x range containing x_r which shares $DP_2(x_r)$ is determined in DP_2.
3. The range found in DP_1 is copied to DP_2 and the range found in DP_2 is copied to DP_1. If either range has a width of 1 (it lies on a diagonal disparity transition) then it is not copied.
4. To the left and right of the copied ranges the path is corrected in order to make it obey the rules of the ordering constraint.

Disparity path crossover is illustrated in Figure 5.12:

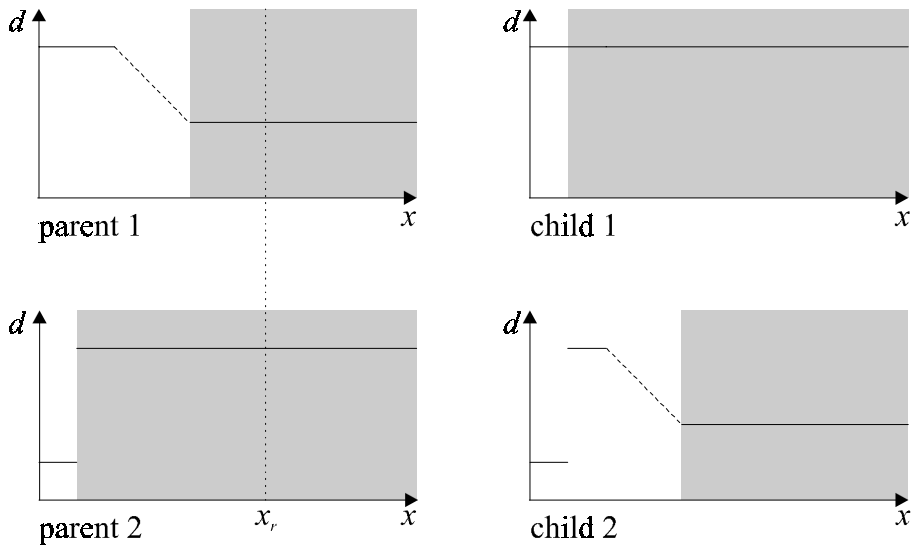

Figure 5.12: Disparity path crossover.

5.7 Experiments

As with the genetic vector field estimator correct settings for some of the parameters of the genetic disparity path finding algorithm are important. A combination of population size and number of generations which works well is a population size of 100 and 100 generations.

Steady state reproduction policy without checking for duplicates was used. Initialisation was performed with paths of random, constant, disparity level.

Just as the dynamic programming algorithm requires penalty parameters p_d and p_v the genetic algorithm for disparity path finding requires the penalty parameter p_g. The difference between the dynamic programming approach and the genetic algorithms approach shows itself most clearly in the fact that it is possible to set a

p_g , for the genetic disparity path finding algorithm, such that it works well on both the sequence “mirror” (in terms of the accuracy scores) and on the “aqua” sequence (subjective evaluation of disparity fields). This is not possible for the dynamic programming version. A p_d which works well on “mirror” does not produce good results on “aqua” or vice-versa.

The accuracy results on “mirror” are shown in Figure 5.13 and the entropy figures are plotted in Figure 5.14.

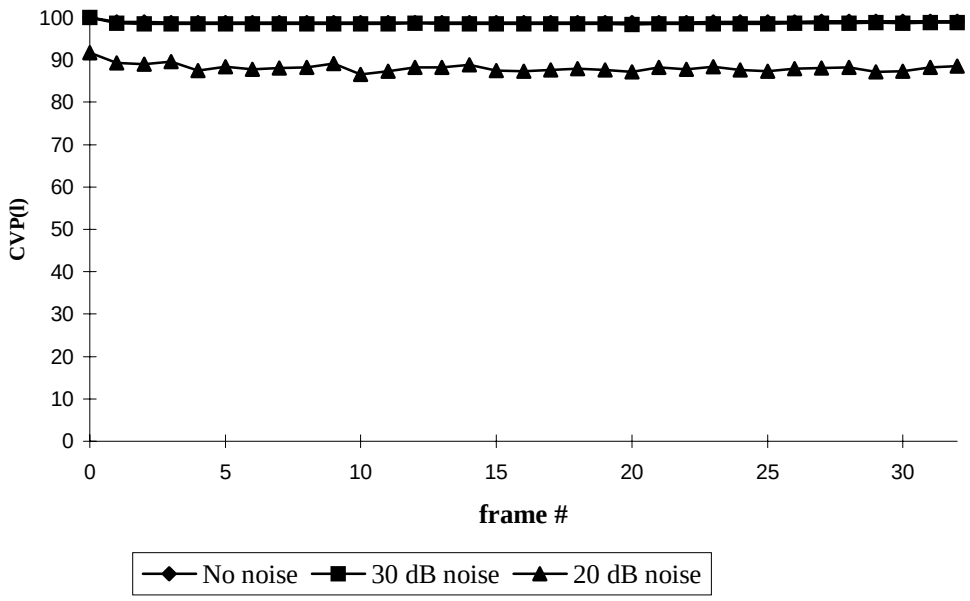

Figure 5.13: Accuracy of genetic DSI disparity estimation.

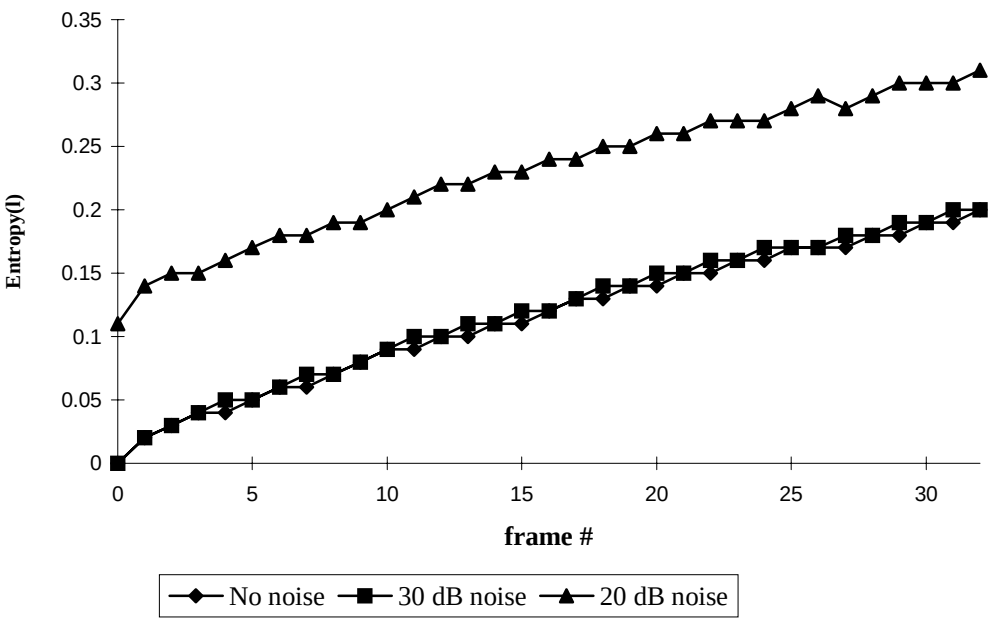

Figure 5.14: Entropy of genetic DSI vector fields.

Accuracy without noise present and at 30dB noise level is very similar. Accuracy at 20 dB noise level is even higher than for the genetic block matcher. Entropy figures are much lower than for any block matcher but it should be noted that these correspond to a dense disparity field.

The left disparity field for “aqua” is shown in Figure 5.15 while the right disparity field is shown in Figure 5.16. The significance of the grey scale values is as with the disparity fields presented in earlier chapters except that white now corresponds to detected occlusion areas.

Figure 5.15: Dense left disparity field for "aqua".

Figure 5.16: Dense right disparity field for aqua.