3. Blockmatchers

Blockmatching is a technique originally developed for motion estimation a problem very similar but not identical to disparity estimation. The most common exponents of this technique are described in Section 3.1.

The demands made on a blockmatcher, be it a motion or a disparity estimator, in the context of hybrid motion/disparity DCT coding are not necessarily the same as the demands which are made on it in image synthesis applications. The two distinct situations are discussed in Sections 3.2.1 and 3.2.2.

In Section 3.3 a synthetically generated test sequence is described and some experiments are done with simple block matchers as disparity estimators.

Finally Section 3.4 reflects on what we should look for when designing a disparity estimator which is to perform better than an adapted motion estimator.

3.1 Classic block based vector field estimators

The principle of blockmatching for motion estimation is illustrated in Figure 3.1. Motion is estimated between the two images *k-1* and *k*. In particular motion is to be

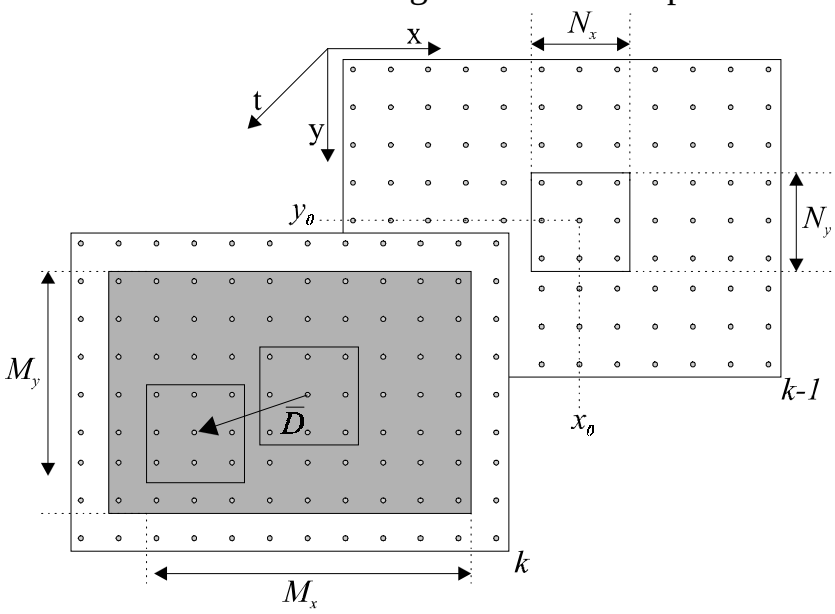

Figure 3.1: principle of blockmatching.

estimated for the reference block with centre $\vec{x}_0 = (x_0, y_0)$ in image *k-1*. The size of this reference block is N_x pixels (wide) by N_y pixels (high). A rectangular search area M of dimensions M_x by M_y is defined in image *k*. Within this search area a vector $\vec{D}(\vec{x}_0, k-1)$ is sought which minimises an error criterion C_{match}. This vector points to a location $\vec{x}_0 + \vec{D}(\vec{x}_0, k-1)$ in image *k*.

The error criterion compares the block at $\vec{x}_0$ to the block at $\vec{x}_0 + \vec{D}(\vec{x}_0, k-1)$ according to a weighting measure *W:*

$$C_{match}(\vec{x}_0, \vec{D}(\vec{x}, k-1)) = \frac{\sum_{x=x_0-N_x/2}^{x_0+N_x/2} \sum_{y=y_0-N_y/2}^{y_0+N_y/2} W(x, y, \vec{D}(\vec{x}, k-1))}{N_x \cdot N_y} \tag{3.1}$$

Two popular weighting functions for such error criteria are the absolute difference which results in a criterion known as mean absolute difference, often abbreviated to MAD:

$$W_{MAD}(x, y, \vec{D}(\vec{x}, k-1)) = \left| s_k(\vec{x} + \vec{D}(\vec{x}, k-1)) - s_{k-1}(\vec{x}) \right|. \tag{3.2}$$

Also commonly used is the mean squared error or MSE.

$$W_{MSE}(\vec{x}, \vec{D}(\vec{x}, k-1)) = \left(s_k(\vec{x} + \vec{D}(\vec{x}, k-1)) - s_{k-1}(\vec{x}) \right)^2. \tag{3.3}$$

The motion model used in blockmatching implies that:

- the motion is pure translational motion,
- the objects in the scene are rigid bodies,
- which are composed of rectangular blocks.

None of these three constraints are guaranteed to hold for arbitrary image pairs.

The dimensions, N_x by N_y, of the blocks for which vectors are assigned are an important parameter of any blockmatcher. Choosing the size of these blocks large enhances robustness against noise. However as block size increases the chances will also increase that a block overlaps two objects in the scene to which different vectors should be assigned. This effect is illustrated in Figure 3.2. Object *S* is

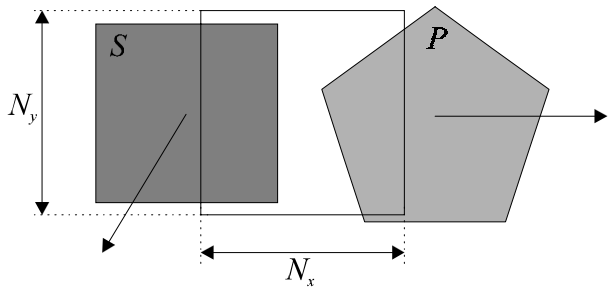

Figure 3.2: large block overlapping multiple objects.

moving down and to the left, object *P* is moving to the right while the background is static. The reference block overlaps both objects and a piece of background. Because only one vector is assigned per block there is then no way in which a completely correct solution may be found for this block. This makes block size is a matter of compromise.

What distinguishes blockmatchers from each other is the search area which is searched for a best matching block and the strategy by which this area is searched. An extensive overview of blockmatching techniques may be found in [DeHa92].

3.1.1 Full search vector field estimation

The most straightforward blockmatcher search strategy is that of the full search also known as exhaustive search. Image *k-1* is divided into a regular grid of N_{block} blocks with centres (x_i, y_i). For each block *i* a rectangular search area of fixed size is centred around (x_i, y_i) in image *k*. All candidate vectors corresponding to this search area are then examined to find the one with the lowest C_{match}.

The main problem with this search strategy occurs as the block size gets smaller. Objects which contain texture (repeating grey scale patterns) will cause a number of minima for C_{match} which do not necessarily correspond to the true motion of the object in question. The full search algorithm chooses the minimum with the smallest numerical value. A choice which is subject to considerable influence from noise. Erroneous matches made in this way are often referred to as spurious matches.

3.1.2 Recursive vector field estimation

In recursive blockmatching schemes the search area is usually much smaller than is the case with a full searcher. Usually the search area is centred around a candidate

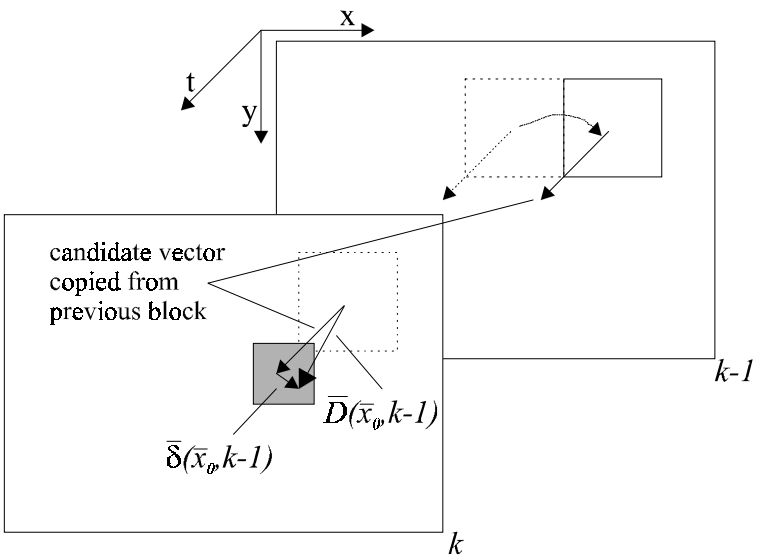

Figure 3.3: recursive blockmatching.

location which was set using the vector from previous blocks within the same or in the previous image. After this (small) search area is examined the best vector found in this search area is assigned for the block and it is passed on to the next block as an initial vector estimate. Hence the name recursive blockmatching. A basic spatially recursive scheme is shown in Figure 3.3 and is described by:

$$\vec{D}(\vec{x}_0, k-1) = \vec{D}(\vec{x}_0 - \begin{pmatrix} N_x \\ 0 \end{pmatrix}, k-1) + \vec{\delta}(\vec{x}_0, k-1) \tag{3.4}$$

In this equation $\vec{\delta}$ is known as the update vector. Convergence effects work to both the detriment and advantage of the recursive search strategy. Because a limited area is searched ($\vec{\delta}$ is small) the vector found cannot deviate a great deal from the initial, recursively passed on, candidate vector. Thus a large discontinuity (in the sense of the recursion) in the vector field will cause a convergence effect across one or more blocks. On the positive side vector fields tend to be quite spatially homogenous so that the recursively passed on initial candidate vector stands a high chance of being correct. The fact that the search area round it is limited helps to prevent to spurious matches which a full search blockmatcher would tend to make in the same situation. For recursive blockmatchers the price of vector field coherence is paid for in convergence effects.

3.1.3 Hierarchical vector field estimation

Hierarchical blockmatchers [Bier88, Bier91] perform a recursion not in a spatial or temporal sense but in resolution or block size. An illustration is given in Figure 3.4. Whilst it is the intention to estimate motion for a small block (3) initially motion is estimated for a larger block (1) which is spatially centred around (3). The search

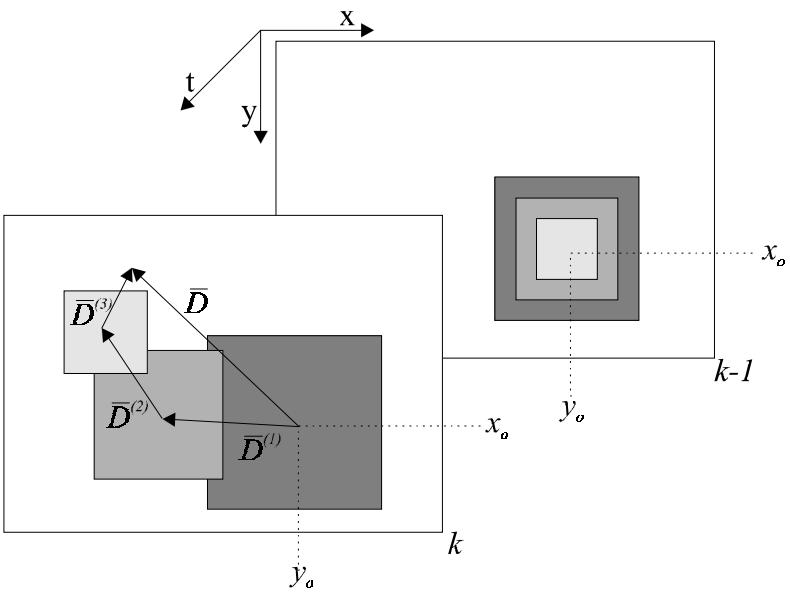

Figure 3.4: hierarchical blockmatching.

area M_3 searched to find a match for (1) is relatively large and thus it is sparsely searched. This search yields an initial vector $\vec{D}^{(1)}$. Next the block size is reduced to the intermediate size of block (2). A smaller search area M_2 is centred around the spot pointed to by $\vec{D}^{(1)}$ and searched. The result is $\vec{D}^{(2)}$. Finally the block size is reduced to the intended target block size (1). A small search area M_1 is centred

around the co-ordinates pointed to by $\vec{D}^{(1)} + \vec{D}^{(2)}$. This area is searched resulting in $\vec{D}^{(3)}$. The vector $\vec{D}^{(1)} + \vec{D}^{(2)} + \vec{D}^{(3)}$ is then assigned for the target block:

$$\vec{D}(\vec{x}_0, k-1) = \vec{D}^{(1)} + \vec{D}^{(2)} + \vec{D}^{(3)} \tag{3.5}$$

The hierarchical blockmatcher described in the above example is a three tier hierarchical blockmatcher. Three block sizes are involved and the vector which is finally assigned for the target block is the sum of three component vectors. The number of hierarchical levels usually lies between two and four.

The elegance of the idea lies in the fact that the advantages of both large and small block sizes are used. Initially a large block size is used. Only a limited subset of the grid positions in the large associated search area is searched. Noise based matches are not likely to occur. Because the search area is coarsely searched and because the initial block is so large these first hierarchical level vector estimates will not be very accurate. The initial block is then subdivided into smaller component blocks for each of which a smaller search area (centred around the position pointed to by the first level vector) is examined. If required further improvement can be made in a third hierarchical level.

Because not all errors made at a large block size can be corrected at smaller block sizes (the search range for these smaller block sizes is limited) a convergence effects which surrounds objects on all sides can occur. This effect is know as the corona effect.

3.2 Existing blockmatchers as disparity estimators

The application of existing blockmatchers to disparity estimation is straightforward. The search area reduces to a one dimensional search along the epipolar line which for the parallel camera model corresponds to the horizontal and which for the converging camera model approximates to the horizontal.

For disparity estimation the actual length of the horizontal search range would have to depend upon the camera configuration and the actual scene in question. The disparity ranges for some of the DISTIMA test sequences are given in Table 3.1.

Table 3.1: Disparity ranges for test sequences.

Sequence name	d_{min}	d_{max}	d_{max}-d_{min}
Aqua	-4	22	26
Jardin	21	32	9
Manege	4	28	24
Piano	3	11	8
Tunnel	-12	19	31

As can be seen from the table three of the five test sequences have disparity ranges which exceed 16 pixels. As a disparity estimator would have to be able to handle all of these sequences the disparity range would have to be at least equal to the maximum occurring disparity minus the minimum occurring disparity: 32-(-12)=44. Then there is also the fact that the disparity ranges in these test sequences are by no means extreme. More recent test sequences shot for video conferencing type applications have disparity ranges of over 100.

3.2.1 Coding context

When a disparity field is to be estimated for use in coding (as discussed in **{???}**) the full search algorithm is a popular and reasonable choice. Full search block-matchers which use the MSE as error criterion minimise the variance of the prediction error which works well when the prediction error is to be subsequently DCT transformed, quantised and encoded. The fact that matches do not necessarily correspond to true disparity is not a problem as the vectors are only used for prediction.

The one argument against using a full search blockmatcher for this application is the fact that erroneous, spurious, matches will be made. These matches will make the estimated disparity vector field less homogenous so that the number of bits required for the coding of the disparity vector field is increased. As this increase is relatively small and the vector fields only account for a fraction of the total bit rate in the first place this problem may well be acceptable. Particularly as the full search blockmatcher is the most commonly implemented vector field estimator in hardware.

3.2.2 Interpolation context

The application of intermediate viewpoint generation is vastly more demanding of the accuracy of the disparity vector field used than is the coding application. This is because the disparity vectors are used for the synthesis of intermediate viewpoint

images. No additional error signal is available to correct any errors made in the synthesised image. This image is presented “as is” to the viewer.

The full search blockmatcher is completely unacceptable for this application. With a full search blockmatcher spurious matches will occur. Such a spurious match vector links two blocks, one in the left and one in the right view image. Either or both blocks would then be used to synthesise a block in the intermediate view image but as the vector along which the interpolation occurs is not the proper disparity vector this synthesised block will appear in an inappropriate location, thus causing an artefact which is likely very noticeable.

3.3 Experiments

In order to be able to perform objective experiments with disparity estimators it helps to have source material where the actual disparity is known for each pixel so that objective measurements can be done. Real world test sequence material like the “aqua” sequence is much less useful for doing automatic objective measurements but inspection of the disparity map generated by a disparity estimator can still be instructive.

3.3.1 The “mirror” sequence

The synthetic sequence used in this work is known as “mirror” and is illustrated in Figure 3.5.

L_Mirror_I (frame 32) R_Mirror_I (frame 32)

Figure 3.5: Synthetic test sequence mirror.

The scene is a composite of the test scenes “clown” and “lenna”. In this composite the main part of the scene is at zero disparity while the view of Lenna through the mirror is at a disparity equal to the frame number (frames being numbered from zero).

Three versions of the scene are used. There is a master version which has not been corrupted by noise (mirror_I). Gaussian noise is added to this sequence at respectively 20dB and 30dB to give the sequences mirror_20 and mirror_30. The noise level of 30dB corresponds to a noise corruption which is visible in straight comparison with the original but otherwise hardly noticeable. The 20dB noise level is perceptible even without comparison but it still corresponds to a visually very acceptable noise level.

3.3.2 Aspects of blockmatchers

As a first experiment a full search block matcher is tried for disparity estimation on the "mirror" sequence at the three associated noise levels. The search range of this block matcher was -16…+39 so that all disparities occurring in the "mirror" sequence could be covered by it. The block size chosen was 8x8.

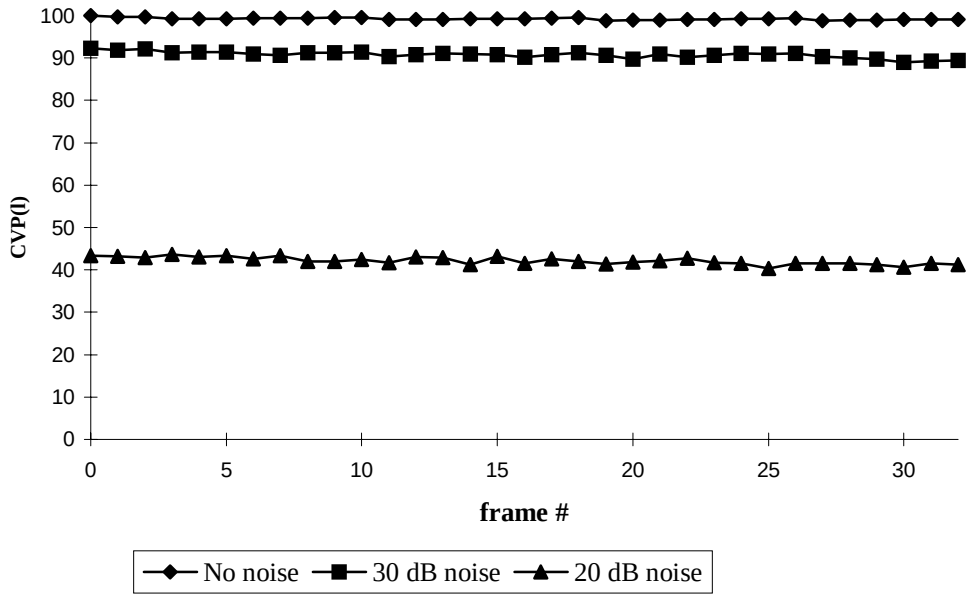

Figure 3.6: Influence of noise on full searcher accuracy.

For each frame disparity was estimated left to right (i.e. matches are sought in the right view image to correspond with the blocks at regularly spaced positions in the left image). Figure 3.6 shows the percentage of disparity vectors which are correctly estimated for each frame, *%CVP(l)*. When there is no noise in the sequence (mirror_I) the full searcher gets more than 99% of vectors right. In fact for frame 0 (disparity 0 all over the frame) where neither occlusion nor block boundaries play any part the full searcher gets 100% of vectors right.

As the noise level increases the accuracy decreases. At 30dB round 90% of vectors are estimated correctly while at 20dB this is only 40%.

The size of the disparity has little or no influence on the full searcher as it's search range covers all occurring disparities in this scene.

It is also possible to look at the entropy or the (lower bound on) the average number of bits per vector required to encode these disparity vectors when they are differentially encoded on a row by row basis. A disparity field where all vectors are equal has zero entropy while a disparity field consisting of uniformly distributed random vectors will have the highest entropy.

These entropy values are shown for the three noise levels of “mirror” in Figure 3.7.

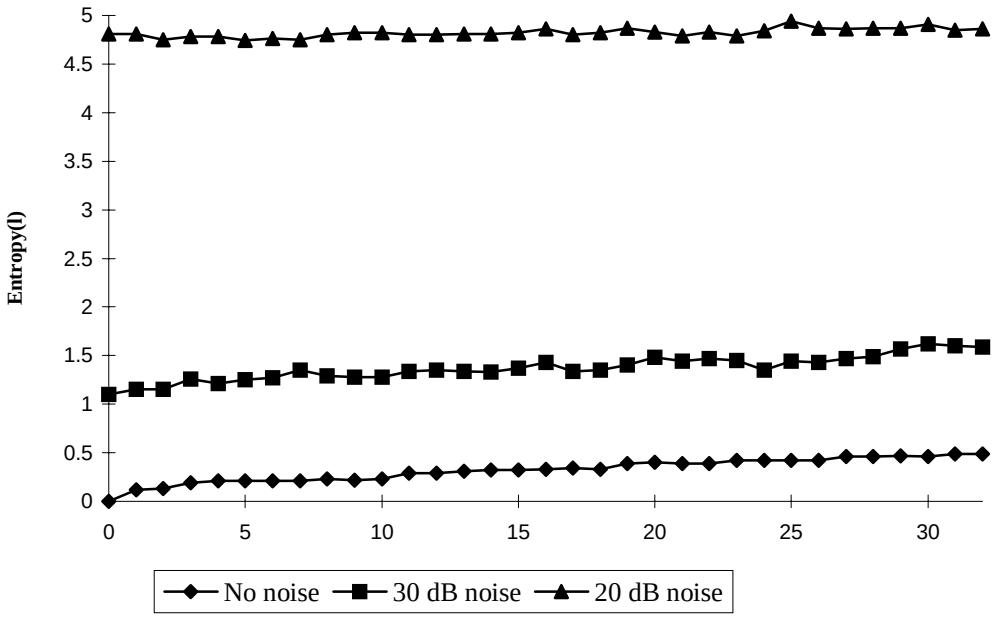

Figure 3.7: Influence of noise on full searcher disparity field entropy.

Inspection of Figure 3.7 shows that as the noise level increases the entropy of the estimated disparity field rises.

So the full search estimator performs very well when there is no noise present. The only imperfections in the disparity vector field occurring because of block boundary limitations. With noise levels of more realistic proportions the accuracy of the estimated disparity field soon drops while the entropy of this same disparity field rises.

The same indicators of accuracy and disparity field entropy are shown in Figure 3.8 and Figure 3.9 for a recursive block matcher with 8x8 block size and a +/- 4 update range.

The accuracy level obtained is similar to what the full searcher achieves. The two noticeable differences are that the accuracy of the recursive block matcher decreases with increasing frame number (i.e. with increasing disparity difference). This is because of convergence effects introduced by the limited update. Also the accuracy in the presence of noise is better for the recursive blockmatcher because the limited update has some success in preventing spurious noise based matches.

This reduction in the number of spurious matches is also reflected in the disparity field entropy.

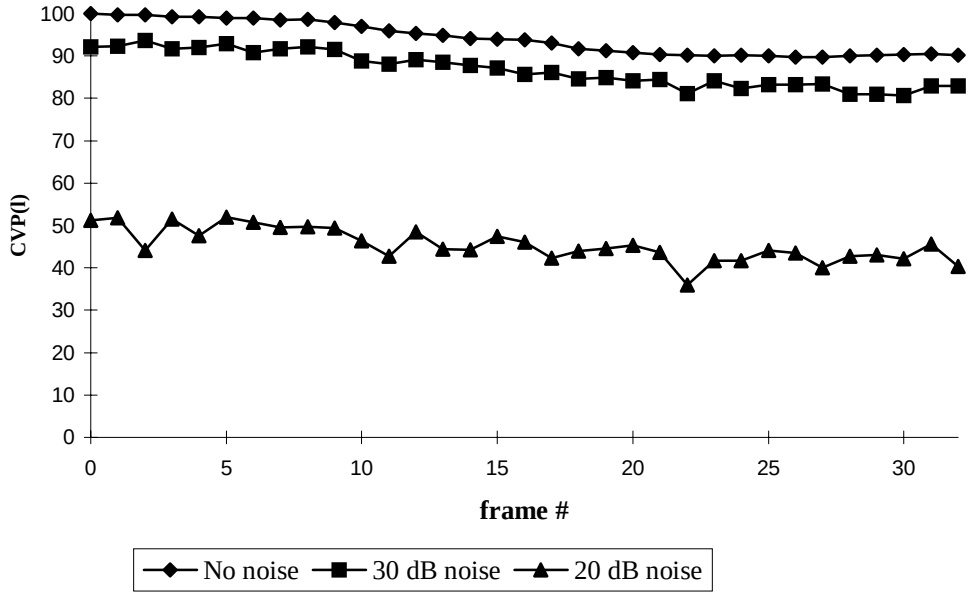

Figure 3.8: Influence of noise on recursive block matcher accuracy.

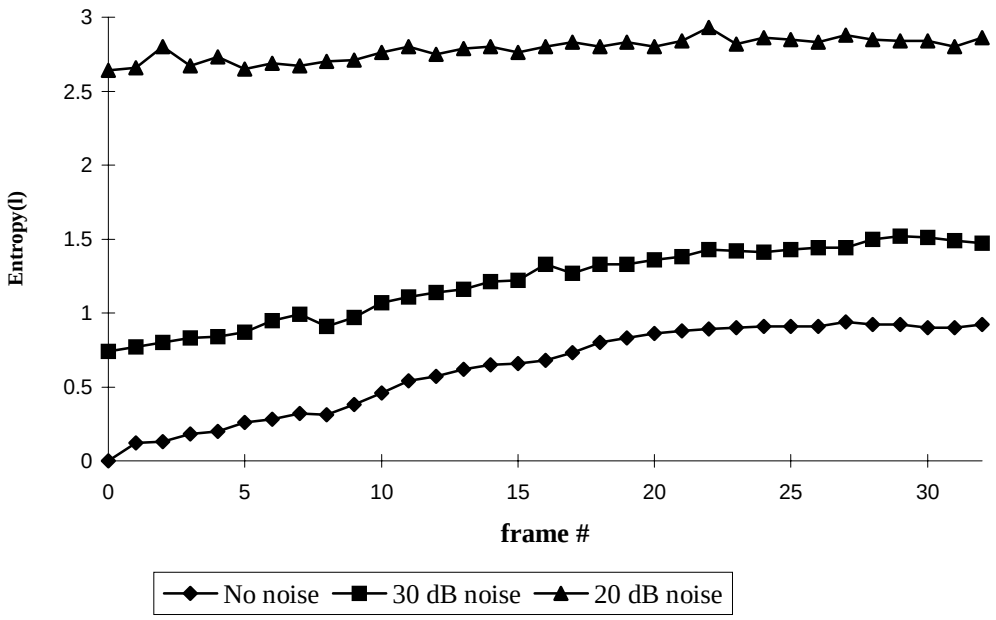

Figure 3.9: Influence of noise on recursive disparity field entropy.

These same estimators were used to estimate a disparity field for the sequence "aqua". This sequence "aqua" is not synthetic as is "mirrror" so that accuracy cannot be calculated. It is however possible to present the calculated disparity fields as grey scale images. The full search block matcher generated disparity field for the

first frame of aqua is shown in Figure 3.10 while the disparity field generated by the recursive blockmatcher for that same frame is shown in Figure 3.11.

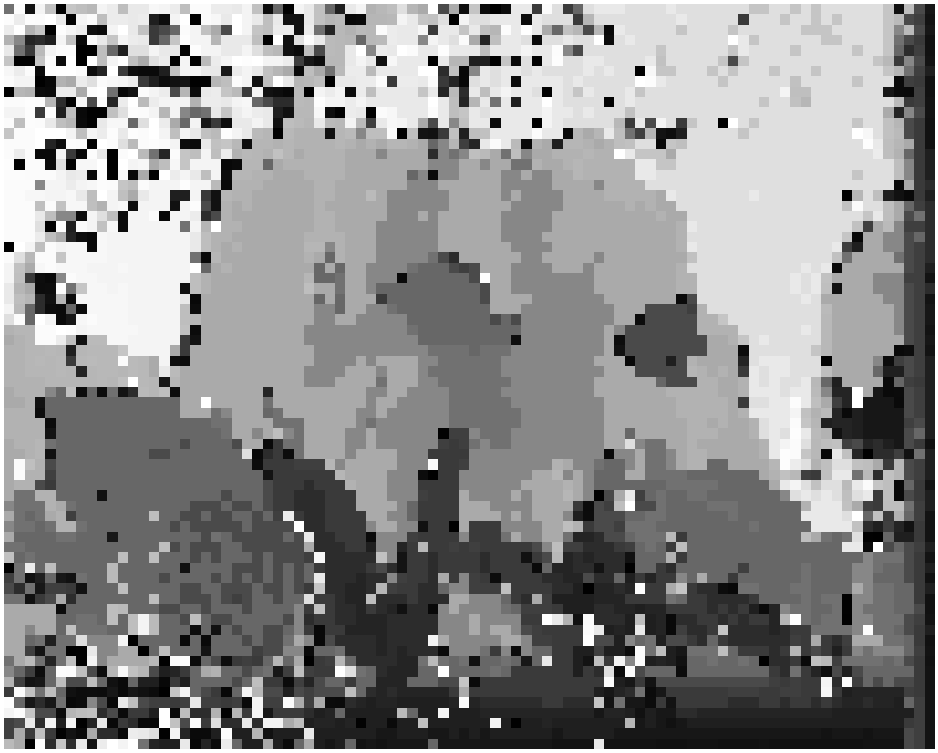

Figure 3.10: Full search generated disparity field for "aqua".

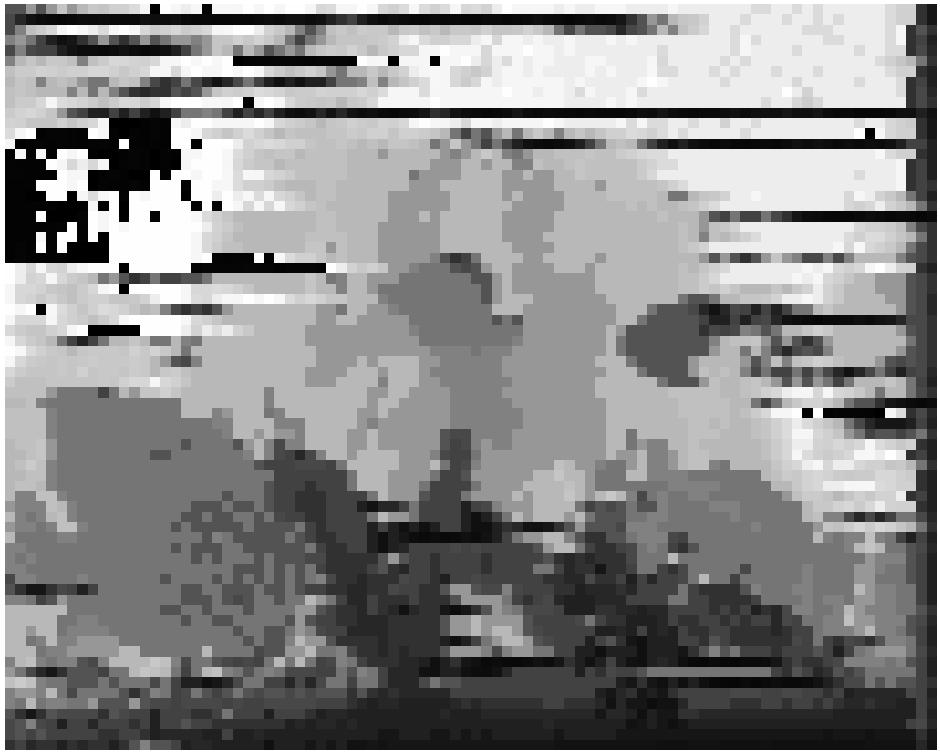

Figure 3.11: Recursive search generated disparity field for "aqua".

The grey scale values in these two figures have been histogram-equalised in order to make them better distinguishable on the printed page. Dark grey scale values corresponds to objects nearer to the camera while lighter values correspond to objects further away from the camera.

The full search generated disparity field shows a lot of spurious vectors which are obvious on inspection. The disparity field generated by recursive search show much fewer of these spurious vectors. The constraints imposed by the recursive search technique prevent most spurious matches from ever being examined at all. Unfortunately the same constraints also causes the “horizontal stripe” artefacts in the disparity field shown in Figure 3.11.

3.4 Discussion: towards a better disparity estimator

If a disparity estimator is going to produce disparity vector fields which correspond to the real disparity of objects within a then a straightforward application of the full search algorithm is not acceptable. Such accurate fields are a necessity if intermediate viewpoint generation is to be attempted.

In the presence of non-negligible noise the full search blockmatcher will produce spurious matches. Recursive and hierarchical schemes are better at avoiding these but the artificial constraints which these schemes impose cause undesirable artefacts in the estimated vector fields.

Also using a blockmatching algorithm originally conceived for motion estimation as a disparity estimator, just constraining the vertical vector component, is not in any way making use of the knowledge we have of the specific properties of the disparity field.

A better disparity estimator should not cause artefacts in the estimated field by over constraining the solution and constraints should be as much as possible based on inherent disparity field properties.