Regularized Block Matching Using Control Points

Regularized Block Matching Using Control Points

Zhong-Dan Lan and Janusz Konrad

Université du Québec
Institut national de la recherche scientifique
INRS-Télécommunications
16, place du Commerce, Verdun
Québec, Canada, H3E 1H6

June 30, 1999

Rapport technique de l'INRS-Télécommunications no. 99-13

Abstract

In this report, we present a block matching method using control points and its application to intermediate view reconstruction (IVR).

In the entertainment industry, stereoscopic video is used for providing 3-D movies to the public. In the case of broadcast TV or computer monitors, where much smaller screens are used, observer head motions pose a big problem. This problem is solved through the reconstruction of intermediate or virtual views.

The IVR is based on disparity estimation, that is performed as a function of the desired intermediate view position. The resulting disparity field is used to reconstruct the intermediate view position.

For the disparity estimation, both pixel-based and block-based matching are possible. However, in the case of large disparities, the pixel-based method, which uses relaxation, can fail because of poor initialization. In this case, block matching method is more appropriate. Since the disparity vectors are unreliable in poorly-textured areas, block-based matching with spatial smoothness constraint is used.

However, adjacent pixels belonging to objects from different depth planes should not be forced to have similar vectors. This problem is inherent in all regularization approaches. We can reduce this problem using control points.

Since the control points can be matched reliably, hence they should not be affected by the regularization. We obtain the control points by matching interest points.

Interest points are low-level features where the signal changes two-dimensionally and where the intensity/color information is abundant. Thus, these points are much more reliable than low- or medium- textured points.

This report is organized as follows. After this introduction, first, we present block matching with and without regularization. Then we present the interest points extraction. Then, we present sparse matching of these points. Finally, we describe regularized block matching method using control points.

Contents

Chapter 1

Introduction

1.1 Stereoscopic and 3-D imaging

There is a lot of work in the television broadcast industry concentrating on high definition TV. The ongoing search for increased realism in video applications has been recently directed towards 3-D imaging to incorporate the sensation of depth.

3-D video is a particular form of 3-D imaging that provides the viewer with supplementary information needed for realistic depth perception of moving images. *Stereoscopic* video is a particular type of 3-D video, where the depth information is provided by two slightly displaced views of the same scene. Consider that observer's eyes view the world from two slightly different angles. Human's brain uses differences in the two acquired projections to perceive the depth of a scene. In similar way, a stereoscopic video system obtains two views from cameras which are slightly displaced, much like the relative location of our eyes. Each acquired view is then projected onto the corresponding eyes retina, and it is through the combination of data from both views that our brain perceives depth. In multi-view video, stereo scenes are captured from several viewpoints, offering a larger viewing angle to the viewer. Only two images are presented at one time as the user selects the stereo image pair that offers the desired perspective.

1.2 Intermediate view reconstruction

Since two views of a scene must be captured, a stereoscopic camera consists of two lenses slightly displaced one from the another. This makes stereoscopic cameras impractical to move around: they are usually large and heavy. 3-D film makers are

sensitive to this and need to be more selective to the scenes they capture. Additionally, the amount of data is doubled in comparison with a regular video camera having only one lens. This doubling of information places heavy demand on the transmission bandwidth. Stereoscopic video compression techniques that exploit correlation between the two perspective views are used to reduce the problem. Methods based on *disparity-compensated prediction* are used for this purpose. Disparity is referred to as the difference of the positions of *homologous* points in two images, i.e., points resulting from the projection of the same 3-D point onto the two image planes. *Disparity estimation* is the process of estimating the disparity in one image with respect to the other. The disparity field is the set of disparity vectors which provide the mapping between images.

While acquiring a stereoscopic image, left and right cameras are fixed in space. Therefore, the two views depict a 3-D scene from a particular viewing angle. If the stereoscopic pair is not viewed from the intended angle, an unnatural representation of the scene will result. The distance between the two stereoscopic lenses is also fixed, and hence it is not guaranteed to match the viewing characteristics of every viewer. The problems with current stereo video systems due to fixed relative positions of cameras need to be addressed.

In the entertainment industry, stereoscopic video is used to provide 3-D movies to the public. For high-quality stereoscopic movies on large screens, viewer head movements are negligible compared to the size of the screen and the distortions resulting from slight head motion are negligible. However, in the case of broadcast TV or computer monitors, where much smaller screens are used, viewer head movements become a big problem.

This problem is solved digitally through *intermediate view reconstruction* (IVR). Intermediate views permit the display of the same scene but from a different viewing angle. As the viewing angle changes, corresponding views are estimated and displayed, and subsequently the distortions are reduced. Intermediate view reconstruction also offers the possibility of adjusting the distance between cameras to suit a particular viewer preference (*parallax adjustment*).

	3-D based method	2-D based method
correspondence	needed	needed
camera calibration	needed	not needed
complex scene	cannot handle	can handle
small-baseline	not required	required

Table 1.1: Comparison of 3-D based and 2-D based intermediate view reconstruction methods.

1.3 How intermediate view reconstruction can be performed

IVR can be performed using either 3-D model-based techniques or 2-D signal-based techniques [9]. Methods based on 3-D modeling attempt to recover 3-D representation of the scene from the left and right images, and then perform a suitable projection onto an arbitrarily-positioned virtual camera. These methods are flexible, as it allows arbitrary location of the true and virtual cameras, however it is limited by the complexity of the scene and usually requires calibrated cameras. Therefore, the approach works only for simple scenes.

2-D signal-based methods stay in the domain of 2-D signals instead of attempting to recover a 3-D representation. Usually, such methods first establish a correspondence between homologous points via disparity estimation. This correspondence can be used to recover the 3-D representation in the calibrated case. In the uncalibrated case, *disparity-compensated interpolation* is used instead to directly reconstruct the virtual camera images. This approach works well only for small baseline stereo cameras, but it is capable of handling quite complex scenes. The comparison of two methods is summarized in the Table 1.1.

This paper describes a 2-D signal based method, that can handle complex scenes.

1.4 Correspondence and disparity estimation

For both 2-D signal and 3-D based methods, the first step of IVR is to solve the correspondence problem. This problem is formulated as a disparity estimation, that can be represented mathematically as a minimization. Given two images I_1 and I_2, for each token (pixel or block) in the reference image, we find a disparity vector $\hat{\mathbf{d}}(k,l)$, that minimizes some cost function ϵ, i.e.,

$$\hat{\mathbf{d}}(k,l) = arg \min_{\mathbf{d}(k,l)} \epsilon(d(k,l)). \tag{1.1}$$

Once $\hat{\mathbf{d}}(k,l)$ is determined for each $(k,l) \in I_1$, we are left with a vector field that completely describes image I_1 in terms of image I_2.

The problem of disparity estimation is similar to that of motion estimation, that is used mainly for temporal redundancy elimination in image sequences. Motion estimation methods can also be applied to disparity estimation, since the difference between them is that motion estimation considers images at different times, and disparity estimation deals with images at the same time from different perspectives.

In the case of IVR, disparity estimation can be performed using block-based or pixel-based algorithms.

Block-based algorithms select as the matching token a block of arbitrary size. The correspondence problem is thus reduced to pairing blocks of pixels between the reference and current images. The resulting field assigns the same disparity vector to all pixels of a block, resulting in a low-resolution vector field.

1.5 How to perform interpolation

Pixel in the intermediate image I_I can be computed using disparity-compensated linear filtering with a two-coefficient kernel. A weighted average of the corresponding picture points in the left and right images, according to the disparity is used, which will be described in detail in Chapter 6.

Non-linear filtering may also be used for image reconstruction, such as "winner take all" approach. It can be argued that the weighted average used in the linear approach results in a blurred image reconstruction. The advantage of the "winner take all" approach is that without averaging (reconstruction based on either left or right image), the detail of the original data is maintained in the intermediate image. The disadvantage is that neighboring blocks, in a block-based approach, could be reconstructed from different images. Hence, any luminance and chrominance differences in the stereoscopic pair could result in a patchy image reconstruction.

The weighted-average approach results in a reconstructed image that is slightly blurred throughout, but in a block-based scheme, avoids patchiness resulting from image-pair mismatches. In addition, this approach preempts the need for developing a decision criterion for deciding from which image, to reconstruct a given pixel in the intermediate image. For these reasons, the weighted-average approach is chosen for

the purpose of intermediate view reconstruction in this report.

Chapter 2

Problems with the estimation of large disparities

2.1 Pixel-based matching

Motion or disparity estimation is an ill-posed problem since the existence, uniqueness, and stability of solutions cannot be guaranteed in the absence of additional constraints. Typically, regularization theory is used whereby an additional smoothness constraint is used to restrict the space of acceptable solutions to smooth vector fields.

The computation of disparity consists of the minimization of a penalty functional, $\mathcal{P}(\mathbf{d})$, that measures the intensity matching error, plus a regularization term $\mathcal{R}(\mathbf{d})$, where $\mathbf{d} = (d_1, d_2)$ is a disparity function. A multiplier λ is introduced to control the compromise between intensity matching error and smoothness error.

The minimization is carried out over all pixels and results in a dense disparity map, i.e., one vector per pixel. The map is obtained by minimizing

$$\epsilon(\mathbf{d}) = \sum_{(i,j)\in I_L} \mathcal{P}(\mathbf{d}) + \lambda \mathcal{R}(\mathbf{d})$$

where

$$\mathcal{P}(\mathbf{d}) = \left(I_1(i,j) - I_2(i + d_1(i,j), j + d_2(i,j))\right)^2$$

and

$$\begin{aligned} \mathcal{R}(\mathbf{d}) &= \left(d_1(i,j) - d_1(i,j-1)\right)^2 + \left(d_1(i,j) - d_1(i-1,j)\right)^2 \\ &+ \left(d_2(i,j) - d_2(i,j-1)\right)^2 + \left(d_2(i,j) - d_2(i-1,j)\right)^2 \end{aligned} \tag{2.1}$$

To find the solution, we need

$$\frac{\partial \epsilon(\mathbf{d})}{\partial \mathbf{d}(i,j)} = 0 \qquad \forall (i,j).$$

This can be solved by iterative Gauss-Seidel relaxation.

2.2 Block matching

In the context of block matching, the cost function ϵ equals U_m, i.e., $\epsilon(d) = U_m(d)$ where

$$U_m(\mathbf{d}(k,l)) = \sum_{(i,j)\in B(k,l)} f(I_1(i+\alpha d_1(k,l), j+\alpha d_2(k,l)) - I_2(i-(1-\alpha)d_1(k,l), j-(1-\alpha)d_2(k,l))) \tag{2.2}$$

A popular choice for f is the absolute value: the sum of absolute differences (SAD) results as the estimation criterion.

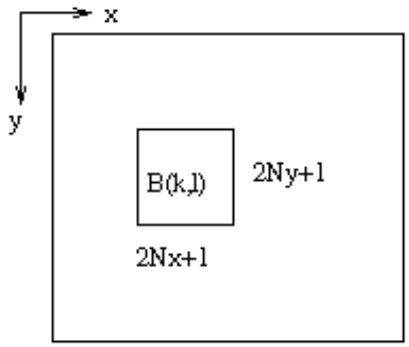


Figure 2.1: Coordinate system used for block matching.

Consider a coordinate system as shown in Figure 2.1. The image is divided into blocks of size $(2N_x + 1) \times (2N_y + 1)$. For each block (k,l) of the image, and given a set of candidate disparity vectors, the optimal vector $\hat{\mathbf{d}}$ is the one that minimizes the functional ϵ over each block.

$$\hat{\mathbf{d}}(k,l) = arg \min_{\mathbf{d}(k,l)} \epsilon(I_1(B(k,l), I_2(B(k,l) + d(k,l)).$$

The minimization can be performed either by an exhaustive or simplified search.

Table 2.1: Comparison of dense matching, block matching and sparse matching

	dense matching	block matching	sparse matching
resolution	very good	acceptable	poor
implementation	gradient based	exhaustive research	exhaustive research
initialization	should be good	not required	not required
large disparity	cannot handle	can handle	can handle

2.3 Comparisons

The resolution of disparity field depends on the method.

Pixel-based disparity estimation (*dense matching*) has good resolution, but it is usually based on gradients (complex implementation), requires a good initialization (which is typically zero everywhere), and it cannot handle large disparities, unless multiresolution implementation is used.

Block-based disparity estimation (*block matching*) has poorer resolution, but it can handle larger disparities, since an exhaustive search is used typically for implementation.

Sparse correspondence via feature matching (*sparse matching*) has very poor resolution, but is quite reliable for large disparities, since the points to be matched are judiciously selected.

Table 2.1 summaries the comparison.

To exploit strengths of different methods, we propose to establish sparse correspondence first and use it in a dense disparity estimation (Figure 2.2)

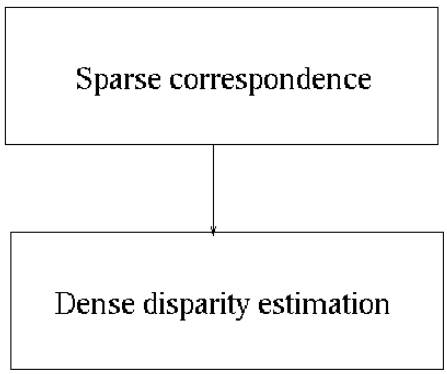


Figure 2.2: From sparse matching to dense matching.

Chapter 3

Block matching with regularization

3.1 Exhaustive search block matching

The exhaustive-search block matching algorithm, often used in video compression for motion estimation, is adapted here to perform disparity estimation for intermediate view reconstruction. The intermediate image is broken up into an integer number of blocks of fixed size $(2N_x + 1) \times (2N_y + 1)$. Then, for each block in the intermediate image, exhaustive search is performed over all candidate disparity vectors. In general, for each block at (k, l), the set of candidate disparity vectors $\mathcal{D}$ is made up of all vectors within the search range given by

$$\mathcal{D} = \{\mathbf{d} = (d_1, d_2)^T \in Z^2 | d_1^{min} \leq d_1 \leq d_1^{max}, d_2^{min} \leq d_2 \leq d_2^{max}\}$$

where Z is the set of all integers. For each block in the intermediate image, the algorithm searches for the best matching pair of blocks in I_1 and I_2 among all candidate pairs defined by the vectors in D. To do this, a cost ψ related to the difference between the blocks of the pair, defined by each candidate vector, is computed, and the one that results in the lowest value is chosen as the estimate. The cost function ψ accumulates individual error functions f that should be even (symmetric with respect to $x = 0$) and monotonically increasing for $x > 0$.

For each block (k, l) in the intermediate image (at position α), we find the optimal disparity vector $\mathbf{d}(k, l)$ by performing the following minimization:

$$\min_{\mathbf{d}(k,l) \in \mathcal{D}} \epsilon((\mathbf{k}, \mathbf{l})) \qquad \forall (k, l) \tag{3.1}$$

where $\epsilon(\mathbf{d}(k, l))$ is defined in equation (2.2).

3.2 Block matching with regularization

The block matching method works well in textured areas. To overcome the ambiguous matches in poorly-textured areas, we allow correct matches to influence incorrect matches. Since the overall vector fields obtained thus far are mostly accurate, reliable vectors can be propagated into low-textured areas where vectors are typically unreliable. Mathematically, this is done through *regularization*, e.g., a smoothness constraint. Regularization penalizes disparity vectors that are very different from their neighbors. This forces a local similarity between adjacent vectors.

However one should be careful not to impose too much regularization, since adjacent pixels belonging to objects from different depth planes should not be forced to have similar disparity vectors. Smoothness across discontinuities is inherent in any regularization approach. In our case, we can use control points obtained by sparse matching to reduce severity of the problem.

The minimization (3.1) is modified to accommodate the regularization (smoothness) term to give

$$\epsilon(\mathbf{d}(k,l)) = arg\min_{\mathbf{d}} \sum_{k,l} U_m(\mathbf{d}(k,l)) + \lambda U_s(\mathbf{d}(k,l))$$

where $U_m(\mathbf{d}(k,l))$ is defined in equation (2.2) and $U_s(\mathbf{d}(k,l))$ is the sum of absolute differences between $\mathbf{d}(k,l)$ and its four neighbors. The parameter λ controls the compromise between the closeness of the solution to the original data and the degree of smoothness. Given a candidate disparity vector $\mathbf{d}(k,l)$, $U_s(\mathbf{d}(k,l))$ is defined as

$$U_s(\mathbf{d}(k,l)) = \sum_{|k-m|+|l-n|=1} |d_1(k,l) - d_1(m,n)| + |d_2(k,l) - d_2(m,n)|.$$

Computationally, the introduction of regularization into the minimization transforms the problem into an iterative algorithm; vectors influence each other because of the U_s term, and a few iterations are required in order to reach convergence. The stopping criterion for the algorithm is typically a fixed number of iterations or convergence rate. Here, we use a threshold for the maximum number of iterations and also the convergence rate; the process stops when a preset number of iterations or a convergence rate is reached.

3.3 Experimental results

The results of block matching on three pairs of images (Figure 3.1, 3.2 and 3.3) are shown in this section. Results for block matching without regularization (Figure 3.4 (a), 3.6 (a), 3.5 (a)) and with regularization (Figure 3.4 (b, c), 3.6 (b, c), 3.5 (b, c)) are included. Two regularization constants are used: 0.1 and 1.

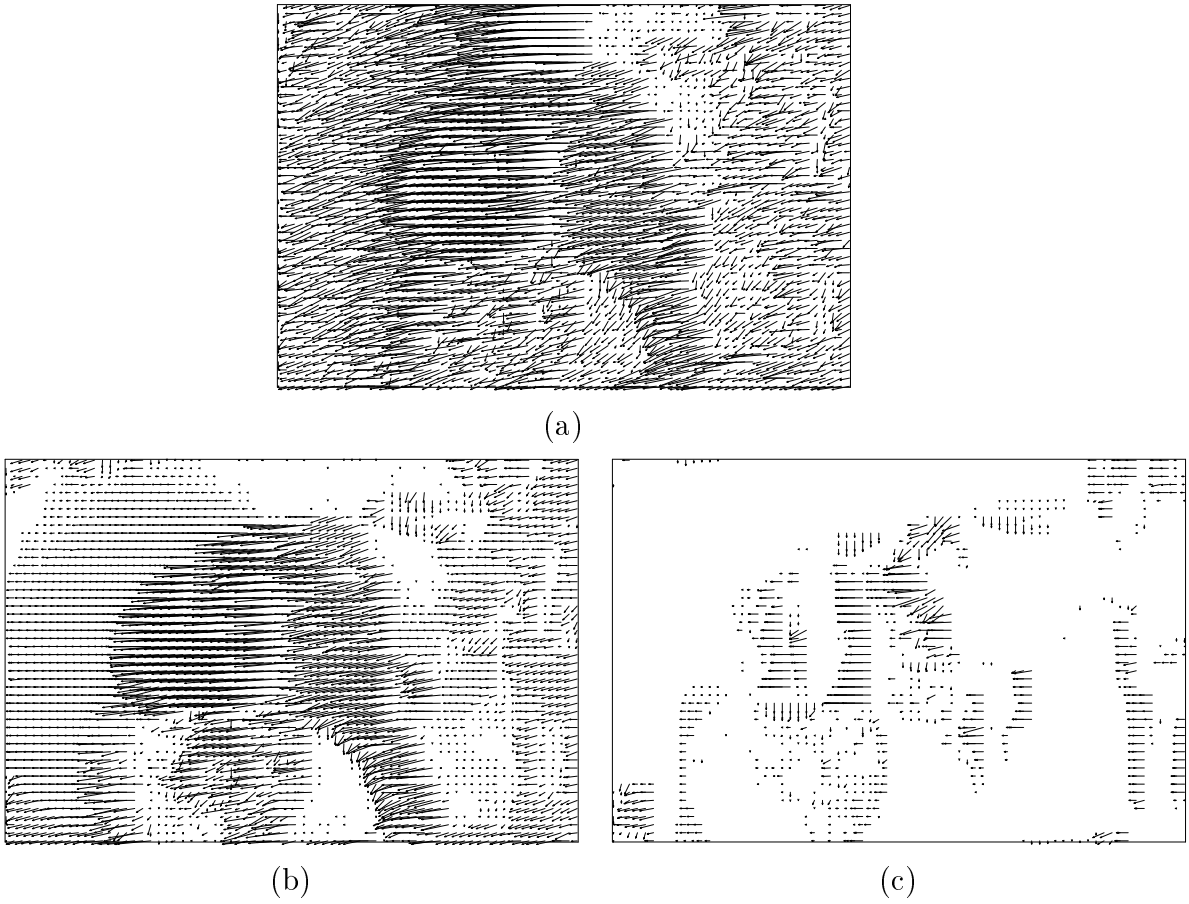
The block size is 8×8 for every image matching. The disparity field is subsampled by 16×16 for image *Basket* and 8×8 for *Piano* and *Flower*.

We see without regularization, the result is noisy and with high regularization the result is smooth, but discontinuity and details are lost.

Figure 3.1: Basketball image pair.

Figure 3.2: Piano image pair.

Figure 3.3: Flower image pair.



(a)

(b) (c)

Figure 3.4: Disparity field for Basket (a) no regularization; (b) regularization $\lambda = 0.1$; (c) regularization $\lambda = 1$.

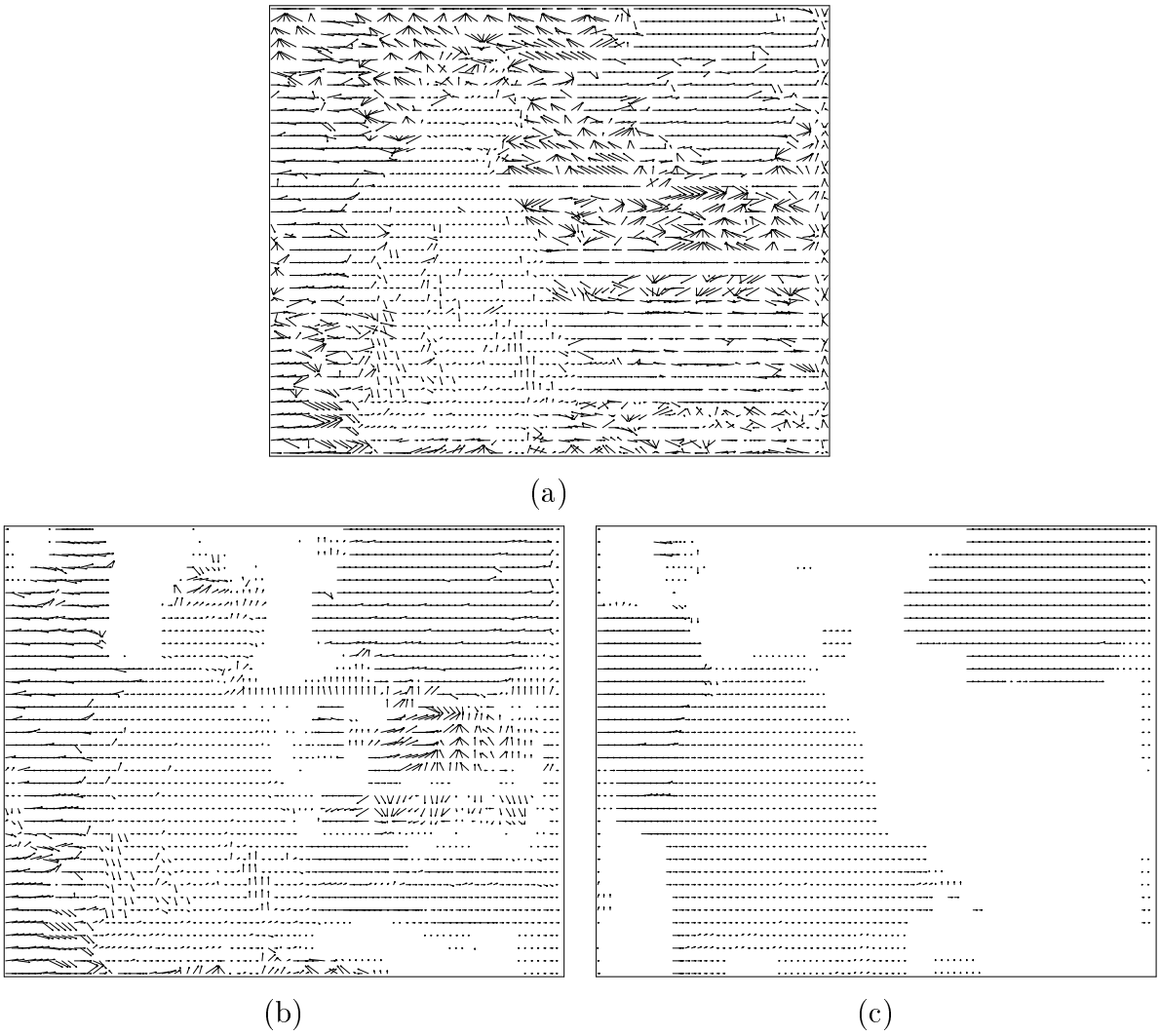


Figure 3.5: Disparity field for Piano (a) no regularization; (b) regularization $\lambda = 0.1$; (c) regularization $\lambda = 1$.

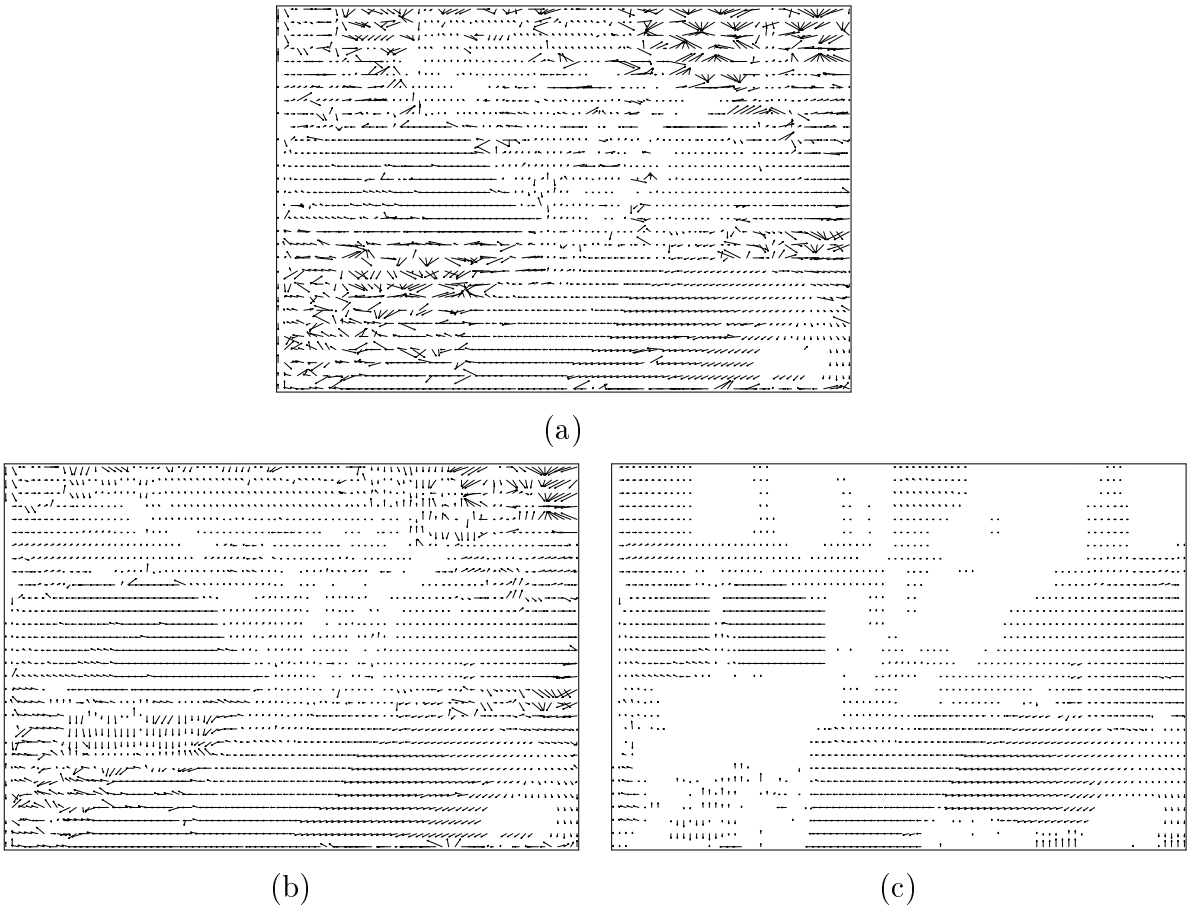


Figure 3.6: Disparity field for Flower (a) no regularization; (b) regularization $\lambda = 0.1$; (c) regularization $\lambda = 1$.

Chapter 4

Review of interest point extraction

A wide variety of interest point extraction methods exist. They can be grouped into three categories. Methods in the first category extract contours and search for maximum curvature or inflexion points along these contours. Methods in the second category extract interest points directly from the grey-level signal, and those in the third category fit a template to the signal.

4.1 Contour-based methods

Contour based methods either search for the maximum curvature or the inflexion points along the contour chains, or perform a polygonal approximation of the contour and search for particular points such as intersection points of the line segments.

Asada and Brady [1] extract interest points for 2-D objects from their curves. A similar approach has been developed by Mokhtarian and Ackworth [11]. Medioni and Yasumoto [10] use B-splines to approximate the contours. Interest points are defined by the maxima of curvature computed from the coefficients of these B-splines. Horaud [7] extracts line segments from the image contours. These segments are grouped and intersection of grouped line segments are the interest points.

4.2 Signal-based methods

The signal-based methods compute a measure, that indicates the presence of an interest point directly from the signal.

Beaudet [2] developed the first signal-based interest point extractor. He used the second derivatives of the signal for computing the measure

$$\kappa = I_{xx}I_{yy} - I_{xy}^2$$

where I_{xx} and I_{yy} are second-order derivatives of the image $I(x, y)$. This measure is invariant to rotation and is related to the Gaussian curvature of the signal. Points where this measure achieves maximum are interest points. Kitchen and Rosenfeld [8] present an interest point detector that uses the curvature of planar curves and the gradient magnitude of the image. They proposed the following measure

$$\kappa = \frac{I_{xx}I_y^2 + I_{yy}I_x^2 - 2I_{xy}I_xI_y}{I_x^2 + I_y^2}.$$

The Moravec detector [12] is based on the auto-correlation function of the signal. This function measures the differences between a window of the signal and its four neighboring windows. If the minimum of these four differences is superior to a threshold, an interest point is present.

Harris [6] has improved the approach of Moravec by calculating a matrix which is related to the auto-correlation function. Compared to the Moravec's approach, it does not use discrete directions and discrete shifts. This matrix averages the first derivatives of the signal in a window

$$\hat{C} = \begin{bmatrix} \hat{I_x^2} & \hat{I_xI_y} \\ \hat{I_xI_y} & \hat{I_y^2} \end{bmatrix} \tag{4.1}$$

where $\hat{}$ denotes convolution, i.e. a filtered I_x^2, I_xI_y and I_y^2, shown as follows:

$$\hat{I_x^2} = e^{-\frac{x^2+y^2}{2\sigma^2}} \otimes I_x^2,$$
$$\hat{I_xI_y} = e^{-\frac{x^2+y^2}{2\sigma^2}} \otimes I_xI_y,$$
$$\hat{I_y^2} = e^{-\frac{x^2+y^2}{2\sigma^2}} \otimes I_y^2$$

This detector was used in our extraction and will be detailed in Chapter 5.

Forstner [5] proposed a feature extraction method based on local statistics of the image function. Image pixels are classified into categories - region, contour and interest point - by using the auto-correlation function. The use of statistics allows a blind estimation of signal-dependent noise variance and thus an automatic selection of thresholds.

Heitger [4] proposed to extract 1D directional characteristics by convolving the image with orientation-selective Gabor filters. In order to obtain 2D characteristics he computed the first- and second-order derivatives for all of the 1D characteristics.

4.3 Template fitting methods

The template fitting methods are used to obtain sub-pixel accuracy. Typically fitting of a parametric model of a specific type of interest point, such as a corner or vertex, is performed on the signal. Clearly, the approach is not applicable in a general context. Examples of such methods are reported in [13] and [3].

Schmid [14] has compared these methods using their repeatability, i.e., the independence of the changes in the imaging conditions: image rotation, scale change, variation of illumination, viewpoint change and noise of the imaging system. In all cases the results of the Harris detector are better or equivalent to those of the other methods. We detail this detector in the following chapter.

Chapter 5

Dense disparity from feature extraction and regularized block matching.

5.1 Extraction of points of interest

In our application, we use the Harris operator as follows:

$$R(x,y) = det[\hat{C}] - k(trace[\hat{C}])^2$$

where $\hat{C}$ is defined in equation (4.1), *det* stands for the determinant and *trace* stands for the trace (sum of diagonal elements) of the matrix.

Harris detector is an improvement of Moravec's detector, which relies directly on gray-level images. The idea is to model different regions by local likelihood of the region. We can classify the regions as follows:

- For a uniform region, which has almost constant gray-level everywhere, whatever the displacement is, the local likelihood is very strong.
- For an edge, when we move in the contour direction, the gray level does not vary too much.
- For an interest point, the gray level changes in all directions.

Moravec gave a mathematic form for the above idea. Suppose I is the image gray-level, and E the local likelihood function. Then the change of image gray level due to displacement (x, y) is given by

$$E(x,y) = \sum_{(u,v)\in\mathcal{P}} \omega(u,v)(I_{x+u,y+v} - I_{u,v})^2$$

where ω are coefficients of a filter defined over mask $\mathcal{P}$. For Moravec, the displacement (x,y) are

$$\{(1,0),(-1,0),(0,1),(0,-1)\}.$$

Harris has improved the Moravec operator by considering all possible small displacements instead of only a discrete subset of displacements. This can be achieved by Taylor expansion of $E(x,y)$:

$$E(x,y) = \sum_{u,v} \omega(u,v)(x\frac{\partial I(u,v)}{\partial x} + y\frac{\partial I(u,v)}{\partial y} + o(x,y))^2$$

$o(x,y)$ is a function, such that $\lim_{(x,y)\to(0,0)} o(x,y)/\sqrt{x^2+y^2} = 0$.

When x and y are small, the term $o(x,y)$ can be neglected, and $E(x,y)$ can be approximated by:

$$E(x,y) \approx \sum_{u,v} \omega(u,v)(x\frac{\partial I(u,v)}{\partial x} + y\frac{\partial I(u,v)}{\partial y})^2$$

that can be written as $E(x,y) = Ax^2 + 2Bxy + Cy^2$, where

$$A = \sum_{u,v} \omega(u,v)(\frac{\partial I(u,v)}{\partial x})^2$$

$$B = \sum_{u,v} \omega(u,v)\frac{\partial I(u,v)}{\partial x}\frac{\partial I(u,v)}{\partial y}$$

$$C = \sum_{u,v} \omega(u,v)(\frac{\partial I(u,v)}{\partial y})^2$$

The derivatives I_x and I_y are estimated by the convolution of I with the derivatives of a gaussian mask, which is usually different from ω.

It was proposed to filter the derivatives by a Gaussian function:

$$\omega(u,v) = e^{-\frac{u^2+v^2}{2\sigma^2}}$$

$E(x,y)$ can be written as $(x,y)M(x,y)^T$, where M is a symmetric 2×2 matrix:

$$M = \begin{pmatrix} A & B \\ B & C \end{pmatrix}$$

Suppose the eigenvalues of M are α_i $i = 1,2$. Consider three cases:

- If both α_i are small, whatever the displacement is, the E is small. The region is uniform in this case.
- If one of α_i is large, and the other small, the displacement in the direction corresponding to the large eigenvalue will result in a large E, which indicates an edge point.
- If both of them are large, whatever the displacement, E is large, which indicates an interest point.

We use the Harris operator to avoid the explicit computation of α_i and M, We use $Trace(M) = \alpha_1 + \alpha_2 = A + C$ and $Det(M) = \alpha_1\alpha_2 = AC - B^2$.

Since $R = \alpha_1\alpha_2 - k(\alpha_1 + \alpha_2)^2$, we have the following cases:

- If R is close to 0, we are in a uniform region.
- A large R indicates an interest point.
- A large negative R indicates an edge.

We are interested in interest points, so in large R.
The proposed method is the following.

- First,

  $$R = Det(M) - kTrace(M)^2 = \alpha_1\alpha_2 - k(\alpha_1 + \alpha_2)^2$$

  is computed for each pixel.

  k is a variable to be determined experimentally (0.06 in our experience).

- Then,

  we find the global maximum of R over the whole image and note it R_{max}, we compare each R with θR_{max} and keep those larger than θR_{max}, where θ is a preset threshold.

- We assume interest points are isolate (the neighboring points of an interest point is not an interest point).

  We search for the local maximum of R in a neighborhood of 3×3 for each pixel and if the latter is superior to θR_{max}, we consider the corresponding point to be an interest point.

5.1.1 Experimental results

The results of Harris detector are displayed in this section for three pairs of images, using two different thresholds θ: 0.01 and 0.001 (Figure 5.1, 5.2, 5.3, 5.4, 5.5, 5.6). Interest points are marked by white or black squares, depending on the surrounding intensities, to maximize visibility). We see in the images that with lower threshold (0.001), more interest points are obtained.

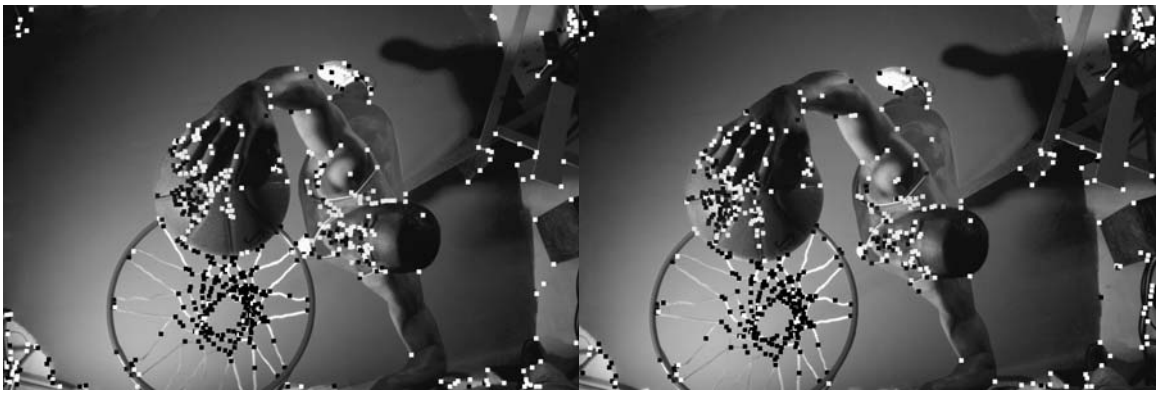


Figure 5.1: Interest points for Basketball: $\theta = 0.001$.

Figure 5.2: Interest points for Basketball: $\theta = 0.01$.

5.2 Establishing correspondence between interest points

Given an interest point $m_1 = (x_1, y_1)$ in image I_1, we use a correlation window of size $(2N_x + 1) \times (2N_y + 1)$ centered at this point. We then select a rectangular search area (Figure 5.7):

Figure 5.3: Interest points for Piano: $\theta = 0.001$.

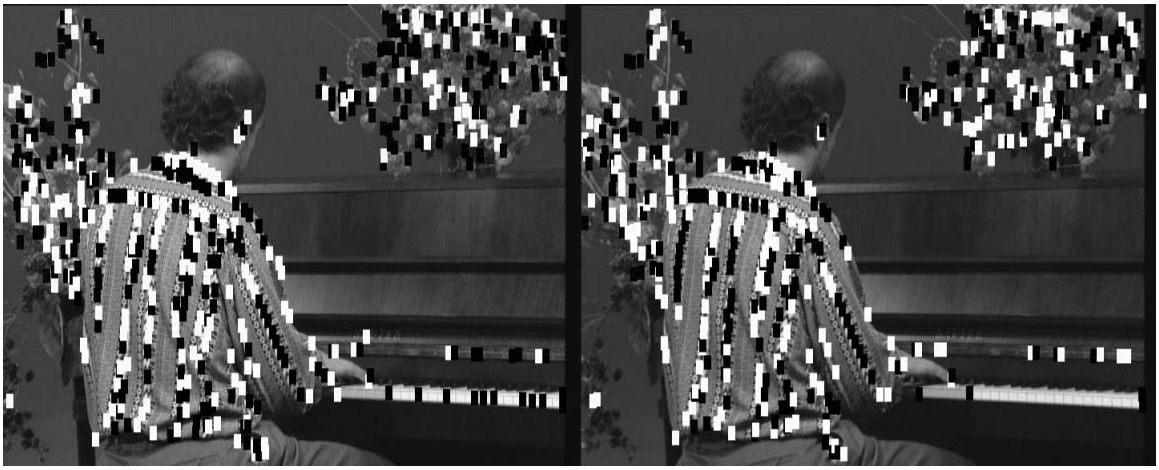


Figure 5.4: Interest points for Piano: $\theta = 0.01$.

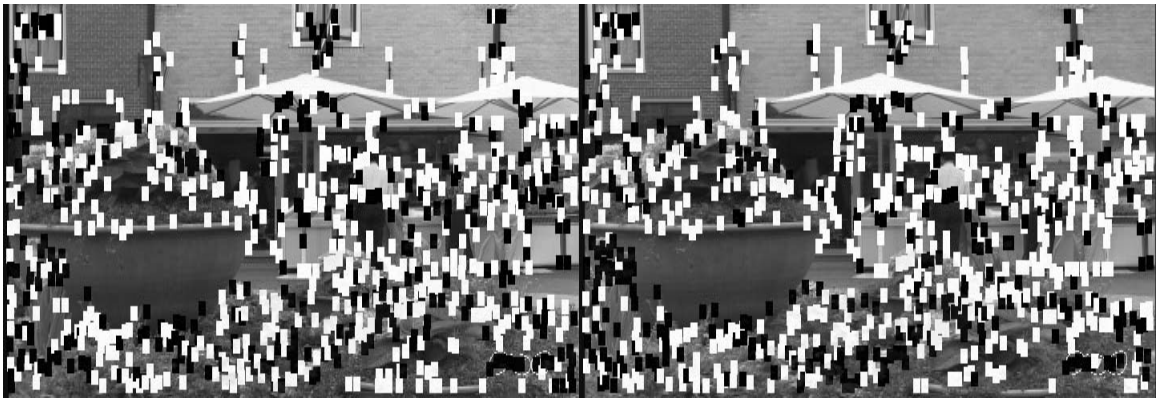


Figure 5.5: Interest points for Flower: $\theta = 0.001$.

Figure 5.6: Interest points for Flower: $\theta = 0.01$.

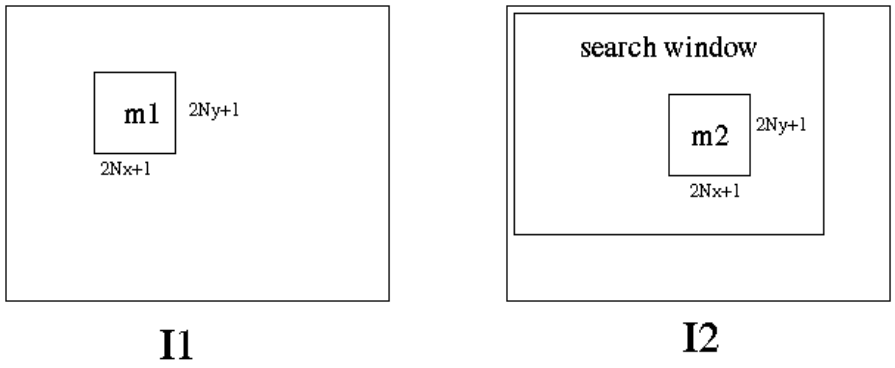


Figure 5.7: Matching interest points.

$$\mathcal{S} = \{(d_1, d_2) | d_1^{min} \leq d_1 \leq d_1^{max}, d_2^{min} \leq d_2 \leq d_2^{max}\}$$

where $d_1^{min}, d_1^{max}, d_2^{min}, d_2^{max}$ are given as disparity limits.

We then perform a correlation operation on a given window between point $m_1 = (x_1, y_1)$ in I_1 and all candidate interest points $m_2 = (x_2, y_2)$ within the search area $\mathcal{S}$ in I_2. This is equivalent to reducing the search area for a corresponding point from all interest points in the whole image to those in search area $\mathcal{S}$. The use of search area also eliminates false matches outside of the area.

We use the following correlation measure [15]:

$$\Psi(m_1, m_2) = \frac{\sum_{i=-N_x}^{i=N_x} \sum_{j=-N_y}^{j=N_y} [I_1(x_1 + i, y_1 + j) - \mu_1(x, y)][I_2(x_2 + i, y_2 + j) - \mu_2(x, y)]}{(2N_x + 1)(2N_y + 1)\sqrt{\sigma_1^2(x, y)\sigma_2^2(x, y)}}$$

where

$$\mu_p(x, y) = \sum_{i=-N_x}^{i=N_x} \sum_{j=-N_y}^{j=N_y} I_p(x + i, y + j)/(2N_x + 1)(2N_y + 1)$$

is the average at point (x, y) of $I_p (p = 1, 2)$ and $\sigma_p(x, y)$ is the standard deviation of the image I_p in the correlation window

$$\mathcal{W} = \{(s, t) \in Z^2 | \, |s - x| \leq N_x, |t - y| \leq N_y\}$$

centered at (x, y), given by:

$$\sigma_p(x, y) = \sqrt{\frac{\sum_{i=-N_x}^{i=N_x} \sum_{j=-N_y}^{j=N_y} I_p^2(x + i, y + j)}{(2N_x + 1)(2N_y + 1)} - (\mu_p(x, y))^2}$$

The score ranges from -1 for two windows which are not similar at all, to 1, for two windows that are identical.

A constraint on the correlation measure is applied in order to eliminate false matches having a correlation score under a threshold. Also, a cross-validation is used to eliminate the false matches, i.e., in order that m_2 in image 2 correspond to m_1 in image 1, m_2 should be the point having the highest correlation score with m_1 among all interest points in the search area of m_1, and m_1 should also be the point having the highest correlation score with m_2 among all interest points in the search area of m_2, their correlation score should be superior to a preset threshold K, i.e. $\psi(m_1, m_2) > K$

Overall, three constraints are used for false match elimination:

- Search area defined by d_1^{min}, d_1^{max}, d_2^{min}, d_2^{max}.

- Threshold of the correlation score, if m_1 and m_2 match each other, we should have $\psi(m_1, m_2) > K$.

- Cross validation.

  If $m_1 \in I_1$ and $m_2 \in I_2$ match each other, then $m_1 \in I_1$ has the highest correlation score with $m_2 \in I_2$ among all interest points in the search area of m_2, and $m_2 \in I_2$ has the highest correlation score with $m_1 \in I_1$ among all interest points in the search area of m_1.

Using these three constraints, almost all false matches can be eliminated. If needed, false matches can be removed manually and correct matches can be added. After this step, we can obtain a set of correct matches, which we will use as control points in block matching.

5.3 Regularized block matching under control point constraint

In the previous section, a method to identify interest points with reliable correspondence has been described. Since the remaining interest points are considered reliable, we will use them as control points for the regularized block matching.

If several matched points are presented in one block, we take a statistics (for example the average) of these disparities.

The regularized block matching using control points works as follows:

- Interest point extraction in left and right images

  Harris operator is used, followed by a local maximum selection and thresholding.

- Matching of interest points

  The correlation technique is used with three constraints: disparity limits, correlation measure threshold and cross validation.

- Block matching initialization

  From the matched interested points, we first initialize the disparity at the associated blocks. Disparity averaging is used if multiple interest points in a block.

  We then propagate the initialization from these blocks, as follows,

We check every uninitialized block and initialize those that have at least one initialized neighbor block. We assign the average of the disparities of all its initialized neighbors.

We loop this procedure until all blocks are initialized.

- Regularized block matching with control points

  From the initialization given by the previous initialization step, we use exhaustive search method to perform regularized block matching, with the constraint that the disparity of control points does not change during the iterations.

5.3.1 Experimental results

The results on three pairs of images (Figure 3.1, 3.2 and 3.3) are displayed in this section Figure (5.8, 5.9, 5.10, using different regulrization factor (0.1 and 1) and different sets of control points (one used threshold 0.01 in the extraction of interest points and the other used 0.001).

The block size is 8×8 for every image matching. The disparity field is subsampled by 16×16 for image *Basket* and 8×8 for *Piano* and *Flower*.

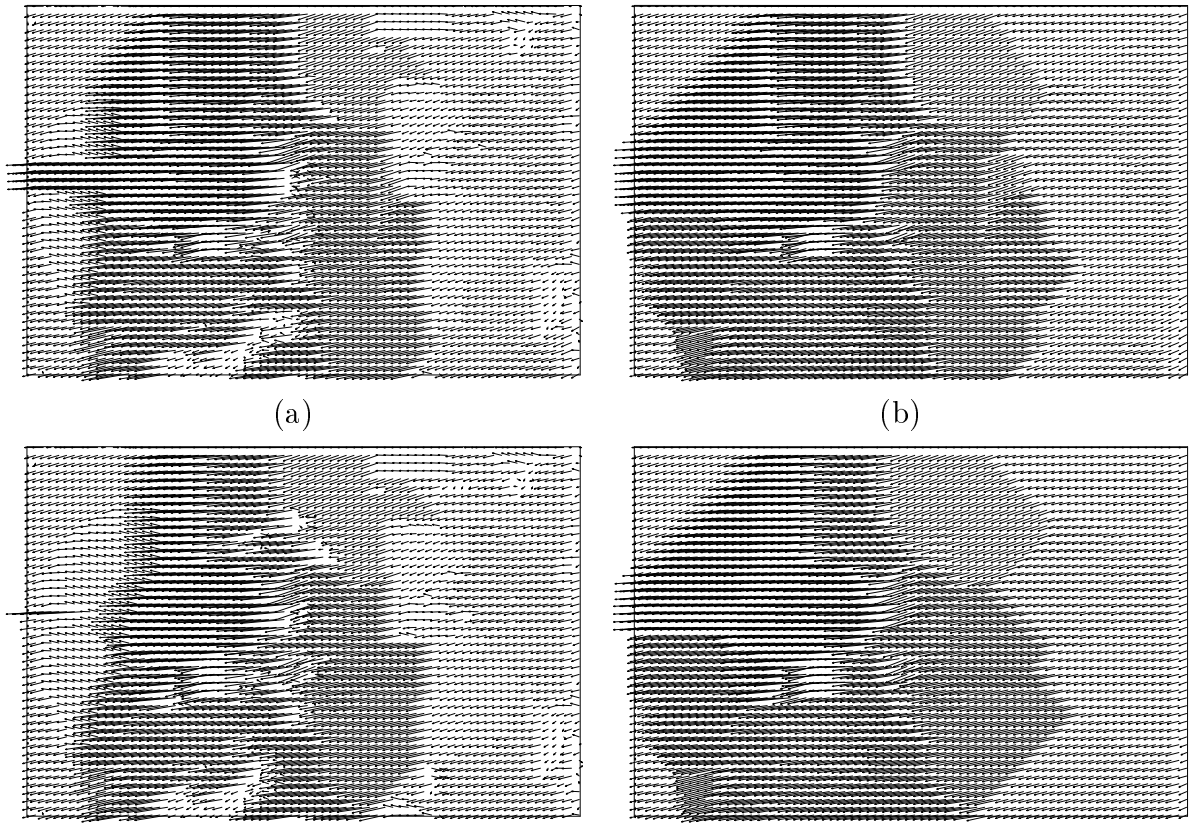


Figure 5.8: Disparity field for *Basket* (a) regularization $\lambda = 0.1$ and $\theta = 0.001$; (b) regularization $\lambda = 1$ and $\theta = 0.001$; (c) regularization $\lambda = 0.1$ and $\theta = 0.01$; (d) regularization $\lambda = 1$ and $\theta = 0.01$.

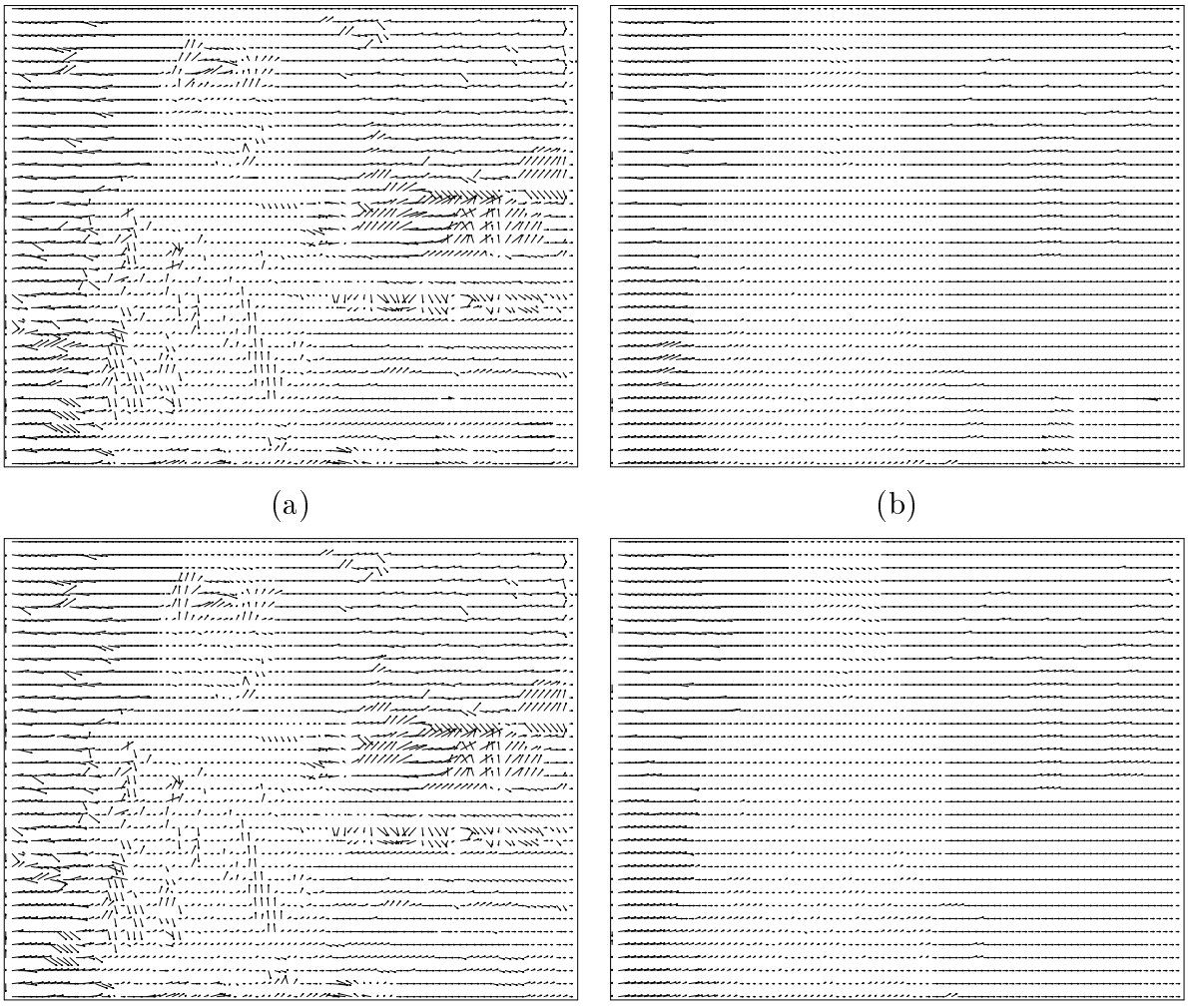


Figure 5.9: Disparity field for *Piano* (a) regularization $\lambda = 0.1$ and $\theta = 0.001$; (b) regularization $\lambda = 1$ and $\theta = 0.001$; (c) regularization $\lambda = 0.1$ and $\theta = 0.01$; (d) regularization $\lambda = 1$ and $\theta = 0.01$.

XS

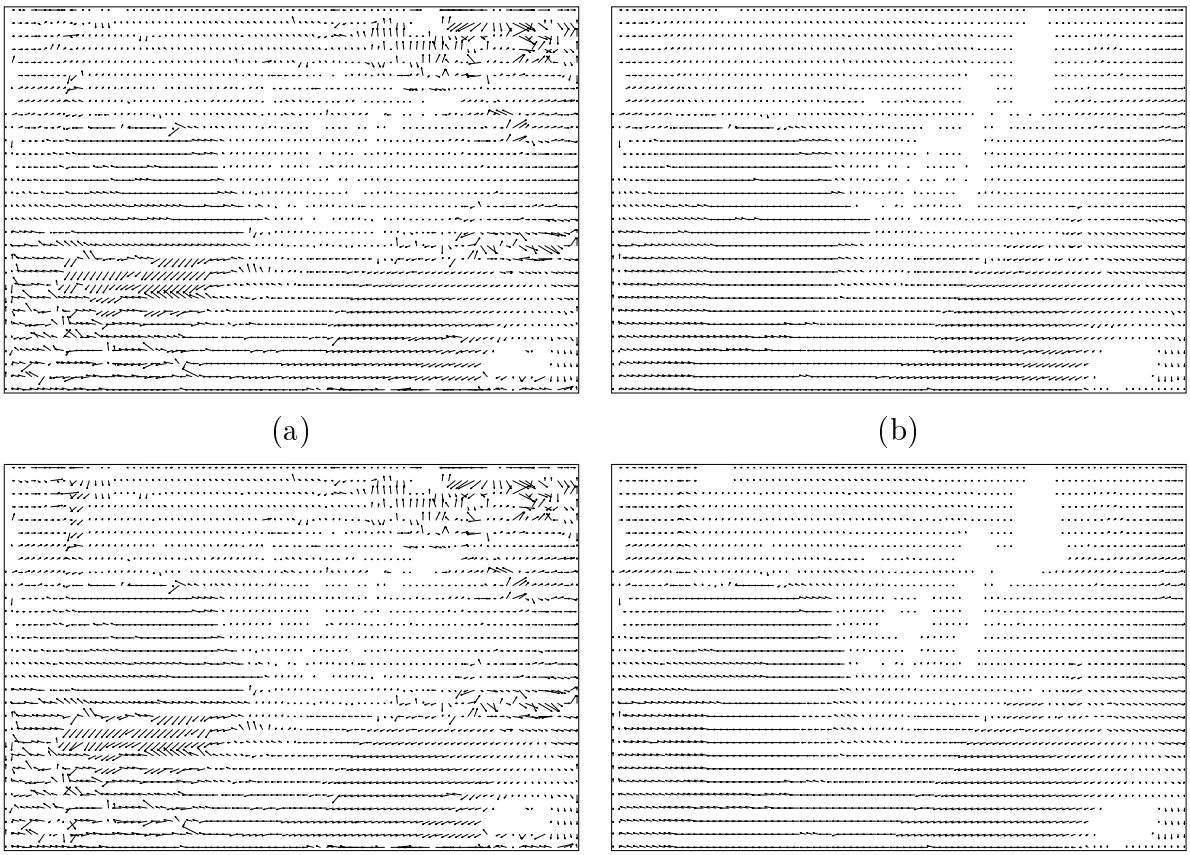


Figure 5.10: Disparity field for *Flower* (a) regularization $\lambda = 0.1$ and $\theta = 0.001$; (b) regularization $\lambda = 1$ and $\theta = 0.001$; (c) regularization $\lambda = 0.1$ and $\theta = 0.01$; (d) regularization $\lambda = 1$ and $\theta = 0.01$.

Chapter 6

Application to intermediate view reconstruction

To reconstruct an intermediate image, vectors from the estimated disparity map for a particular position are indicated as a function of α, where $\alpha = 0$ corresponds to the left image and $\alpha = 1$ corresponds to the right one. Pixel at (i, j) in the intermediate image may be computed using disparity-compensated filtering with a two-coefficient kernel. Here, we use a weighted average of the corresponding picture in the left and right images, according to $d_1(i, j)$ and $d_2(i, j)$ as follows:

$$I_I(i,j) = (1-\alpha)I_L(i-\alpha d_1(i,j), j-\alpha d_2(i,j)) + \alpha I_R(i+(1-\alpha)d_1(i,j), j+(1-\alpha)d_2(i,j)) \tag{6.1}$$

It is easy to verify that when $\alpha = 0$, $I_I(i, j) = I_L(i, j)$ and when $\alpha = 1$, $I_I(i, j) = I_R(i, j)$, also we have: the line passing through $(i + (1 - \alpha)d_1(i, j), j + (1 - \alpha)d_2(i, j))$ and $(i - \alpha d_1(i, j), j - \alpha d_2(i, j))$ passes (i, j) also (Figure 6.1).

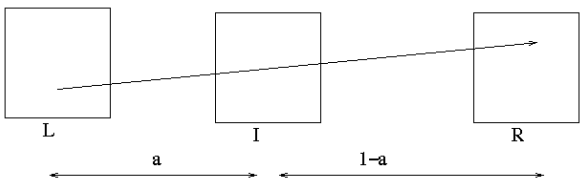


Figure 6.1: Disparity-compensated filtering of disparity vector. Disparity is computed at position (i, j).

Having found matching blocks in I_L and I_R via disparity estimation, in ideal case, the corresponding left and right blocks (to the intermediate image I_I) should have an

identical intensity: that of the intermediate image. In this case, the reconstruction should be simply copying the corresponding left or right image to the intermediate one. However, due to noise, illumination effects and disparity estimation error, the simple copying does not suffice in general. Thus, to reconstruct the I_I at position α, we propose to use disparity-compensated linear interpolation with a simple-coefficient interpolation kernel:

$$I_\alpha(i,j) = \lambda_L I_L(i-\alpha u, j-\alpha v) + \lambda_R I_R(i+(1-\alpha)u, j+(1-\alpha)v)$$

at all positions (i,j) in the intermediate image I_α.

We note $I_L(i-\alpha u, j-\alpha v)$ and $I_R(i+(1-\alpha)u, j+(1-\alpha)v)$ as I_L and I_R for brevity.

Note that normally we should have $\lambda_L + \lambda_R = 1$, which assures a unit gain of the filter.

In general, good results can be obtained with the selection of λ_L and λ_R as follows: $\lambda_L = \frac{|1-\alpha|}{|1-\alpha|+|\alpha|}$ and $\lambda_R = \frac{|\alpha|}{|1-\alpha|+|\alpha|}$.

It can be proved that this choice of coefficients is optimal if we suppose that I_L and I_R all have expectation $\hat{I_\alpha}(i,j)$ and that the ratio of the variance of I_L and that variance of I_R is $|\alpha| \,/\, |1-\alpha|$.

More clearly, suppose the variance of I_L is $\alpha\sigma^2$ and the variance of I_R is $(1-\alpha)\sigma^2$, where σ^2 is the variance of the difference between left and right image. The variance of $\lambda_L I_L + (1-\lambda_L) I_R$ is

$$\lambda_L^2 \,|\alpha|\, \sigma^2 + (1-\lambda_L)^2 \,|1-\alpha|\, \sigma^2$$

It is minimized when

$$\lambda_L = \frac{|1-\alpha|}{|1-\alpha| + |\alpha|}$$

and

$$\lambda_R = 1 - \lambda_L = \frac{|\alpha|}{|1-\alpha| + |\alpha|}$$

6.1 Experimental results

The results of intermediate image reconstruction at $\alpha = 0.5$ for three pairs of images (Figure 3.1, 3.2 and 3.3) are displayed in the following figures (Figure 6.2, 6.3, 6.4) except some small problems for image *basket*, due to very large disparity (up to 100), disparity discontinuity and occlusion, the quality of reconstruction is very satisfactory.

Figure 6.2: Intermediate image for *Basketball*, with regularization and control points.

Figure 6.3: Intermediate image for *Piano*, with regularization and control points.

Figure 6.4: Intermediate image for *Flower*, with regularization and control points.

Chapter 7

Summary and conclusion

This report dealt with the IVR through 2-D signal-based matching. IVR permits adjustment of screen parallax. By reconstruction of intermediate views, the camera separation can be adjusted (increased or reduced), which in turn affects the amount of induced screen parallax on the display. Such a scenario will allow viewers to adjust the "3-D level" of the stereo image, much like typical computer monitors today allow viewers to adjust other parameters such as contrast and brightness.

IVR also plays an interesting role in porting large-screen stereo images to a small one. The large disparities between perspective views that a large-screen display can afford are no longer tolerable on a small screen with smaller viewing distances. IVR can also be used for missing view replacement in a multiview system.

The approach of IVR taken in this report is based on 2-D signal processing techniques. Unlike 3-D model based techniques which typically perform arbitrary view generation for objects only, the approach here makes no assumption about image-content. In order to construct an intermediate view, a mapping between the left and right images is first estimated. This mapping comes from a vector field (called *disparity*) which describes the displacement of each pixel in the right image with respect to its corresponding position in the left image. The process of *disparity estimation* is used to solve this correspondence problem and obtain a disparity field. Then a disparity-compensated interpolation is used to reconstruct the intermediate view. While the interpolation is carried out in the usual manner, the disparity estimation algorithms are novel.

We used regularization in the block matching method in order to get a smooth disparity field, while regularization filters the noise, it also blurs the details, especially at disparity discontinuities. We used *control points* to prevents this problem. The *control points* are those obtained in *sparse matching* of interest points, which are

usually reliable. The use of *control points* prevent the blur of discontinuity, and gives very good detail for the intermediate view reconstruction. It is a type of weak continuity, which is typically used for detail-remained filtering.

Future work can be the use of control points in pixel-based matching, which gives a higher resolution than block-based matching. The occlusion is also a very challenging problem for the future.

Bibliography

[1] H. Asada and M. Brady, "The curvature primal sketch," IEEE *Transactions on Pattern Analysis and Machine Intelligence*, vol. 8, no. 1, pp. 2–14, 1986.

[2] P. Beaudet, "Rotationally invariant image operators," in *Proceedings of the 4th International Joint Conference on Pattern Recognition, Tokyo*, pp. 579–583, 1978.

[3] R. Deriche and T. Blaszka, "Recovering and characterizing image features using an efficient model based approach," in *Proceedings of the Conference on Computer Vision and Pattern Recognition, New York, USA*, pp. 530–535, 1993.

[4] R. d. H. E. F. Heitger, L. Rosenthaler and O.Kuebler, "Simulation of neural contour mechanism: from simple to end-stopped cells," Vision Research, vol. 32, no. 5, pp. 963–981, 1992.

[5] W. Forstner, "A framework for low level feature extraction," in *Proceedings of the 3rd European Conference on Computer Vision, Stockholm, Sweden*, 1994.

[6] C. Harris and M. Stephens, "A combined corner and edge detector," in *Alvey Vision Conference*, pp. 147–151, 1988.

[7] R. Horaud, T. Skordas, and F. Veillon, "Finding geometric and relational structures in an image," in *Proceedings of the 1st European Conference on Computer Vision, Antibes, France*, Lecture Notes in Computer Science, pp. 374–384, Springer-Verlag, April 1990.

[8] L. Kitchen and A. Rosenfeld, "Gray-level corner detection," *Pattern Recognition Letters*, vol. 1, pp. 95–102, 1982.

[9] J. Konrad, "View reconstruction for 3-d video entertainment: Issues, algorithms and applications," in *Proceedings of the* IEEE *International Conference on Image Processing and its applications*, pp. 374–384, July 1999.

[10] G. Medioni and Y. Yasumoto, "Corner detection and curve representation using cubic b-splines," *Computer Vision, Graphics and Image Processing*, vol. 39, no. 1, pp. 267–278, 1987.

[11] F. Mokhtarian and A. Mackworth, "Scale-based description and recognition of planar curves and two-dimensional shapes," IEEE *Transactions on Pattern Analysis and Machine Intelligence*, vol. 8, no. 1, pp. 34–43, 1986.

[12] H. Moravec, "Towards automatic visual obstacle avoidance," in *Proceedings of the 5th International Joint Conference on Artificial Intelligence, Cambridge, Massachusetts, USA*, p. 584, 1977.

[13] K. Rohr, "Recognition corners by parametric fitting," *International Journal of Computer Vision*, vol. 9, no. 3, pp. 213–230, 1992.

[14] C. Schmid, R. Mohr, and C. Bauckhage, "Comparing and evaluating interest points," 1997.

[15] Z. Zhang, R. Deriche, O. Faugeras, and Q. Luong, "A robust technique for matching two uncalibrated images through the recovery of the unknown epipolar geometry," Rapport de recherche 2273, INRIA, May 1994.