Implementation of the phase correlation algorithm: motion estimation in the frequency domain

Citation for published version (APA):

Document status and date:
Published: 01/01/2003

Document Version:
Publisher’s PDF, also known as Version of Record (includes final page, issue and volume numbers)

Please check the document version of this publication:
• A submitted manuscript is the version of the article upon submission and before peer-review. There can be important differences between the submitted version and the official published version of record. People interested in the research are advised to contact the author for the final version of the publication, or visit the DOI to the publisher's website.
• The final author version and the galley proof are versions of the publication after peer review.
• The final published version features the final layout of the paper including the volume, issue and page numbers.

Link to publication

General rights
Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.
• Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
• You may not further distribute the material or use it for any profit-making activity or commercial gain
• You may freely distribute the URL identifying the publication in the public portal.

If the publication is distributed under the terms of Article 25fa of the Dutch Copyright Act, indicated by the “Taverne” license above, please follow below link for the End User Agreement:
www.tue.nl/taverne

Take down policy
If you believe that this document breaches copyright please contact us at:
openaccess@tue.nl
providing details and we will investigate your claim.

Download date: 05. May. 2021
IMPLEMENTATION OF THE PHASE CORRELATION ALGORITHM.
- Motion estimation in the frequency domain -.

M.H.G. Peeters

Supervisor:  prof.dr.ir. G. de Haan
Date:  June 2003
Abstract

In a practical training for the Faculty of Electrical Engineering at Eindhoven University of Technology, a literature search to the past developments concerning the Phase Correlation algorithm was carried out and the algorithm was implemented. The Phase Correlation algorithm is a method for measuring the motion in a video sequence, based on the Fourier shift theorem. It was found that an enhanced implementation of the Phase Correlation algorithm gives very good results, comparable to the results of 3-D Recursive Search (a popular method for motion estimation in consumer electronics equipment). The estimated motion vectors have a close relation to the true motion in a scene, which makes this algorithm very suitable for video format conversion. It is concluded that Phase Correlation and 3-D Recursive Search, together with object based motion estimation, are the best methods for motion estimation currently available.
Contents

1 Introduction 5

2 Overview of Phase Correlation 7
   2.1 Basic principles of Phase Correlation 7
   2.2 Measuring the motion of multiple objects 8
   2.3 Windowing of the phase correlation blocks 9
   2.4 Interpolation of the correlation surface 9
      2.4.1 Sinc-interpolation (in the frequency domain) 9
      2.4.2 Fitting a polynomial curve 11
      2.4.3 Vector refinement using block matching 11

3 Developments in Phase Correlation 13
   3.1 Reduction of the computational complexity 13
   3.2 Motion measurement for multiple objects 14
   3.3 Extension to translation, rotation and scaling 14
   3.4 Further developments 15

4 Implementation of Phase Correlation 17
   4.1 Stage 1: Phase correlation 17
      4.1.1 Acquiring the input blocks 17
      4.1.2 Calculating the phase correlation surface 18
Chapter 1

Introduction

The book for the course 5P530 “Video Processing for multimedia systems” [12] at Eindhoven University of Technology comes with two software tools that allow the students to experiment with the techniques presented in the course. One of these tools demonstrates the results of different motion estimation algorithms. Until now, Phase Correlation, a technique for motion estimation in the frequency domain, was missing in this tool. The main goal of this practical training was to implement the Phase Correlation method for motion estimation for use in this tool. Furthermore, a literature search concerning the developments in Phase Correlation was carried out.

Chapter 2 presents an overview of the Phase Correlation method. Starting from the mathematical theory, the basic principles of Phase Correlation are explained. Some important extensions (e.g. measuring the motion of multiple objects and different methods for achieving a sub-pixel accuracy) are also discussed. The information in this chapter represents the knowledge required to understand (the implementation of) the Phase Correlation algorithm.

In Chapter 3, a summary of the developments in Phase Correlation, found during the literature search, is presented. This chapter gives a clear overview of the past developments and the different trends of the research, like reduction of the computational complexity, motion measurement for multiple objects and extensions to cover translation, rotation and scaling. References to a more detailed description of the developments are included.

An implementation of the Phase Correlation algorithm, which will be used in the tool mentioned above, is discussed in Chapter 4. Each step of both stages (phase correlation and block matching) will be explained in detail, including some important enhancements that do not follow directly from the principles of Phase Correlation.

Measurements and results for the described implementation of the Phase Correlation algorithm are compared with two other popular motion estimation algorithms (Full Search Block Matching and 3-D Recursive Search) and a basic implementation of Phase Correlation in Chapter 5. Beside graphs showing the Modified Mean Squared prediction Error and the smoothness function, snapshots from calculated motion vector fields, displayed in a color overlay, are included for subjective evaluation. All tests are performed on the Renata and the
Car & Gate sequences.

Finally, in Chapter 6, the conclusions from this practical training can be found. Some recommendations for future work concerning the Phase Correlation algorithm are discussed in that chapter too.
Chapter 2

Overview of Phase Correlation

The Phase Correlation method for motion estimation, exploits the property that translation in the spatial domain has its counterpart in the frequency domain. Since the Phase Correlation method uses only the phase information for correlation, the method is relatively insensitive to illumination changes [3]. Consequently, this technique achieves excellent robustness against correlated and frequency-dependent noise.

Section 2.1 of this chapter explains the basic principles of the Phase Correlation method. Departing from the Fourier shift theorem, a two stage method for measuring the motion of multiple objects in a scene is developed in Section 2.2. Section 2.3 emphasizes the importance of applying a smooth window function on the input image blocks. Finally, in Section 2.4, three techniques for achieving a sub-pixel accuracy can be found.

2.1 Basic principles of Phase Correlation

The original Phase Correlation image alignment method [15] was designed for measuring the relative displacement between images. Consider two (infinite) images \(g_1\) and \(g_2\), and let \(g_2\) be a replica of \(g_1\), shifted over \(\vec{d} = (d_x, d_y)\), such that \(g_2(\vec{x}) = g_1(\vec{x} - \vec{d})\). Then, according to the Fourier shift theorem [11], their Fourier transforms are related by:

\[
G_2(\vec{f}) = G_1(\vec{f}) \cdot e^{-j2\pi \vec{f} \cdot \vec{d}}
\]  

(2.1)

Calculating the cross-power spectrum and extracting its phase gives

\[
e^{j\phi(\vec{f})} = \frac{G_1(\vec{f})G_2^*(\vec{f})}{|G_1(\vec{f})G_2^*(\vec{f})|} = \frac{G_1(\vec{f})G_2^*(\vec{f})}{|G_1(\vec{f})G_2^*(\vec{f})|} \cdot e^{-j2\pi \vec{f} \cdot \vec{d}} = e^{-j2\pi \vec{f} \cdot \vec{d}}
\]  

(2.2)

from which it follows that the phase correlation function, \(p(\vec{x})\), is a delta-function located at the point of registration [15], [17]:

\[
p(\vec{x}) = \mathcal{F}^{-1}\{e^{j\phi(\vec{f})}\} = \mathcal{F}^{-1}\{e^{-j2\pi \vec{f} \cdot \vec{d}}\} = \delta(\vec{x} - \vec{d})
\]  

(2.3)

7
CHAPTER 2. OVERVIEW OF PHASE CORRELATION

In the practical case of finite images, even when the overlap between the images is small, the delta-function is replaced by a correlation surface, which is characterized by a narrow peak at the point of registration and lower amplitude peaks at other locations. The resulting correlation measure is relatively scene-independent and not confused by brightness changes in the scene, or noise [15]. Large shifts can be measured with a high accuracy.

2.2 Measuring the motion of multiple objects

The Phase Correlation method as described above can only be used for measuring global motion, but a two stage method has been proposed to be able to measure the movements of multiple objects in a scene [20]. First, on fairly large blocks, a limited number of candidate vectors are calculated using a phase correlation step; then, for smaller blocks, one of these candidate vectors is selected using a block matching criterion.

In the first stage [20], the input picture is divided into fairly large blocks, typically 64 pixels square. The dimensions of these blocks must be at least twice the size of the largest movement that can be measured, to make sure there is enough overlap between the image material in corresponding blocks. The phase correlation function (Equation 2.3) is calculated for corresponding blocks in two successive images, resulting in a correlation surface for each block in the image. Each correlation surface is searched for a small number of dominant peaks, which are the candidate vectors for the second stage. The highest peaks in the correlation surface are the result of the motion of objects in the block, and the relative height of the peaks reflects the relative size of the objects. Candidate vectors with a sub-pixel accuracy can be estimated by interpolating the correlation surface (see Section 2.4).

In the second stage, each of the large blocks is divided into smaller blocks, typically 8 pixels square. The small blocks from the current image are shifted over each candidate vector from the phase correlation step, and the match error with the previous image is calculated. The vector with the lowest match error is assigned to each block. A simple, and popular, match error function is the \textit{Summed Absolute Difference} (SAD) criterion [12]:

\[
\epsilon(\vec{x}, \vec{d}) = SAD(\vec{x}, \vec{d}) = \sum_{\vec{x} \in B(\vec{x})} |g_2(\vec{x}) - g_1(\vec{x} - \vec{d})| \tag{2.4}
\]

where \(\epsilon(\vec{x}, \vec{d})\) is the match error, \(\vec{x}\) is the center of the current small block, \(\vec{d}\) is the displacement vector under evaluation, \(B(\vec{x})\) is the small block centered around \(\vec{x}\) and \(g_2(\vec{x})\), resp. \(g_1(\vec{x})\), are the luminance values at position \(\vec{x}\) in the current, resp. previous, image. To improve the result, candidate vectors from surrounding blocks might be included.

Thus, the Phase Correlation method is similar to the full search block matching algorithm, except that the number of candidate displacements is limited to those resulting from the phase correlation process. This allows the number of candidate vectors to be kept low, while still enabling large displacements to be measured accurately [20].
2.3 Windowing of the phase correlation blocks

The easiest way to obtain the blocks from the input images, is by applying a rectangular window, i.e. the image is simply cut off at the edges of the block. However, due to the periodicity of the Fourier transform of a sampled signal, the left and right, resp. top and bottom, edges of the block effectively join each other, so there will usually be sharp luminance transitions at these edges. As the edges of the blocks are at the same location in both input images, this will increase the height of a peak at zero displacement, or cause a spurious peak. If the correlation surface is interpolated to get a sub-pixel accuracy (see Section 2.4), this causes small movements to be underestimated.

This problem can be solved by using a smoother window, e.g. a Hanning window, which fades to mid-grey at the edges of the window. The top-left graph in Figure 2.1 on page 10 shows an example input signal, consisting of only one frequency. Before calculating the 64-point DFT (Discrete Fourier Transform), the signal is filtered with a rectangular window (middle row of graphs) and a Hanning window (bottom row of graphs). The graphs on the left show the shape of the filter (solid line) and the filtered signal (dashed line). The graphs on the right show the amplitude of the 64-point DFT in decibels. As can be seen from these graphs, the relative height of the peaks at the frequency of the input signal is much larger in case of the smoother window filter.

The noise on the correlation surface caused by revealed and obscured background during camera pans, is reduced too, because new picture material that appears at the edges, contributes less to the correlation process [20]. Overlapping blocks have been proposed, to compensate for the loss of detail near the edges of the blocks.

2.4 Interpolation of the correlation surface

Even though digital images are represented by pixels, the displacement in a scene is not limited to an integer number of pixels. Being able to measure the motion vectors for a scene with a sub-pixel accuracy, is especially useful for the very demanding de-interlacing [12].

In the literature, three methods have been described for measuring the fractional part of the motion vectors, which will be described in the three subsections below. The first two methods to be described (Subsections 2.4.1 and 2.4.2), interpolate the candidate vectors to sub-pixel accuracy in the phase correlation step, whereas the third method (Subsection 2.4.3) refines the motion vectors in the block matching step.

2.4.1 Sinc-interpolation (in the frequency domain)

A method which is easy to implement, but computationally expensive, is to interpolate the correlation surface by performing the inverse Fourier transform on a larger array [2], [8]. If, for example, the inverse Fourier transform is performed on a 256 × 256 array, which is formed
from a 64 x 64 array by zero filling, the candidate vectors in the correlation surface can be found to an accuracy of a quarter pixel. In general, an accuracy of 1/n pixels can be achieved by performing the inverse Fourier transform on an array extended to n times its original size horizontally and vertically (by adding zeroes). Consequently, the horizontal and vertical parts of all candidate vectors from this larger correlation surface are n times as large as without interpolation.

An efficient way to find the highest peak in the interpolated correlation surface, is to take the highest integer vectors from the low resolution (not interpolated) correlation surface, and use them as starting points for a two dimensional 'hill-climbing' search on the high resolution surface [2]. The phase correlation values of the eight points surrounding one of these starting points are examined and the point with the highest value is selected as the new starting point. This process is repeated until the center point of a square of nine pixels has a higher value than its eight neighbors.

A major problem with this approach is that it is computationally expensive, as the inverse Fourier transform is performed on n² (see above) times as many points for an accuracy of 1/n pixels. This means the computational complexity increases quadratically with the required
2.4. **INTERPOLATION OF THE CORRELATION SURFACE**

accuracy.

### 2.4.2 Fitting a polynomial curve

The method used in the software implementation (see Chapter 4) fits a polynomial (in this case a quadratic) curve to the original correlation surface. First, the correlation surface is searched for a limited number of dominant peaks (at integer locations). Then, a polynomial curve is fit to such a peak and a number of surrounding points and the maximum of this curve is selected as the candidate vector. It is advantageous to use a quadratic curve for this purpose rather than a higher order polynomial, because the maximum of a quadratic curve can be found uniquely and explicitly [20]. One quadratic curve can be fit to an integer peak and the points just to its left and right to find the \( x \)-coordinate of the displacement vector, and another one can be fit to the peak and the points just above and below to find the \( y \)-coordinate.

An accuracy better than a tenth of a pixel has been achieved by combining these first two methods [20]. Additional points at half-pixel locations are interpolated by performing the inverse Fourier transform on a \( 2N \times 2N \) array (where the original array is \( N \) pixels square). For each of the integer peaks, one quadratic curve is fitted to the integer peak and the interpolated points just to its left and right, and another one is fitted to the integer peak and the interpolated points just above and below.

### 2.4.3 Vector refinement using block matching

In this last method, the integer candidate vectors are refined to sub-pixel accuracy in the block matching step [10]. The search procedure first considers all candidate vectors \( \pm \frac{1}{2} \) horizontally and \( \pm \frac{1}{4} \) vertically. Then, starting from the half-pixel displacement with the lowest match error (equation 2.4), all quarter-pixel displacements are compared. The procedure is repeated with \( 2^{-n} \)-pixel displacements, until the desired accuracy is obtained.

It is clear that, for all three methods described above, samples at non-integer locations are required for calculating the error function (equation 2.4) in the block matching step. These samples can be computed using one of the many methods for spatial scaling. As these interpolation techniques are not the subject of this study, they will not be discussed in detail. The software implementation (Chapter 4) applies a straightforward bi-linear interpolation.
Chapter 3

Developments in Phase Correlation

The original Phase Correlation image alignment method (see Section 2.1) was invented by Kuglin and Hines from the Lockheed Palo Alto Research Laboratory in California, who presented the method in 1975 [15]. Two years later, a digital processor was designed and built with the help of Pearson and Golosman [17], to implement this technique at a rate of 30 correlations per second on 128 x 128 element images digitized to eight bits. The processor was built with conventional components, employed only a moderate amount of parallelism and used floating point arithmetic.

3.1 Reduction of the computational complexity

To reduce the computational complexity of the Phase Correlation method, Morandi, Piazza and Capancioni replaced the two-dimensional \( N \times N \) Fourier transforms by one-dimensional transforms of length \( N^2 \), which was done by rearranging the input matrices, in 1986 [16]. It was verified in a number of experiments that this method and the original Phase Correlation image alignment method are perfectly equivalent, and yield similar results.

In the same year, Alliney and Morandi proposed a method for digital image registration using projections [1], with the potential of reducing the computational complexity of the required transformations from \( O(N^2 \log N^2) \) to \( O(N \log N) \). They aimed at replacing the two-dimensional Fourier transformations with one-dimensional Fourier transforms. The horizontal displacement, \( d_x \), may be determined by locating the peak of the 1-D inverse Fourier transform

\[
p(x) = \mathcal{F}^{-1}\left\{ \frac{G_{1y}(f_x)G_{2y}^*(f_z)}{|G_{1y}(f_z)G_{2y}(f_z)|} \right\} = \mathcal{F}^{-1}\{e^{-j2\pi f_x d_x}\} = \delta(x - d_x)
\]

(3.1)

where \( G_y(f_x) = \mathcal{F}\{g_y(x)\} \) and

\[
g_y(x) = \int_{-\infty}^{+\infty} g(x, y) dy
\]
A similar procedure is used to determine $d_y$. In the practical case of finite images, it is necessary to window the raw data with a weighting function to get satisfactory results.

### 3.2 Motion measurement for multiple objects

Probably one of the greatest improvements on the Phase Correlation algorithm, i.e. the possibility to measure the motion for multiple objects in a scene, was proposed by Thomas, from the BBC Research Department, in 1987 [20]. A patent for this two stage method was granted in Europe (EP0261137B1) and the United States (US4890160). A description can be found in Section 2.2, and will not be repeated here. In the same year, Belloeil, from the Philips Research Laboratories, did a number of experiments with an algorithm for estimating the motion of multiple objects in a scene too [2], which differed only slightly from Thomas' algorithm. No smooth window filter was applied on the blocks of the input images (see Section 2.3) and a different method for achieving sub-pixel accuracy (Section 2.4.1) was used. The report of this work gives an easy to understand description of the algorithm.

An application of the two stage method described above was investigated by Fernando and Parker, from the Philips Research Laboratories, in 1988 [8]. Their approach for motion compensated field rate conversion, is to calculate the candidate vectors in the studio and send them to the receiver, where the best matching of these is assigned to each pixel of the image. Only a limited number of vectors for any one field are allowed, and a number of these vectors are selected for each region, because of the limited DATV capacity available. For a HDTV picture (1440 x 1152 interlaced) with regions of 32 x 32, with 16 vectors per field and 4 vectors per region, with an accuracy of a quarter pixel, a total data rate of 659.2 kbits/sec is required.

On the third international workshop on signal processing of HDTV in 1989, a number of developments considering Phase Correlation have been presented. Fernando and Parker, together with Rogers, continued on the motion compensated 50 Hz to 100 Hz display up-conversion described above [9]. A real time motion vector measurement hardware design was described by Dabner, from the BBC Research Department [5]. The hardware performed well over a wide range of input material and object velocities and verified the computer simulations of the algorithm. Ziegler, from Siemens Corporate Research and Development, used the Phase Correlation method for hierarchical motion estimation in 140 Mbits/sec HDTV-coding [22]. The effects of using motion estimation in a DCT-hybridcoder (with a given bitrate) were compared through computing the signal to noise ratio.

### 3.3 Extension to translation, rotation and scaling

A generalization of the Phase Correlation method for the registration of translated as well as rotated images was presented by De Castro and Morandi in 1987 [4]. Let $g_2$ be a replica of $g_1$ translated by $\vec{d} = (d_x, d_y)$ and rotated by $\theta$, such that

$$g_2(x, y) = g_1(x \cos \theta + y \sin \theta - d_x, -x \sin \theta + y \cos \theta - d_y)$$
Then, according to the Fourier shift theorem and the Fourier rotation theorem, their transforms are related by:

$$G_2(f_x, f_y) = G_1(f_x \cos \theta + f_y \sin \theta - f_y \sin \theta + f_y \cos \theta) \cdot e^{-j2\pi(f_x d_x + f_y d_y)} \quad (3.2)$$

The procedure consists of determining the angle $\theta$ using a numerical algorithm first, and then evaluating $\vec{d} = (d_x, d_y)$ as in the case of pure translations (see Chapter 2). In general, a rotated (and translated) point will not be at an integer position, so it will be necessary to use some suitable interpolated value. This method was successfully applied in a series of experiments with synthetic and real images of various types.

In 1996, Reddy and Chatterji extended the Phase Correlation image alignment method to cover translation, rotation and scaling [18]. If $g_2$ is a scaled replica of $g_1$ with scale factors $(a, b)$ for the horizontal resp. vertical directions then, according to the Fourier scale property, their Fourier transforms are related by:

$$G_2(f_x, f_y) = \frac{1}{|ab|} G_1(f_x/a, f_y/b) \quad (3.3)$$

By converting the axes to logarithmic scale (and ignoring the factor $1/|ab|$), scaling can be reduced to a translational movement:

$$G_2(\log f_x, \log f_y) = G_1(\log f_x - \log a, \log f_y - \log b) \quad (3.4)$$

Now, the scaling can be found by the phase correlation technique. If $g_2$ is a translated, rotated and scaled replica of $g_1$, with rotation $\theta$ and scale factor $a$ (in horizontal as well as vertical direction), their Fourier magnitude spectra $M(f_x, f_y)$ in polar representation are related by [18]:

$$M_2(\rho, \gamma) = M_1(\rho/a, \gamma - \theta) \quad (3.5)$$

This can be reduced to a translational movement by converting the horizontal axis to logarithmic scale:

$$M_2(\log \rho, \gamma) = M_1(\log \rho - \log a, \gamma - \theta) \quad (3.6)$$

Both the scale factor $a$ and the rotation angle $\theta$ can be found using the phase correlation technique. After the image is scaled and rotated, the translation can be found using the original Phase Correlation method.

Different ways of using the Phase Correlation approach for the estimation of global motion, including translation, rotation and scaling, were compared by Hill and Vlachos in 1999 [14]. Other methods than the one described above could not match its performance, although methods that subsample the image and use a smaller block are much faster. Ertürk and Dennis used this technique for an image sequence stabilization system in 2000 [7]. The system compensates for undesired jitter, while preserving desired global camera motions.

### 3.4 Further developments

In 2001, Hill and Vlachos proposed a shape adaptive Phase Correlation method, which minimizes the influence of the background and simplifies the identification of the dominant peak in
the correlation surface [13]. This was done by adding two steps prior to calculating the Fourier transforms of the input image blocks: First, for a given segmentation map, take an arbitrary area around the object, sufficiently large enough to include the displaced version. Then, replace background pixels by the average intensity of the object. Padding the background with the average prevents that it contributes to the correlation.

At the same time, Sangwine, Eli and Moxey proposed and demonstrated a vector approach to Phase Correlation, based on hypercomplex Fourier transforms [19]. It allows Phase Correlation to be applied to color and other vector images, without making arbitrary choices about correlating luminance, chrominance or separate image components, and thus to apply Phase Correlation to the image as a whole, making maximum use of all the signal information in the image.
Chapter 4

Implementation of Phase Correlation

As was already described in Chapter 2, the Phase Correlation algorithm consists of two stages. In the first stage the candidate vectors for each large block are calculated, and in the second stage one of the candidate vectors is assigned to each small block. The size of the large blocks must be an integer multiple of the size of the small blocks. In this implementation, the large blocks are typically 32 pixels square, with an overlapping border of 16 pixels on each side. This means the phase correlation step is actually performed on blocks of 64 pixels square, as shown in Figure 4.1 on page 18. The small blocks are typically 8 pixels square.

The sections of this chapter describe both stages of the algorithm. The implementation of the phase correlation stage can be found in Section 4.1, whereas Section 4.2 shows the implementation of the block matching stage.

4.1 Stage 1: Phase correlation

Both input images, i.e. the previous and the current frame of the input sequence, are divided into large blocks. For each of the blocks, a limited number of candidate vectors is calculated, as shown in the flow chart of Figure 4.2 (on page 19). The subsections below will explain this first stage of the Phase Correlation algorithm in more detail.

4.1.1 Acquiring the input blocks

For both input images, the luminance values of all pixels inside the current block and its overlapping border are read from the frame buffer, filtered using a smooth window function (see Section 2.3), and stored in a $N \times N$ matrix. The window function used is a separable two-dimensional filter $w[n,m]$, which is created by applying a Hanning window [21] both
Figure 4.1: The blocks of the Phase Correlation algorithm (also see Table 4.1)

horizontally and vertically:

\[ w[m, n] = \left(0.5 - 0.5 \cdot \cos \frac{2\pi m}{N-1}\right) \cdot \left(0.5 - 0.5 \cdot \cos \frac{2\pi n}{N-1}\right) \]  \hspace{1cm} (4.1)

for \(0 \leq m < N\) and \(0 \leq n < N\), where \(N\) is the size of the large block including its overlapping border. For efficiency reasons, the filter coefficients are calculated only once (and stored in a matrix), before the actual algorithm starts. Since the filter is symmetric both horizontally and vertically and since \(w[n, m] = w[m, n]\), only \(\frac{1}{8}N^2\) filter coefficients need to be calculated.

4.1.2 Calculating the phase correlation surface

The two-dimensional DFT (Discrete Fourier Transform) is calculated for both input matrices, using a Split-Radix FFT (Fast Fourier Transform). A two-dimensional DFT can be calculated by first calculating a one-dimensional DFT for all rows of the input matrix, and then calculating a one-dimensional DFT for all columns of the matrix resulting from the first DFT. Because the input matrices are real valued, the Fourier transforms will be symmetric and the output matrix contains \(\frac{1}{2}N^2 + N\) real and \(\frac{1}{2}N^2 + N\) imaginary values.

Now, the phase of the cross-power spectrum (Equation 2.2) is calculated. First, the real and imaginary parts of the cross-power spectrum are calculated from the Fourier transforms
4.1. STAGE 1: PHASE CORRELATION

![Flow chart showing stage 1 of the algorithm](image)

Figure 4.2: Flow chart showing stage 1 of the algorithm

of the input matrices:

\[
\begin{align*}
\mathcal{R}\{CPS[m,n]\} &= \mathcal{R}\{G_1[m,n]\} \cdot \mathcal{R}\{G_2[m,n]\} + \mathcal{I}\{G_1[m,n]\} \cdot \mathcal{I}\{G_2[m,n]\} \\
\mathcal{I}\{CPS[m,n]\} &= \mathcal{I}\{G_1[m,n]\} \cdot \mathcal{R}\{G_2[m,n]\} - \mathcal{R}\{G_1[m,n]\} \cdot \mathcal{I}\{G_2[m,n]\}
\end{align*}
\] (4.2)

(4.3)

for \(0 \leq m \leq \frac{1}{2}N\) and \(0 \leq n < N\), where \(\mathcal{R}\{a\}\), resp. \(\mathcal{I}\{a\}\), denotes the real, resp. imaginary, part of \(a\), \(CPS[m,n]\) is point \([m,n]\) of the cross-power spectrum and \(G_1[m,n]\), resp. \(G_2[m,n]\), is point \([m,n]\) of the Fourier transform of the block from the previous, resp. current, image. Then, both \(\mathcal{R}\{CPS[m,n]\}\) and \(\mathcal{I}\{CPS[m,n]\}\) are divided by the modulus of the cross-power spectrum and again stored in a matrix (also containing \(\frac{1}{2}N^2 + N\) real and \(\frac{1}{2}N^2 + N\) imaginary values). If both \(\mathcal{R}\{CPS[m,n]\}\) and \(\mathcal{I}\{CPS[m,n]\}\) equal zero, division is not possible (due to a zero modulus) and the values stored in the matrix will be zero too.

Finally, the two-dimensional inverse DFT of the matrix containing the phase of the cross-power spectrum is calculated. Because the symmetry did not get lost in the calculations
above, the resulting phase correlation surface will be a real valued $N \times N$ matrix.

### 4.1.3 Finding the candidate vectors

There are two main actions in finding the candidate vectors. The highest peaks with an integer accuracy need to be found from the correlation surface and these peaks need to be refined to a sub-pixel accuracy. A peak, in this case, is defined as a point (at an integer location) in the correlation surface with a higher value than its eight neighbors.

The highest peaks in the correlation surface are stored in a sorted list. The point with the highest phase correlation value is at the top of this list and the length of the list is equal to $L$, i.e. the maximum number of candidate vectors per block (typically 4). The value at each point of the correlation surface is compared with the value at the last position in the list. If the value of the current point is not higher, then $L$ peaks with a higher (or equal) value are already found and the current point can no longer be one of the $L$ highest peaks. Otherwise the value at the current point is compared with the value of its eight neighbors. If one of those is higher, then the current point is not a peak and thus it can't be one of the $L$ highest peaks. If the current point is a peak and it is higher than the peak at the bottom of the list, then the current point is added to the list at the correct (sorted) position and all peaks at lower positions in the list are shifted down by one position, discarding the peak at the bottom of the list.

When the operation described above has been repeated for all points in the correlation surface, a list with the $L$ highest peaks for the current block has been created. Before these peaks are refined to a sub-pixel (in this case quarter-pixel) accuracy, all peaks are compared with the highest peak in the list. To make sure only peaks resulting from object motion are used, all peaks that are not higher than 25% of the highest peak are considered to be noise peaks, and therefore these peaks are removed from the list.

In the final step of this first stage, the highest peaks from the correlation surface are refined to a quarter-pixel accuracy. A quadratic curve is fit to the peak and its two (lower) neighbors, both horizontally and vertically. The location of the maxima of these two curves are an approximation of the horizontal and vertical components of the actual motion vector. If $[x, y]$ is the location of the current peak and $p[x, y]$ is the value of the correlation surface at position $[x, y]$, then the horizontal and vertical components of the candidate vector $\mathbf{d} = (d_x, d_y)$ are given by:

\begin{align*}
d_x &= x + \frac{1}{2} + \frac{p[x, y] - p[x + 1, y]}{p[x - 1, y] - 2p[x, y] + p[x + 1, y]} \quad (4.4) \\
d_y &= y + \frac{1}{2} + \frac{p[x, y] - p[x, y + 1]}{p[x, y - 1] - 2p[x, y] + p[x, y + 1]} \quad (4.5)
\end{align*}

However, due to the periodicity of the phase correlation, the candidate vector $\mathbf{d}$ is not uniquely defined. By assuming that the velocities are limited by $\pm N/2$ horizontally and vertically, this
ambiguity can easily be solved [6]:

\[
    d_x = \begin{cases} 
        d_x & \text{for } d_x < N/2 \\
        d_x - N & \text{for } d_x \geq N/2 
    \end{cases} \quad (4.6)
\]

\[
    d_y = \begin{cases} 
        d_y & \text{for } d_y < N/2 \\
        d_y - N & \text{for } d_y \geq N/2 
    \end{cases} \quad (4.7)
\]

where \( N \) is the size of the (large) blocks once again. In the actual implementation, \( d_x \) and \( d_y \) are multiplied by four and then rounded to the nearest integer, to get an accuracy of a quarter pixel.

4.2 Stage 2: Block matching

In the first stage of the algorithm, a list of candidate vectors for each (large) block has been calculated. In this stage, these large blocks are divided into small blocks (typically 8 pixels square) as shown in Figure 4.1 on page 18. Now, for each small block, all candidate vectors form the sorted list are tried and the SAD (Equation 2.4) is calculated. The vector with the lowest SAD is assigned to the small block as its motion vector. Required luminance values at non-integer locations are calculated by simple bilinear interpolation.

To improve the results of the block matching step, a penalty can be assigned to each candidate vector, depending on its position in the sorted list of candidate vectors. The penalty which is assigned to a vector is given by \( c \cdot i \cdot M^2 \), where \( c \) is a user-defined constant (typically 1, see Section 4.3), \( i \) is the position in the list (with the highest peak at position \( i = 0 \)) and \( M \) is the size of the small blocks. This way, the block matching step is biased towards higher peaks that correspond to more probable motion vectors.

Before the actual block matching step, the candidate vectors of the surrounding (large) blocks can be added to the list of the current large block. This might be useful, because the motion of objects that move into one of the large blocks can not always be estimated correctly otherwise, especially since the parts of a block near its edges contribute less to the phase correlation surface due to the smooth window function (see Section 2.3). To bias the block matching step towards the candidate vectors from the current block, the height of the candidate vectors from neighboring blocks can be reduced to a user-defined percentage (typically 50%, see Section 4.3) before adding them to the sorted list of candidate vectors. Note that the list needs to remain sorted while adding these candidate vectors. The total number of candidate vectors in the list is limited.

4.3 Parameters of this implementation

The performance of this implementation of the Phase Correlation algorithm can be controlled by six parameters. Table 4.1 on the next page describes these parameters and gives their default values. Measurements for different settings of these parameters can be found in
Chapter 5. Two other parameters are the block erosion and vector gain parameters. The block erosion parameter (be) controls a postprocessing step, which leads to a smoother vector field. The vector gain parameter (vg) improves the visibility of vector fields displayed in a color overlay (see Section 5.3). Since the block erosion and vector gain parameters do not affect the two stages of the actual Phase Correlation algorithm, they will not be discussed in this report.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Default</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>pcbs</td>
<td>64</td>
<td><em>Phase correlation block size:</em> The size (i.e. width and height) of the blocks on which the phase correlation step is performed (see Figure 4.1).</td>
</tr>
<tr>
<td>pcss</td>
<td>32</td>
<td><em>Phase correlation step size:</em> The size (i.e. width and height) of the blocks for which a sorted list of candidate vectors is computed (see Figure 4.1).</td>
</tr>
<tr>
<td>bs</td>
<td>8</td>
<td><em>Block size:</em> The size (i.e. width and height) of the small blocks to which one motion vector is assigned in the block matching step (see Figure 4.1).</td>
</tr>
<tr>
<td>ncv</td>
<td>4</td>
<td><em>Number of candidate vectors:</em> The maximum number of candidate vectors in the sorted list of one large block. This is also the maximum number of peaks that will be found in the phase correlation step.</td>
</tr>
<tr>
<td>chp</td>
<td>1</td>
<td><em>Candidate height penalty:</em> Controls the penalty assigned to a candidate vector depending on its position in the list, as explained in Section 4.2 (where the user-defined constant c is controlled by this parameter).</td>
</tr>
<tr>
<td>nch</td>
<td>50</td>
<td><em>Neighbor candidate height:</em> Controls the height of the candidate vectors from neighboring blocks, as explained in Section 4.2 (where the user-defined percentage is controlled by this parameter).</td>
</tr>
</tbody>
</table>

Table 4.1: Parameters for the Phase Correlation algorithm
Chapter 5

Measurements and results

The quality of the motion vectors estimated by the described Phase Correlation algorithm will be evaluated in three ways. First, two measures that are related to the vector prediction quality (Section 5.1) and the vector field consistency (Section 5.2) respectively, are calculated. The results for the Phase Correlation algorithm (as described in Chapter 4) are compared with the results for a basic implementation of the Phase Correlation algorithm and two other popular motion estimation algorithms. As a third evaluation, pictures of the visualized vector fields for both implementations of the Phase Correlation algorithm and the two other motion estimation algorithms are discussed (Section 5.3).

Measurements for the Phase Correlation algorithm in the following sections are performed with the default parameter settings. Measurements for other parameter settings can be found in the graphs of Figure 5.1 and Table A.1 in Appendix A. Sections 5.1 and 5.2 explain these measures. As can be seen from these graphs, the default parameter settings usually give the lowest average $M^2SE$ (see Section 5.1) for the Renata and Car & Gate sequences. A phase correlation block size (pcbs) of 32 pixels gives a lower (and thus better) $M^2SE$, but this gives a very low smoothness figure (see Section 5.2) and limits the velocities that can be measured to $\pm 16$ pixels per frame period horizontally and vertically. Note that the results for the default parameter settings might be different for other test sequences, so these defaults can not be considered the best parameter settings in general.

The basic implementation of the Phase Correlation algorithm is an implementation in accordance with the theory of Chapter 2, but without the enhancements of Chapter 4. All block sizes are the same as for the enhanced implementation and the same number of candidate vectors is calculated for each large block. However, candidate vectors that are lower than 25% of the highest peak are not excluded. Moreover, candidate vectors from neighboring blocks are added to the list without reducing their height (see Section 4.2) and the maximum number of candidate vectors in this list is not limited, which gives a total number of 36 candidate vectors per block. Besides, when assigning the candidate vector with the lowest $SAD$ to each small block, no penalty depending on the position in this list is assigned. Including this implementation in the measurements clearly shows the importance of the enhancements described in Chapter 4.
Figure 5.1: Measurements for the Phase Correlation algorithm
The two other popular motion estimation algorithms that are included in the tests are Full Search Block Matching and 3-D Recursive Search Block Matching. Full Search Block Matching [12] simply tries all possible displacements on every \((8 \times 8\) pixel) block in the image. The displacements are limited by \(\pm 8\) pixels per frame period horizontally and vertically, and only integer displacements are considered. 3-D RS [12] is based on the assumptions that objects are larger than a block and that objects have inertia. The consequence of the first assumption is that the vector describing the velocity of the object in the current block, can be found in at least one of the neighboring blocks. Since not all neighbors can be available yet (due to causality), those vectors that have not yet been calculated in the current image are taken from the previous image, profiting from the second assumption.

All tests are performed on two different sequences, selected to provide critical test material for motion estimation algorithms. The test sequences used are the Renata sequence and the Car & Gate sequence. Snapshots from those sequences can be found in Figure 5.2. The first ten frames of the test sequences are processed with each motion estimation algorithm, but only the last five frames are used for the measurements, to allow the algorithm using temporal prediction (3-D RS) to converge. The shown snapshots are always the tenth frame of a certain sequence.

\[
g_t(\vec{x}) = \frac{1}{2} \left( g_{n-1}(\vec{x} - \vec{d}) + g_{n+1}(\vec{x} + \vec{d}) \right)
\]

Figure 5.2: Snapshots from the used test sequences

5.1 The Modified Mean Squared prediction Error

The Modified Mean Squared prediction Error (M2SE) is such that the resulting figure reflects the quality of the vector/true-motion relation to some extent [12]. The estimated motion vectors are extrapolated one picture period, as this extrapolation is expected to be more legitimate if the motion vectors represent the true velocity of the objects, rather than just a good match between two blocks of pixels. If a motion vector \(\vec{d} = (d_x, d_y)\) is calculated as the displacement between the previous image \(g_{n-1}\) and the current image \(g_n\), then an image \(g_t\) at time \(n\) is interpolated from the previous image \(g_{n-1}\) and the next image \(g_{n+1}\) as their motion compensated average:
Now, the Modified Mean Squared prediction Error can be calculated as:

\[ M2SE(n) = \frac{1}{\|W\|} \sum_{\bar{x} \in W} (g_n(\bar{x}) - g_i(\bar{x}))^2 \]  

(5.2)

where \( W \) is the measurement window, \( \|W\| \) is the number of pixels in \( W \), \( g_n(\bar{x}) \) is the luminance value at location \( \bar{x} \) of the actual image at time \( n \) and \( g_i(\bar{x}) \) is the luminance value at location \( \bar{x} \) of the interpolated image at time \( n \). The measurement window \( W \) equals the entire image, excluding a margin of 32 pixels (the vector range) on all sides. \( M2SE \) can be averaged over a number of frames to get a more reliable measurement.

Figure 5.3 shows measurements of the Modified Mean Squared prediction Error for the methods under evaluation. The values are the average of the results for the Renata sequence and the Car & Gate sequence. As expected, Phase Correlation has a lower Modified Mean Squared prediction Error than Full Search Block Matching. The error measure is about the same as for 3-D Recursive Search. For the basic implementation of the Phase Correlation algorithm, the error measure is much higher than that of all other methods under evaluation.

![Figure 5.3: Modified Mean Squared prediction Error](image)

5.2 The smoothness figure

The smoothness of a vector field is of major importance in video format conversion, because inconsistencies in the vector field could spoil the result dramatically. For each block in the measurement window, the difference between its vector and the vector of each neighboring
5.3. SUBJECTIVE VECTOR FIELD EVALUATION

block is calculated and the sum of these eight differences is calculated. The average over all blocks in the measurement window is calculated and the result is inverted, so the smoothness figure increases if the consistency improves [12]. Again the measurement window equals the entire image, excluding a margin of 32 pixels on all sides. The smoothness figure can also be averaged over a number of frames to get a more reliable measurement.

Figure 5.4 shows measurements of the smoothness figure for the methods under evaluation. Again, the values are the average of the results for the Renata sequence and the Car & Gate sequence. Here too, Phase Correlation scores much better than Full Search Block Matching, but the smoothness figure for Phase Correlation is also much higher than that of the 3-D Recursive Search method. The basic Phase Correlation implementation scores slightly better than Full Search Block Matching on this measure. Note that this performance criterion cannot be judged independently from other performance indicators, because a higher smoothness figure does not always indicate a better performance. For example, if all blocks would have the same motion vector, the smoothness figure would be infinite. However, if two algorithms have a comparable Modified Mean Squared prediction Error, the algorithm with the higher smoothness figure is probably more suited for video format conversion.

<table>
<thead>
<tr>
<th>Method</th>
<th>Smoothness Figure</th>
</tr>
</thead>
<tbody>
<tr>
<td>FSBM</td>
<td>0.82</td>
</tr>
<tr>
<td>3-D RS</td>
<td>4.31</td>
</tr>
<tr>
<td>Basic PC</td>
<td>1.64</td>
</tr>
<tr>
<td>PC</td>
<td>8.61</td>
</tr>
</tbody>
</table>

Figure 5.4: Smoothness figure

5.3 Subjective vector field evaluation

A visual representation of calculated vector fields can be used to improve the confidence in the performance indicators discussed in the previous sections. It also gives a better understanding of the peculiarities of the different methods for motion estimation. Therefore, Figure 5.5 again shows snapshots from both test sequences, but now with a color overlay showing the estimated
motion vectors for each method under evaluation. The hue of the vector overlay represents the orientation of the motion vectors and the saturation represents the length. This relation between the colors in the vector overlays and the estimated motion, can also be found in Figure 5.5. Note that positive velocities are defined as up and to the right. For example, a motion vector of +6 horizontally, means a velocity of 6 pixels per frame period to the right. To improve the visibility of the colors in the vector overlays, the velocities have been limited to ±8 pixels per frame period (both horizontally and vertically). Larger motion vectors were clipped to this value.

Once again, judging on the quality of the generated vector fields, Phase Correlation seems to be better than Full Search Block Matching and just as good as 3-D Recursive Search. The results from the basic Phase Correlation implementation look better than the results from Full Search Block Matching, especially for the Renata sequence.

The low smoothness figure of the Full Search Block Matching method can most clearly be recognized from Figure 5.5 (a). Full Search Block Matching shows a noisy behavior, because it searches for the lowest match error (within its search range) instead of the true motion. Phase Correlation, which calculates a limited number of candidate vectors before the block matching stage, greatly improves the subjective quality of the vector field. When a larger number of candidate vectors is used, such as in the basic implementation, the noisy behavior of the block matching step becomes visible. Since 3-D Recursive Search uses motion vectors present in neighboring blocks most of the time, the resulting vector field looks very smooth too.
5.3. SUBJECTIVE VECTOR FIELD EVALUATION

Figure 5.5: Vector overlays showing the estimated motion vectors
Chapter 6

Conclusions and recommendations

In this practical training, the Phase Correlation algorithm for motion estimation has been implemented with success and an overview of the developments concerning Phase Correlation has been created. The literature search shows that the research in this area was concentrated in the late eighties and early nineties, with the extension to motion measurement for multiple objects by Thomas [20] as the most important improvement.

Beside an overview of the principles of Phase Correlation and a summary of the developments in this area of research, this report presented a software implementation based on the proposals of Thomas [20] and Belloeil [2] from 1987. A comparison was made with Full Search Block Matching, 3-D Recursive Search [12] and a basic implementation of the Phase Correlation algorithm.

Although Phase Correlation is more difficult to implement than Full Search Block Matching and 3-D Recursive Search, the results look clearly better than those of Full Search Block Matching and the measurements in this report even look slightly better than those of 3-D Recursive Search. However, it should be noted that the performance of Phase Correlation and 3-D Recursive Search depends on the used parameter settings and can be different for other test sequences. The enhancements that were introduced in the implementation of the Phase Correlation algorithm clearly improve the results, as can be seen from the comparison with the basic implementation of Phase Correlation.

The computational complexity of Phase Correlation is much lower than that of Full Search Block Matching and when some care is taken for the efficiency of the implementation, it is not that much higher than that of 3-D Recursive Search. The computation time per image depends heavily on the used parameters, such as the overlap between blocks and the number of candidate vectors per block.

Given the comparison between Full Search Block Matching, 3-D Recursive Search and Phase Correlation in this report and the comparison with other motion estimation methods in [12], it can be concluded that 3-D Recursive Search and Phase Correlation, together with object based motion estimation [12], are the best methods for motion estimation currently available. The smooth vector fields corresponding to the true motion in the scene, make these
algorithms very suitable for video format conversion.

One possibility for further research could be to try and find a more general relation between certain properties of an input sequence and the parameter settings that give the best results, for a large amount of test sequences. It seems like the results for one sequence increases when a certain parameter (e.g. the number of candidate vectors) is increased, whereas the results for another sequence decrease when this parameter is increased. It would be advantageous, for applications of the algorithm, if good parameter settings could be determined without actually calculating vector fields.

Some way to improve the quality of vector fields estimated by the Phase Correlation algorithm could be to use a different windowing function (see Sections 2.3 and 4.1.1). The Hanning window is a very smooth window function (shaped as one period of a cosine waveform), but details in the input image are reduced in a relatively large part of the phase correlation blocks. A raised cosine window with a larger flat area in the center of the blocks might improve the candidate vectors.

A final possibility is to combine the principles of 3-D Recursive Search with the principles of Phase Correlation. This could be done in two ways: 3-D Recursive Search could profit from Phase Correlation for updating its candidate vectors and Phase Correlation could profit from 3-D Recursive Search for finding the peaks in the correlation surface. The first idea would raise the computational complexity, but possibly the performance of 3-D Recursive Search could be improved even further. The second idea might even reduce the computational complexity of Phase Correlation and could also improve the quality of the calculated vector fields. More detailed study is required to find out which of the two ways described above is best, and what (if any) improvement could actually be achieved.
Appendix A

Measurements for Phase Correlation

Table A.1 on the next page contains the measurements for the described implementation (Chapter 4) of the Phase Correlation algorithm. The first five columns show the used parameters for each measurement, as explained in Section 4.3. The next four columns columns show the Modified Mean Squared prediction Error \((M2SE, \text{Section 5.1})\) and the smoothness figure (Section 5.2) for the Renata and the Car & Gate sequences. The last two columns show the average M2SE and the average smoothness figure.

The table is divided in six parts. The first five parts each show the influence of changing one parameter, with all other parameters at their default values. Graphs of these measurements can be found in Figure 5.1 on page 24. The last part of the table compares the described implementation of the Phase Correlation algorithm with a basic implementation of the Phase Correlation algorithm and two other popular motion estimation algorithms, as explained in Chapter 5. Graphs of these measurements can be found in Figures 5.3 and 5.4 on pages 26 and 27.
## APPENDIX A. MEASUREMENTS FOR PHASE CORRELATION

<table>
<thead>
<tr>
<th>pcbs</th>
<th>pcss</th>
<th>ncv</th>
<th>chp</th>
<th>nch</th>
<th>Renata M2SE</th>
<th>Smooth</th>
<th>Car &amp; Gate M2SE</th>
<th>Smooth</th>
<th>Average M2SE</th>
<th>Smooth</th>
</tr>
</thead>
<tbody>
<tr>
<td>64</td>
<td>32</td>
<td>4</td>
<td>1</td>
<td>0</td>
<td>27.85</td>
<td>5.94</td>
<td>30.36</td>
<td>3.58</td>
<td>29.11</td>
<td>4.76</td>
</tr>
<tr>
<td>64</td>
<td>32</td>
<td>4</td>
<td>1</td>
<td>25</td>
<td>27.47</td>
<td>7.30</td>
<td>26.98</td>
<td>5.04</td>
<td>27.23</td>
<td>6.17</td>
</tr>
<tr>
<td>64</td>
<td>32</td>
<td>4</td>
<td>1</td>
<td>50</td>
<td>28.93</td>
<td>9.72</td>
<td>21.57</td>
<td>7.50</td>
<td>25.25</td>
<td>8.61</td>
</tr>
<tr>
<td>64</td>
<td>32</td>
<td>4</td>
<td>1</td>
<td>75</td>
<td>32.99</td>
<td>10.11</td>
<td>22.32</td>
<td>7.74</td>
<td>27.66</td>
<td>8.93</td>
</tr>
<tr>
<td>64</td>
<td>32</td>
<td>4</td>
<td>1</td>
<td>100</td>
<td>33.26</td>
<td>10.16</td>
<td>22.25</td>
<td>7.98</td>
<td>27.76</td>
<td>9.07</td>
</tr>
<tr>
<td>64</td>
<td>32</td>
<td>4</td>
<td>0</td>
<td>50</td>
<td>29.06</td>
<td>8.14</td>
<td>21.71</td>
<td>6.45</td>
<td>25.39</td>
<td>7.30</td>
</tr>
<tr>
<td>64</td>
<td>32</td>
<td>4</td>
<td>1</td>
<td>50</td>
<td>28.93</td>
<td>9.72</td>
<td>21.57</td>
<td>7.50</td>
<td>25.25</td>
<td>8.61</td>
</tr>
<tr>
<td>64</td>
<td>32</td>
<td>4</td>
<td>2</td>
<td>50</td>
<td>29.10</td>
<td>9.87</td>
<td>21.97</td>
<td>7.63</td>
<td>25.54</td>
<td>8.75</td>
</tr>
<tr>
<td>64</td>
<td>32</td>
<td>4</td>
<td>4</td>
<td>50</td>
<td>29.56</td>
<td>10.13</td>
<td>22.18</td>
<td>7.83</td>
<td>25.87</td>
<td>8.98</td>
</tr>
<tr>
<td>64</td>
<td>32</td>
<td>4</td>
<td>6</td>
<td>50</td>
<td>30.60</td>
<td>10.19</td>
<td>22.81</td>
<td>7.77</td>
<td>26.71</td>
<td>9.98</td>
</tr>
<tr>
<td>64</td>
<td>32</td>
<td>4</td>
<td>8</td>
<td>50</td>
<td>31.16</td>
<td>10.55</td>
<td>23.66</td>
<td>7.80</td>
<td>27.41</td>
<td>9.18</td>
</tr>
<tr>
<td>64</td>
<td>32</td>
<td>1</td>
<td>1</td>
<td>50</td>
<td>40.50</td>
<td>11.46</td>
<td>30.05</td>
<td>9.06</td>
<td>35.28</td>
<td>10.26</td>
</tr>
<tr>
<td>64</td>
<td>32</td>
<td>2</td>
<td>1</td>
<td>50</td>
<td>32.59</td>
<td>10.37</td>
<td>25.53</td>
<td>7.92</td>
<td>29.06</td>
<td>9.15</td>
</tr>
<tr>
<td>64</td>
<td>32</td>
<td>4</td>
<td>1</td>
<td>50</td>
<td>28.93</td>
<td>9.72</td>
<td>21.57</td>
<td>7.50</td>
<td>25.25</td>
<td>8.61</td>
</tr>
<tr>
<td>64</td>
<td>32</td>
<td>8</td>
<td>1</td>
<td>50</td>
<td>25.35</td>
<td>9.23</td>
<td>28.41</td>
<td>5.02</td>
<td>26.90</td>
<td>7.13</td>
</tr>
<tr>
<td>64</td>
<td>32</td>
<td>12</td>
<td>1</td>
<td>50</td>
<td>25.27</td>
<td>8.88</td>
<td>29.00</td>
<td>4.69</td>
<td>27.14</td>
<td>6.79</td>
</tr>
<tr>
<td>64</td>
<td>32</td>
<td>16</td>
<td>1</td>
<td>50</td>
<td>25.30</td>
<td>8.78</td>
<td>29.35</td>
<td>4.59</td>
<td>27.33</td>
<td>6.69</td>
</tr>
<tr>
<td>64</td>
<td>32</td>
<td>4</td>
<td>1</td>
<td>50</td>
<td>28.93</td>
<td>9.72</td>
<td>21.57</td>
<td>7.50</td>
<td>25.25</td>
<td>8.61</td>
</tr>
<tr>
<td>64</td>
<td>40</td>
<td>4</td>
<td>1</td>
<td>50</td>
<td>27.01</td>
<td>10.43</td>
<td>24.48</td>
<td>7.67</td>
<td>25.75</td>
<td>9.05</td>
</tr>
<tr>
<td>64</td>
<td>48</td>
<td>4</td>
<td>1</td>
<td>50</td>
<td>30.06</td>
<td>10.29</td>
<td>26.28</td>
<td>8.45</td>
<td>28.17</td>
<td>9.37</td>
</tr>
<tr>
<td>64</td>
<td>56</td>
<td>4</td>
<td>1</td>
<td>50</td>
<td>29.65</td>
<td>10.88</td>
<td>23.07</td>
<td>8.06</td>
<td>26.36</td>
<td>9.47</td>
</tr>
<tr>
<td>64</td>
<td>64</td>
<td>4</td>
<td>1</td>
<td>50</td>
<td>31.39</td>
<td>9.40</td>
<td>21.96</td>
<td>6.62</td>
<td>26.68</td>
<td>8.01</td>
</tr>
<tr>
<td>16</td>
<td>8</td>
<td>4</td>
<td>1</td>
<td>50</td>
<td>26.76</td>
<td>2.46</td>
<td>22.46</td>
<td>2.95</td>
<td>24.61</td>
<td>2.71</td>
</tr>
<tr>
<td>32</td>
<td>16</td>
<td>4</td>
<td>1</td>
<td>50</td>
<td>25.35</td>
<td>6.02</td>
<td>20.83</td>
<td>4.69</td>
<td>23.09</td>
<td>5.46</td>
</tr>
<tr>
<td>64</td>
<td>32</td>
<td>4</td>
<td>1</td>
<td>50</td>
<td>28.93</td>
<td>9.72</td>
<td>21.57</td>
<td>7.50</td>
<td>25.25</td>
<td>8.61</td>
</tr>
<tr>
<td>128</td>
<td>64</td>
<td>4</td>
<td>1</td>
<td>50</td>
<td>28.11</td>
<td>10.60</td>
<td>25.58</td>
<td>8.29</td>
<td>26.85</td>
<td>9.45</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>Renata M2SE</th>
<th>Smooth</th>
<th>Car &amp; Gate M2SE</th>
<th>Smooth</th>
<th>Average M2SE</th>
<th>Smooth</th>
</tr>
</thead>
<tbody>
<tr>
<td>Full Search Block Matching</td>
<td>30.96</td>
<td>0.64</td>
<td>25.40</td>
<td>0.99</td>
<td>28.18</td>
<td>0.82</td>
</tr>
<tr>
<td>3-D Recursive Search</td>
<td>27.80</td>
<td>4.89</td>
<td>22.81</td>
<td>3.73</td>
<td>25.31</td>
<td>4.31</td>
</tr>
<tr>
<td>Basic Phase Correlation</td>
<td>28.17</td>
<td>1.77</td>
<td>40.64</td>
<td>1.51</td>
<td>34.41</td>
<td>1.64</td>
</tr>
<tr>
<td>64</td>
<td>28.93</td>
<td>9.72</td>
<td>21.57</td>
<td>7.50</td>
<td>25.25</td>
<td>8.61</td>
</tr>
</tbody>
</table>

Table A.1: Measurements for the Phase Correlation algorithm
Bibliography

DIGITAL IMAGE REGISTRATION USING PROJECTIONS.
IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 8, no. 2, March

MOTION ESTIMATION WITH THE PHASE CORRELATION METHOD.
Internal report no. 2595.

A SURVEY OF IMAGE REGISTRATION TECHNIQUES.

REGISTRATION OF TRANSLATED AND ROTATED IMAGES USING FINITE
FOURIER TRANSFORMS.
IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 9, no. 5, Septem­
ber 1987, p. 700–703.

REAL TIME MOTION VECTOR MEASUREMENT HARDWARE.
In: Proc. signal processing of HDTV, II: 3rd international workshop, Turin, Italy, 30

[6] Ernst, M.
NORMWANDLUNG FÜR HDTV
Report no. BMFT FB TK 441.

IMAGE SEQUENCE STABILISATION BASED ON DFT FILTERING.
IEE proceedings — vision, image and signal processing, vol. 147, no. 2, April 2000,
p. 95–102.

[8] Fernando, G.M.X. and D.W. Parker
MOTION COMPENSATED DISPLAY CONVERSION.
[9] Fernando, G.M.X. and D.W. Parker, P.T. Rogers
MOTION COMPENSATED DISPLAY FIELD RATE CONVERSION OF BAND-WIDTH COMPRESSED HD-MAC PICTURES.

[10] Girod, B.
MOTION-COMPELLASING PREDICTION WITH FRACTIONAL-PEL ACCURACY.

INTRODUCTION TO FOURIER OPTICS. 2nd ed.

[12] Haan, G. de
VIDEO PROCESSING FOR MULTIMEDIA SYSTEMS.

MOTION MEASUREMENT USING SHAPE ADAPTIVE PHASE CORRELATION.

[14] Hill, L. and T. Vlachos
ON THE ESTIMATION OF GLOBAL MOTION USING PHASE CORRELATION FOR BROADCAST APPLICATIONS.

THE PHASE CORRELATION IMAGE ALIGNMENT METHOD.

[16] Morandi, C. and F. Piazza, R. Capancioni
ROBUST 1-D EQUIVALENT OF PHASE-CORRELATION IMAGE REGISTRATION ALGORITHM.

VIDEO-RATE IMAGE CORRELATION PROCESSOR.
[18] Reddy, B. and B. Chatterji
AN FFT-BASED TECHNIQUE FOR TRANSLATION, ROTATION AND SCALE-IN Variant IMAGE REGISTRATION.

VECTOR PHASE CORRELATION.

TELEVISION MOTION MEASUREMENT FOR DATV AND OTHER APPLICATIONS.
BBC Research Department, Engineering Division, 1987.
Internal report no. BBC RD 1987/11.

[21] Verkroost, G.
DIGITALE SIGNAALBEWERKING.
Collegedictaat nr. 5625.

[22] Ziegler, M.
HIERARCHICAL MOTION ESTIMATION USING THE PHASE CORRELATION METHOD IN 140 MBIT/S HDTV-CODING.