Using 2-D wavelet analysis for matching two images

Jiri Walder,
jiri.walder@vsb.cz

Technical University of Ostrava
Ostrava / Czech Republic

Abstract

In this article, we deal with the problem of general matching of two images taken by a camera. To find the correspondence between two images, we use the multiresolution wavelet analysis of the image functions. The wavelet analysis is good at scale adaptivity and, therefore, very effective in general image matching. We test various wavelet functions, and evaluate their reliability.

Keywords: image matching, image correspondence, wavelet analysis

1. Two images matching problem

Suppose we have some part of a real or an artificial world. This world we call a scene. Furthermore, we suppose that objects of the scene are static. This means that they don't move, and don't change their shapes in time. Next we have a camera which moves in the scene. In certain short time intervals, the camera takes images of the scene.

Suppose we have two images of one scene each taken from a different position of the camera. The image matching problem refers to a process of establishing the correspondence between each pair of visible homologous image points on a given pair of images. Thus, we work with two two-dimensional (2-D) discrete image functions. In order to find the correspondence, the images have to satisfy some assumptions. Each pair of the images has no less than 50% overlapping, which is defined as the minimum ratio between the scene surface commonly depicted in both images over the surfaces depicted in each one of them. We require at least 60% overlapping, and the vergence angle formed by both image planes to be less than $\pi/2$.

2. Wavelet analysis and image matching

In this chapter, we explain how wavelet decomposition of 2-D image function $f(x, y)$ is to be used for matching two images.

2.1 Continuous 2-D wavelet transform

A continuous 2-D wavelet transform, as a way of image decomposition, is a projection of the image function $f(x, y)$ onto a family of functions which are linear combinations (dilations and translations) of a unique function $\Psi(x, y)$. The function $\Psi(x, y)$ is called a 2-D mother wavelet, and the corresponding family of wavelet functions is given by

$$\Psi_{s,u,v}(x, y) = \sqrt{s}\,\Psi(s(x - u, y - v)), \quad \text{where} \quad s, u, v \in \mathbf{R}, \tag{2.1}$$

and where s and (u, v) are called the scale and the translation. The 2-D wavelet transform of the function $f(x, y) \in \mathbf{L}_2$ is defined by

$$T_\Psi f(s,u,v) = \int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty} f(x,y)\boldsymbol{\Psi}_{s,u,v}(x,y)\mathrm{d}x\mathrm{d}y\,. \qquad (2.2)$$

This transform can be interpreted as a decomposition of an image function $f(x, y)$ such that the frequency spectrum of the transformed function $T_\Psi f(s,u,v)$ has the same bandwidth in a logarithmic scale s. The adaptivity to scale and translation leads to its good locality in both frequency and spatial domain. This property is suitable in our problem of general image matching. Let the function $\boldsymbol{\Psi}$ be a differential operator, then the transform $T_\Psi f(s,u,v)$ will be a continuous scale-space representation of the image edges. If we use low-pass approximation operator $\boldsymbol{\Phi}$ instead of high-pass differential operator $\boldsymbol{\Psi}$, the transform will produce a continuous scale-space representation of the image approximations. From now on, we shall let $\boldsymbol{\Phi}(x, y)$ denote 2-D *scaling function* corresponding to low-pass approximation operator, and $\boldsymbol{\Psi}(x, y)$ the *wavelet function* corresponding to high-pass differential operator. Similarly, we shall use $\phi(x)$ for 1-D scale function and $\psi(x)$ for 1-D wavelet function.

Example of wavelet function and scale function is shown in Fig. 2.1. In the figures, we can also see the amplitude spectrum of the wavelet functions and the scale functions. Notice the low-pass characteristic of the scale function and the high-pass characteristic of the wavelet function.

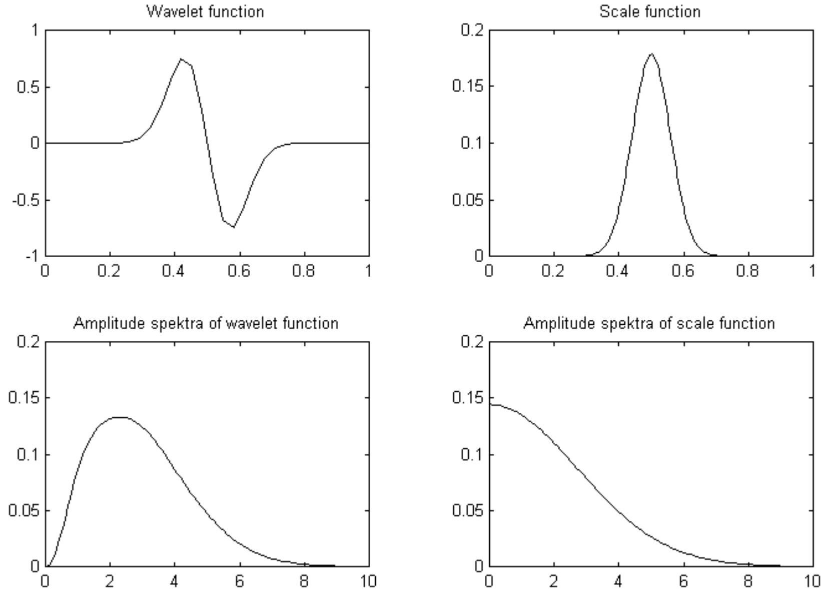

Figure 2.1: Gauss wavelet function ($-2xe^{-x^2}$), scale function (e^{-x^2}) and their amplitude spectra

2.2 Continuous wavelet transform and image matching problem

From previous chapter, we already know that when using the 2-D continuous wavelet transform $T_\Psi f(s,u,v)$ or the 2-D continuous scale transform $T_\Phi f(s,u,v)$, we can decompose the image function $f(x,y)$ on different levels of resolution (depending on the scale s). On the coarsest level, we can easily find singularities (such as edges, corners, peaks etc.). Then we can track them down from a high level (coarse) to the lowest level (detail). In this way we can determine the whole correspondence between two images.

To demonstrate this point I will give you an example. Figs. 2.2 - 2.4 show the maxima of continuous Gaussian wavelet transform of two 1-D image profile functions along their homologous epipolar lines. It can be clearly seen that from the scale 32 downwards, the maxima correspondence can be easily tracked down. Homologous maxima have been determined by a naked eye, and are signed by homologous numbers. We can extend the same idea to 2-D image matching.

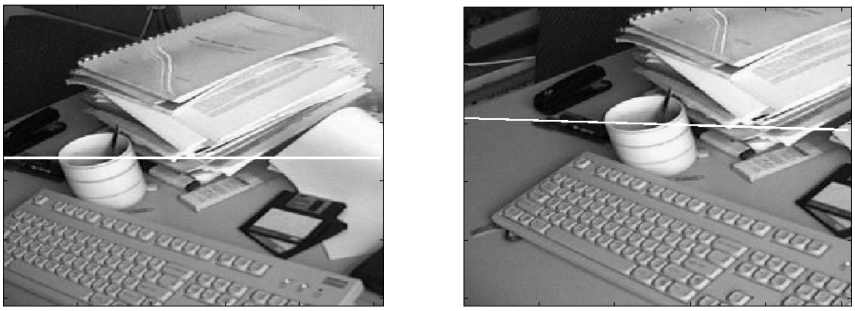

Figure 2.2: Two images of the same scene and two homologous epipolar lines

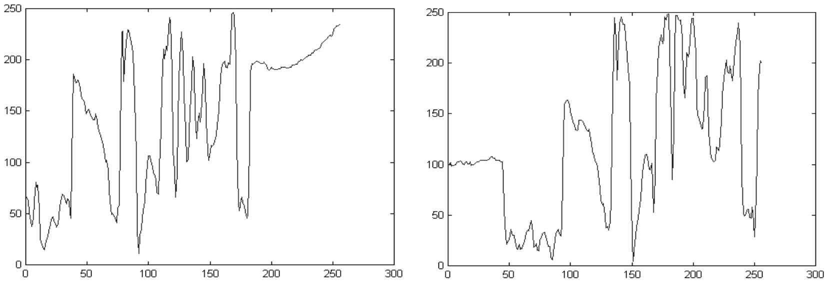

Figure 2.3: 1-D image functions along their homologous epipolar lines

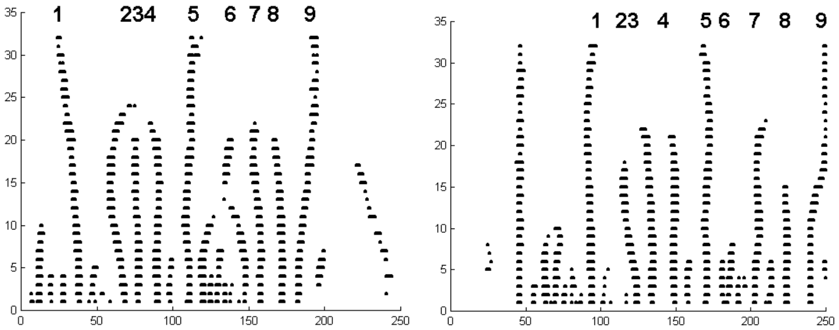

Figure 2.4: Local maxima of the continuous Gaussian wavelet transform

2.3 Discrete dyadic wavelet functions

For the special class of wavelet functions, the redundant information of the continuous wavelet transform can be cleared by discretizing both the scale factor s and the translation (u, v). Strictly speaking, we will work in a dyadic wavelet space

$$s = \frac{1}{2^j} \quad \text{and} \quad (u = k, v = l), \text{ where } j, k, l \in \mathbf{Z}, \tag{2.3}$$

and where **Z** denotes the set of integers. The corresponding set of the dyadic wavelet functions and the scale functions is

$$\psi_m(x) = \sqrt{s}\psi(sx - k), \tag{2.4}$$

where *m* is actually the function of *j* and *k*, as follows:

$$m = 2^j + k, \ k = 0, 1, \ldots, 2^{n-j} - 1 \text{ for } j = 0, 1, \ldots, n$$

Now the coefficients of the wavelet function can be computed according to the formula

$$c_m = \langle f(x), \psi_m(x) \rangle = 2^{j/2} \int_{-\infty}^{\infty} f(x)\psi(2^j x - k)\,\mathrm{d}x, \tag{2.5}$$

where $\langle \cdot, \cdot \rangle$ denotes the scalar product. The corresponding inverse wavelet transform is

$$f(x) = \sum_{m=0}^{\infty} c_m \overline{\psi}_m(x). \tag{2.6}$$

The signification $\bar{\cdot}$ denotes the complex conjugate. Here, a continuous function is being represented by a single infinite sequence, as with a Fourier series representation. We have here, as well, the basis of the discrete wavelet transform if we use discrete versions of Eqs. (2.5) and (2.6).

If we use the scale function $\phi(x)$ instead of the wavelet function $\psi(x)$, we get formulas for dyadic scale functions, and we will work in a scale space.

2.4 Multidimensional analysis of the image function *f*(*x*, *y*)

By multidimensional analysis of the function *f*(*x*, *y*) we understand a sequence of closed scale subspaces $V_n \subset V_{n-1} \subset \ldots \subset V_1 \subset V_0$. We introduce an approximation operator A_j, then $A_j f$ will be an approximation of *f* on the *j*th level of resolution, A_0 represents the identity, $A_j f \in V_j$ holds. The scale *s* on the *j*th level of approximation is $s = \frac{1}{2^j}$. Practically, we use limited number of levels $j = 0, 1, \ldots n$, where the *n*th level will be the coarsest resolution with the smallest scale $s = \frac{1}{2^n}$. Next we introduce a differential operator D_j. The term $D_j f$ will denote the difference between two approximations $A_j f$ and $A_{j-1} f$ on the *j*th and the (*j* – 1)th levels of resolution, i.e.,

$$D_j f = A_{j-1} f - A_j f \quad \text{for} \quad j = 1, 2, \ldots, n. \tag{2.7}$$

Mallat first proved that for 2-D multiresolution analysis, there are three components for the difference between approximations $A_j f$ and $A_{j-1} f$

$$D_j f = A_{j-1} f - A_j f = D_{j,1} f + D_{j,2} f + D_{j,3} f. \tag{2.8}$$

So the multiresolution analysis of the function *f*(*x*, *y*) can be written as

$$\begin{aligned} f(x, y) &= A_1 f + D_{1,1} f + D_{1,2} f + D_{1,3} f \\ &= A_2 f + D_{2,1} f + D_{2,2} f + D_{2,3} f + D_{1,1} f + D_{1,2} f + D_{1,3} f \\ &\ldots \\ &= A_n f + \sum_{j=1}^{n} [D_{j,1} + D_{j,2} + D_{j,3}]. \end{aligned} \tag{2.9}$$

Each approximation $A_j f(x, y)$ and difference $D_{j,p} f(x, y)$ can be fully characterized with a 2-D scale function $\Phi(x, y)$ and its associated wavelet functions $\Psi(x, y)$, $p = 1, 2, 3$,

$$A_j\, f(x,y)=\sum_{k=-\infty}^{+\infty}\ \sum_{l=-\infty}^{+\infty} a_{j,k,l}\mathbf{\Phi}_{j,k,l}(x,y),\quad D_{j,p}\, f(x,y)=\sum_{k=-\infty}^{+\infty}\ \sum_{l=-\infty}^{+\infty} d_{j,k,l,p}\mathbf{\Psi}_{j,k,l}(x,y)\,, \tag{2.10}$$

where

$$\mathbf{\Phi} j,k,l(x,y)=\frac{1}{2^j}\mathbf{\Phi}(\frac{x-k}{2^j},\frac{y-l}{2^j}),\quad \mathbf{\Psi} j,k,l,p(x,y)=\frac{1}{2^j}\mathbf{\Psi}_p(\frac{x-k}{2^j},\frac{y-l}{2^j}),\quad j,k,l\in \mathbf{Z} \tag{2.11}$$

$$a_{j,k,l}=\left\langle f(x,y),\mathbf{\Phi}_{j,k,l}(x,y)\right\rangle,\quad d_{j,k,l,p}=\left\langle f(x,y),\mathbf{\Psi}_{j,k,l,p}(x,y)\right\rangle. \tag{2.12}$$

Suppose that the coordinates x and y are not correlated, and the 2-D scale function $\mathbf{\Phi}(x, y)$ and 2-D wavelet functions $\mathbf{\Psi}(x, y)$, $p = 1, 2, 3$ are separable. Than we can write

$$\mathbf{\Phi}(x,y)=\phi(x)\phi(y),$$

$$\mathbf{\Psi}_1(x,y)=\phi(x)\psi(y),\quad \mathbf{\Psi}_2(x,y)=\psi(x)\phi(y),\quad \mathbf{\Psi}_3(x,y)=\psi(x)\psi(y)\,, \tag{2.13}$$

where $\phi(x)$ is a 1-D scale function and $\psi(x)$ is a 1-D wavelet function. Apparently, $\mathbf{\Psi}_1$, $\mathbf{\Psi}_2$, $\mathbf{\Psi}_3$ extract the details of 2-D image function $f(x, y)$ in the x-axis, y-axis and in the diagonal directions, respectively.

This representation is called the wavelet pyramid of 2-D image function. Given a discrete image $f(x, y)$ with a limited support $x, y = 1, 2, \ldots, 2^n$, the actual procedure for constructing this pyramid involves computing the coefficients $a_{j,k,l}$ and $d_{j,k,l,p}$ which can be grouped into four matrices A_j, $D_{j,\,p}$, $p = 1, 2, 3$, on each level j

$$A_j=(a_{j,k,l}),\qquad D_{j,p}=(a_{j,k,l,p})\qquad \text{for}\qquad k,l=1,\,2,\,\ldots,\,2^{\,n-j}. \tag{2.14}$$

Since we work in a discrete space, we need discrete versions h and g of the functions $\phi(x)$ and $\psi(x)$. The coefficients $a_{j,k,l}$ and $d_{j,k,l,p}$ then can be computed via an iterative procedure. Figure 2.5 illustrates the process of the wavelet analysis. By crossing between levels $j–1$ and j, we first convolve the rows of the image approximation $A_{j–1} f(x, y)$ with the discrete scale function $h(x)$ and the wavelet function $g(x)$. Then we discard the odd-numbered columns (denoted by ↓). We get the approximation and the difference of given function in the x-axis direction. We repeat the whole process for all columns. The approximation $A_j f(x, y)$ and three difference components $D_{j,p} f(x, y)$, $p = 1, 2, 3$ form the result.

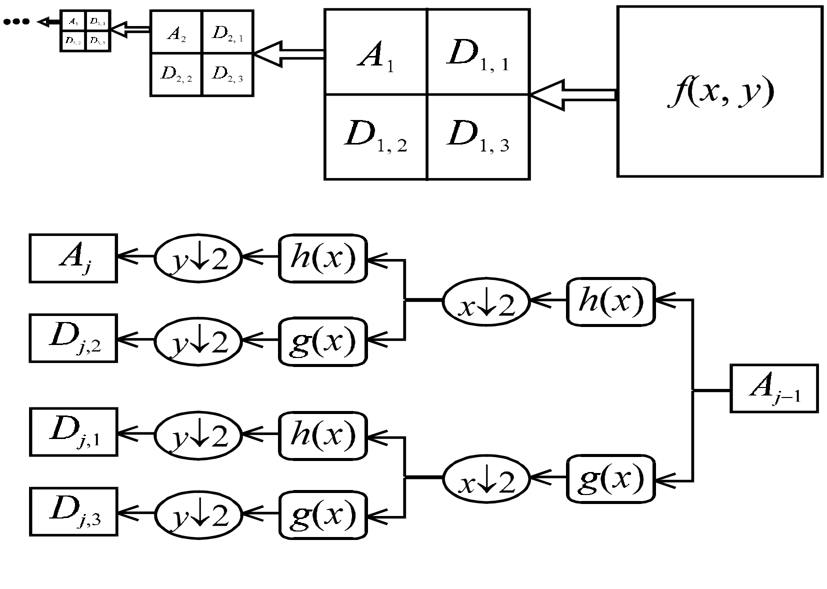

Figure 2.5: Wavelet pyramid and process of wavelet analysis between two levels $j – 1$ and j

3. Similarity distance measuring

Using wavelet analysis we can define a similarity distance $S((x, y),(x', y'))$ for any pair of image points on the reference image $f(x, y)$ and the matched image $f'(x', y')$ on any given level.

3.1 Point-to-point similarity distance

If we consider single point to point match on the jth level, the similarity distance can be defined using only three differential components $D_{j,p} f(x, y)$, $p = 1, 2, 3$. For image matching to be invariant to the local image intensity, we use a normalization. This leads to the following implicit feature vector

$$\mathbf{B}_j(x,y) = [B_{j,1}(x,y) \quad B_{j,2}(x,y) \quad B_{j,3}(x,y)], \text{ where } B_{j,p} = \frac{D_{j,p}(x,y)}{|A_j(x,y)|}, \ p = 1, 2, 3, \tag{3.1}$$

where $|\cdot|$ denotes $\mathbf{L}_2$ norm. Then we can define a normalized similarity distance as

$$SB_j((x,y),(x',y')) = |\mathbf{B}_j(x,y) - \mathbf{B}'_j(x',y')|. \tag{3.2}$$

Note that in the above formulation, the position (x, y) refers to the continuous 2-D space, so we will need another definition which will by explained in the next chapter.

3.2 Local parallax continuity and generic pattern matching

For the robustness of image matching, we suppose local parallax continuity in a neighbourhood of two integer positions (k, l) and (k', l'), in another worlds, the parallax field in a neighbourhood of that two points will be nearly the same.

Let N denote the minimal neighbourhood of a given point containing the central position and four closest diagonal positions. We have

$$N = \{(0,0),(0.5,0.5),(-0.5,0.5),(0.5,-0.5),(-0.5,-0.5)\}. \tag{3.3}$$

Let $\mathbf{PA}_j(k, l)$ and $\mathbf{PD}_j(k, l)$ denote the approximation and difference pattern vector on an integer-indexed positions

$$\mathbf{PA}_j(k,l) = [A_j(k+r,l+c) \mid (r,c) \in N], \tag{3.4}$$

$$\mathbf{PD}_j(k,l) = [\mathbf{B}_j(k+r,l+c) \mid (r,c) \in N]. \tag{3.5}$$

Note that locations $(k+r, l+c)$ for $(r,c) \in N - \{(0,0)\}$ correspond to diagonal positions which can also be computed with rigorous bottom-up wavelet transform. Similarly, a normal position (k, l) on the reference image may be matched to a normal position (k', l') or diagonal position $(k'+r, l'+c)$ on the compared image. In order to use the generalized similarity distance $S_j((k,l),(k',l'))$, we need to compute two wavelet pyramids for each image $f(x, y)$: a standard one on normal integer positions (k, l) and another one on diagonal positions $(k+r, l+c)$.

The generalized similarity distance $S_j((k,l),(k',l'))$ is defined as

$$S_j((k,l),(k',l')) = SA_j((k,l),(k',l')) * SD_j((k,l),(k',l')), \tag{3.6}$$

where $SA_j((k,l),(k',l'))$ and $SD_j((k,l),(k',l'))$ are the similarity distances expressed in terms of approximation and difference on the jth level of analysis

$$SA_j((k,l),(k',l')) = 1 - \frac{\left|\mathbf{PA}_j(k,l) * \overline{\mathbf{PA}}'_j(k',l')\right|}{\left|\mathbf{PA}_j(k,l)\right|\left|\mathbf{PA}'_j(k',l')\right|}\,, \tag{3.7}$$

$$SD_j((k,l),(k',l')) = \left|\mathbf{PD}_j(k,l) - \mathbf{PD}'_j(k',l')\right|. \tag{3.8}$$

The term (3.7) is a multidimensional coefficient of the correlation. Using generalized similarity distance, we use full image information available on the *j*th level.

4. Matching two images

In this chapter, we deal with a problem of finding full correspondence between two given images. First, we get knowledge of finding the corresponding point if we know an approximate estimation of its position. Next, we find approximate estimation using spiral and hierarchical propagation.

4.1 Finding exact position in estimated neighbourhood

For any image point (*k*, *l*) in the reference image, its approximate correspondence (k'_0, l'_0) in the matched image may be obtained through some general strategies, such as a spiral and hierarchical propagation to be explained later on. The simplest way to catch the precise correspondence (k', l') is the discrete search in a small neighbourhood of (k'_0, l'_0) which is defined by a distance threshold T_r.

$$\min_{(k',l')} S_j((k,l),(k',l')) \quad \forall (k',l') : \left|(k',l') - (k'_0,l'_0)\right| \le T_r\,. \tag{4.1}$$

The distance threshold T_r should be defined in such a way that the allowed errors in the parallax field can be corrected. Typically, the threshold is $1 \le T_r \le 2$.

4.2 Spiral and hierarchical propagation

Without loss of generality, we only consider images with size $2^n \times 2^n$. By symbol M_j we denote the discrete parallax field on the *j*th level of the wavelet pyramid

$$M_j = (\mathbf{M}_j(k,l))_{k,l=0,1,\ldots,m-1}, \quad \text{where} \quad m = 2^{n\text{-}j}\,. \tag{4.2}$$

Each element of the parallax field $\mathbf{M}_j(k,l)$ contains the vector for a pair of homologous points (k,l) and (k',l')

$$\mathbf{M}_j(k,l) = (k',l') - (k,l)\,. \tag{4.3}$$

If there is no correspondence then $\mathbf{M}_j(k,l) = \emptyset$.

With at least 60% of overlapping, the central area on a chosen highest level should have correspondence on the other image. For example, if we start with searching on the level $j = n - 3$ with size 8 × 8, the central area of size 2 × 2 on the reference image should have correspondence on the matched image. So we can start searching for correspondence with the best matching of the central area using the generalized similarity distance defined in Chapter 3.2 in Eq. (3.6). Known parallax vectors can then be propagated from the central area to the outer rings. In each ring, we accurate the parallax vector with process described in the previous chapter in Eq. (4.1). This propagation is called a spiral propagation.

Gross errors of the resultant parallax field can be detected and corrected automatically by using the local parallax continuity constraint

$$|\,\mathbf{M}_j(k,l) - \overline{\mathbf{M}}_j(k,l)\,| \le T_M\,, \tag{4.4}$$

where $\overline{\mathbf{M}}_j(k,l)$ denotes the mean parallax field vector in the smallest nighbourhood of central position (k, l). The constant T_M denotes the maximal allowed parallax difference, usually $1 \le T_M \le 2$.

After image matching on the higher (j + 1)th level, the parallax field should then be propagated to the next lower jth level. This propagation is called a hierarchical propagation. We again accurate the parallax vectors with process described in the previous chapter in Eq. (4.1), we search for the precise correspondence in the smallest neighbourhood of the approximate estimation of the correspondence.

5. Used wavelet functions

All used wavelet and its associated scale functions are shown in Fig. 5.1. The discrete wavelet function g and its associated scale function h are mutually, therefore, we can easily compute the coefficients of the wavelet function g from the coefficients of the scale function h

$$g_k = (-1)^{k+1}\overline{h}_{1-k}\,. \tag{5.1}$$

On this account, I will write only the coeficients of the scale function.

Haar wavelets

Haar wavelets are the oldest wavelets. They are known to have the most compact support of 2, and they are orthogonal. The scale function h is

$$h(\text{Haar}) = (\tfrac{1}{2}, \tfrac{1}{2}). \tag{5.2}$$

Daubechie wavelets (Daub-4)

Daubechie wavelets are orthogonal with compact support of 4. They are constructed to have the maximum vanishing moments. The vanishing moments of l-order are defined as

$$\int_{-\infty}^{\infty} x^l \psi(x)\mathrm{d}x = 0. \tag{5.3}$$

The scale function for Daub-4 is

$$h(\text{Daub-4}) = (\frac{1+\sqrt{3}}{8}, \frac{3+\sqrt{3}}{8}, \frac{3-\sqrt{3}}{8}, \frac{1-\sqrt{3}}{8}). \tag{5.4}$$

Unfortunately, these wavelet functions are far from symmetric. Daubechie wavelets with the compact support length $L > 4$, denoted by Daub-L, also exist.

Least asymmetric Daubeshies wavelets (Symmlet-8)

They are constructed in the same way as Daubechie wavelets, but they are at least asymmetric. The scale function is

$$h(\text{Symmlet-8}) = (-0.05357445070, -0.02095548256, 0.35186953432, 0.56832912170, 0.21061726710, -0.07015881208, -0.00891235072, 0.02278517294). \tag{5.5}$$

Meyer Wavelets

They are orthogonal and symmetric, but they have no compact support. Nevertheless, for computing, the discrete version of Meyer wavelets exist. They have compact support of 62.

Symmetric complex Daubechies wavelets (Scd-4, Scd-6)

It can be shown that only complex valued scale and wavelet functions exist under the four hard conditions for orthogonality, compact support, maximum vanishing moments and finally symmetry. We denote Scd-6 the wavelet function with the shortest support of 6. It satisfies all these conditions, and it is associated with Daub-6 wavelet function. The scale function corresponding to Scd-6 is

$$h(\text{Scd-6}) = \frac{1}{64}(-3-i\sqrt{15},\ 5-i\sqrt{15},\ 30+i2\sqrt{15},\ 30+i2\sqrt{15},\ 5-i\sqrt{15},\ -3-i\sqrt{15}). \tag{5.6}$$

We already know that the Daubechie wavelet of length 4 exists. We can build the associated Scd-4 complex wavelet function, but it is highly non-smooth (see. Fig. 5.1). The vanishing moments doesn't exist for this wavelet. The corresponding scale function is

$$h(\text{Scd-4}) = \frac{1}{4}(1+i,\ 1-i,\ 1-i,\ 1+i). \tag{5.7}$$

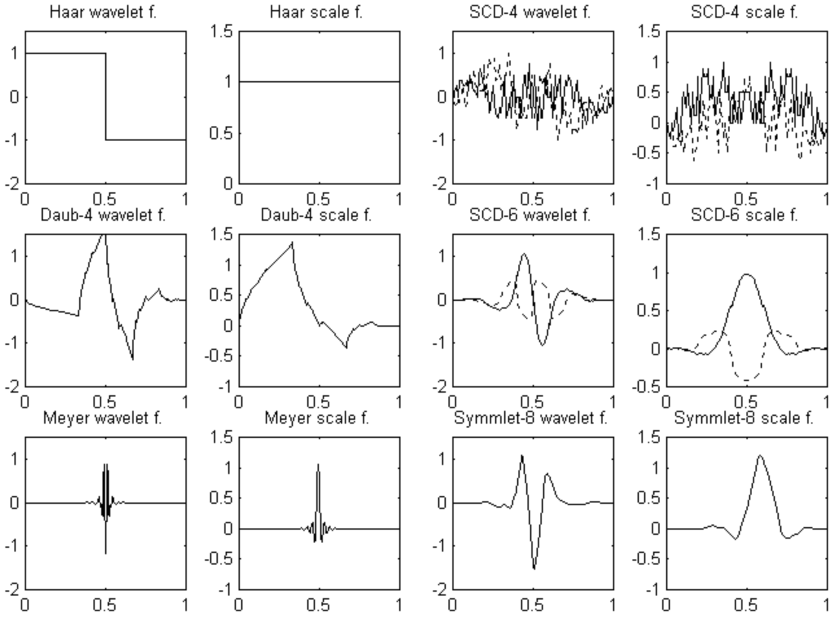

Figure 5.1: Tested wavelet and scale functions

6. Experiments

For my experiments, I chose a complex scene. I took two images of a clerical table with a digital camera from two different positions (see Fig. 2.2 or 6.3). I then used the above explained algorithm, and I experimented with different wavelet functions.

6.1 Relative gross error

Firstly, I measured the error of bad parallax field determination with respect to the chosen wavelet function. I measured the difference $\mathbf{V}_j(k, l)$ between the parallax vector $\mathbf{M}_j(k, l)$ for a given position (k, l) with the mean parallax vector $\overline{\mathbf{M}}_j(k, l)$ in the closest 3×3 neighbourhood

$$\mathbf{V}_j(k, l) = \mathbf{M}_j(k, l) - \overline{\mathbf{M}}_j(k, l). \tag{6.1}$$

Errors of the parallax vector $\mathbf{M}_j(k, l)$ satisfying the constraint $|\mathbf{V}_j(k, l)| < L_M$ (usually $1 \le L_M \le 2$.) can be corrected in a natural way. Nevertheless, the errors for which the expression $|\mathbf{V}_j(k, l)| \ge L_M$ holds are gross errors, and for such errors I provided a criterion I used for comparing the wavelet functions. The criterion of *relative gross error RGE* is

$$RGE = (\text{number of errors for } |\mathbf{V}_j(k,l)| \ge L_M) \,/\, (\text{all matched points}) \tag{6.2}$$

The results are summarized in Table 6.1. The best was the complex wavelet function Scd-4. The second one Scd-6 is better than all other real wavelet functions. From the set of real wavelets, the Meyer wavelet function is the best.

Wavelet	$RGE(M_5)$	$RGE(M_4)$	$RGE(M_3)$
Haar	0.05	0.133333	0.2531
Doub-4	0	0.1	0.24866
Symmlet-8	0.1	0.125	0.28496
Meyer	0.05	0.09167	0.23656
Scd-4	0	0.05	0.16578
Scd-6	0	0.06666	0.17972

Table 6.1: Relative gross errors (RGE) on levels 5, 4 and 3

6.2 Relative gross error *RGE* compared to value L_M

In the second test, I plotted the amount of relative gross error *RGE* for different values of L_M into the chart (see Fig. 6.1 and 6.2).

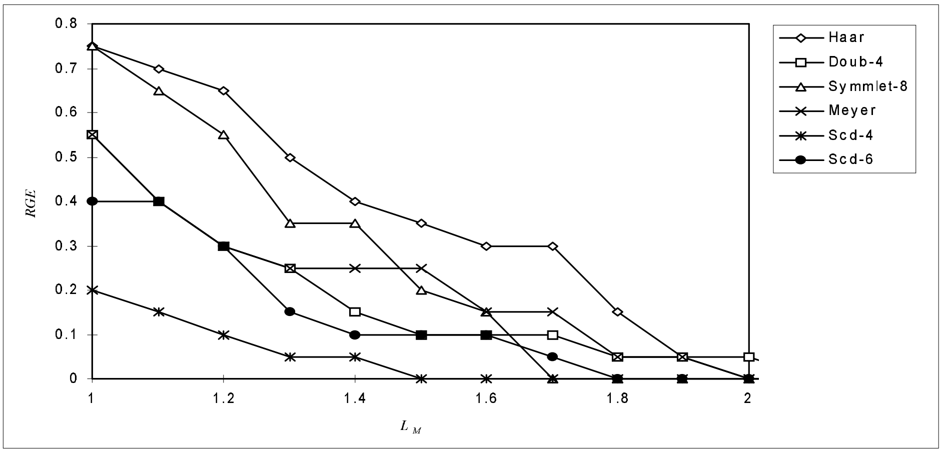

Figure 6.1: Relative gross error in comparison with the value L_M on 4th level of wavelet pyramid

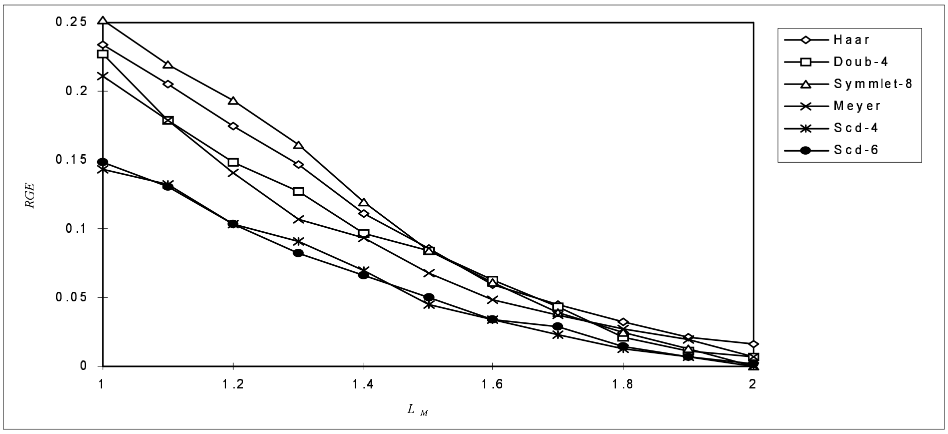

Figure 6.2: Relative gross error in comparison with the value L_M on 3th level of wavelet pyramid

This result shows us how large error is made by a given wavelet function. The faster the wavelet function falls the more suitable is for our use. Again, both the complex wavelet functions Scd-4 and Scd-6 are the best. From the set of the real wavelets, the Meyer wavelet function and the function Symmlet-8 are better than all others.

6.3 The resultant parallax field

As a last result, I depict resultant parallax field for the best wavelet function, which was the complex wavelet function Scd-4. In the figures, the parallax fields on different levels of a wavelet pyramid are depicted by making use of a mesh. I started finding the correspondence on the 5th level of the wavelet pyramid with the approximation of size 8 × 8. The result of searching on this level can be seen in Fig. 6.3. The figure on the left shows the reference image. The figure in the middle shows the parallax field just after the initialization in the reference image. The figure on the right shows the corrected parallax field in the reference image corrected with the discrete search in the neigbourhood of the approximate estimation of the correspondence).

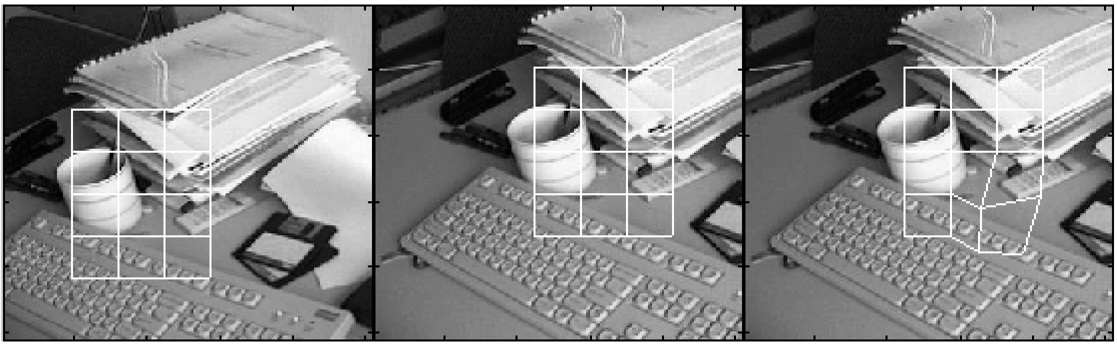

Figure 6.3: Parallax field found on the highest 5th level of size 8 × 8

Crossing from the level 5 to the level 4 with size of 16 × 16 is shown in Fig. 6.4, and crossing from the level 4 to the level 3 of size 32 × 32 is shown in Fig. 6.5. The figure in the middle shows the parallax field initiated just after the hierarchical propagation from the level 5 to 4 and from the level 4 to 3, respectively. The figure on the right shows the corrected parallax field.

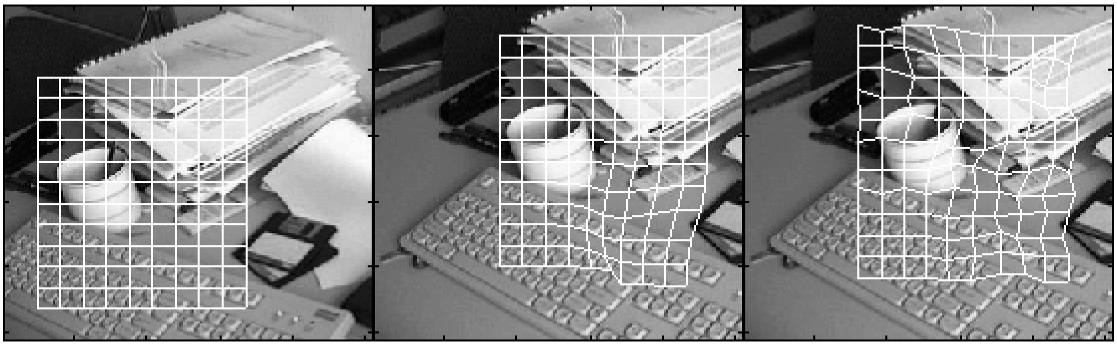

Figure 6.4: Parallax field found on the 4th level of size 16 × 16

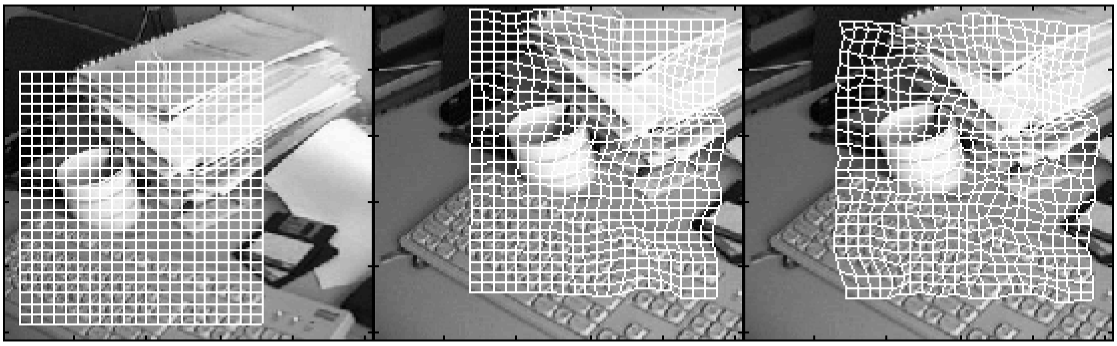

Figure 6.5: Parallax field found on the 3rd level of size 32 × 32

I evaluated the found correspondence between the two images by a naked eye. At a first glance, we can see that the mentioned algorithm works very well. There are only few errors in the middle area of the parallax field. The most errors are in the border area in which the constraint of continuity of the parallax field is hard to apply. Therefore, the next improvement of this algorithm should aid at removing the errors in the border area.

7. Conclusion

In this work, I explained the algorithm for matching two images taken by a digital camera. The algorithm is based on the usage of wavelet transform. I used several of the most important wavelet functions for testing. The symmetric complex wavelet functions (Scd-4 and Scd-6) proved to be the best choice. From the set of real wavelets, we recommend to use the Meyer wavelet function. Except for the border parts, the above explained algorithm for the image matching works very well.

8. References

[1] Castleman, C.R.: *Digital image processing*, Prentice Hall, Upper Saddle River, New Jersey, 1996.

[2] Častová, N.: Teaching text for seminar about Integral transforms and spectral analysis, VŠB TU Ostrava, 2000.

[3] Wavelet Toolbox User’s Guide by The MathWorks, http://www.mathworks.com, 2000.

[4] Pan, H.: *General Stereo Matching Using Symmetric Complex Wavelets*, paper presented at SPIE Conference: Wavelet Applications in Signal and Image Processing IV, August 1996, Denver, USA, published in SPIE Proceedings vol. 2825.