Hierarchical Multipoint Matching

Mengxiang Li
Department of Photogrammetry and Computational Vision and Active Perception (CVAP) Laboratory, Department of Numerical Analysis and Computing Science, The Royal Institute of Technology, S-100 44 Stockholm, Sweden

ABSTRACT: The problem of hierarchical structures for multipoint matching is addressed. Related to this problem, three structures, namely, multigrid, multiresolution, and the combination of them, have been developed and investigated. The multigrid structure extends the regular grid points, on which the multipoint matching is performed, to a number of levels in spatial resolution from coarse to fine, while the multiresolution structure extends the image resolution to a number of levels from lower to higher resolution. Each of these two structures, in turn, removes one of the factors which handicap the efficiency of the multipoint matching. The combination of these two structures then leads to a structure which achieves high computational efficiency in multipoint matching. The developed structures have been tested on a number of images, and results are presented.

INTRODUCTION AND MOTIVATIONS

THE MULTIPOINT MATCHING ALGORITHM by Rosenholm (1986a, 1986b, 1987, 1988) is a very attractive image matching algorithm, in particular, when applied for automatic digital terrain model (DTM) generation. It is a global approach, and has higher reliability compared with single point matching algorithms, due to its global smoothness constraints. On the other hand, it has the drawback of being computationally expensive. It has been noticed by, e.g., Rosenholm (1987, 1988) and Kölbl *et al.* (1988), that it has a very slow convergency rate. In order to overcome this problem, Rosenholm (1987, 1988) uses a low-pass filtering technique to preprocess the images. His results show that low-pass filtering usually improves the convergency speed, but not always. It is known that low-pass filtering has the effect of removing or smoothing high frequency information (and errors) and sometimes also smoothing over disturbances in the images. In this aspect it improves the convergency speed. On the other hand, low-pass filtering may also smooth important information, decreasing precision. This has been shown by Rosenholm (1988). Korten *et al.* (1988) have investigated methods for improving the convergency in their FAST-Vision system, which is similar to our problem in the sense of convergency. The most favorable approach for speeding up convergency, according to their experiences, is the frequency related average (FRA) method. This method also reduces image frequency and smoothes high frequency errors.

Reasons for the slow convergency speed of the multipoint matching are due to (a) high frequencies information of image (gray values), (b) disturbances over images, and (c) low frequencies components of the solution, i.e., parallaxes. The first two factors come from the image (gray value) information. The third point is due to the structure of the problem, i.e., a large equation system with regularly (grid) distributed points. The slow convergency speed is a common problem for such a structure. All these attempts aforementioned for improving the convergency speed are effective in smoothing the first two factors, which are related to the image gray values. But the third factor, which is not related to the image gray value but rather to the structure of the problem, is not effected.

The speeding up of the convergency from the third point of view is motivated by the idea of using the multigrid method. The multigrid method is a well developed numerical method for solving large equation systems with regularly grid spaced points to achieve high efficiency. The structure of our multipoint matching problem is very similar to that so that the multigrid structure can be adapted, that is to extend the grid structure to a number of levels from coarse to fine. The coarse levels deal very efficiently with low frequency components of the solution, while finer levels deal with the high frequency components of the solutions so as to obtain accuracy.

To speed up convergency from the first two points of view, the methods mentioned above, such as low-pass filtering and the FRA, are effective. But together with the multigrid structure, a more effective way to obtain efficiency is the use of multiresolution. The multipoint image matching processing is not just to solve a linear or non-linear equation system, but also involves the formation of the equation system, which is a processing on the pixel level (each pixel is involved). Besides that, the matching also involves some other processing which is also on the pixel level, e.g., resampling of the right image gray values. This means we have to reduce the number of pixels if we want to speed up the processing. This motivates the use of multiple resolution of image. Multiresolution together with the multigrid structure forms the hierarchical structure of our multipoint matching. At the lower multigrid level, lower resolution is used; at higher multigrid level higher resolution is used.

In this paper, we present the multigrid and multiresolution structures for multipoint matching. The aim of the study is to achieve high computational efficiency and reliability. First, we present the multipoint matching algorithm, multigrid method, and multiresolution technique briefly. Then we formulate the multipoint matching in multigrid structure, and multiresolution structure separately. And, finally, we combine the multigrid and multiresolution together into the multipoint matching to form a hierarchical structure. Experiments are performed and numerical results are provided. At the end of this paper, we give some general discussion on the developed method to related works and draw some conclusions.

THE MULTIPOINT MATCHING ALGORITHM

The multipoint matching algorithm (Rosenholm 1986a, 1986b, 1987, 1988) is basically an area-based matching method using the least-squares technique. It is regarded as an extension of single point least-squares matching techniques (Förstner, 1982; Ackermann, 1984; Grün, 1985). But unlike a conventional single point area-based matching method, which matches one point (window) at a time, the multipoint matching matches a group of points simultaneously. The points are usually located in a regular grid form in one of the images (see Figure 1). These grid points are connected to each other by the finite element method by means of a bilinear function. Smoothness constraints on object surface are imposed on the solution, which makes it different from a single point matching algorithm too. In this aspect, it is similar to some other global matching algorithms, e.g., the stochastic optimization approach by Barnard (1989), or matching using regularization (Poggio, 1985; March, 1988), and

PHOTOGRAMMETRIC ENGINEERING & REMOTE SENSING,
Vol. 57, No. 8, August 1991, pp. 1039–1047.

0099-1112/91/5708–1039$03.00/0

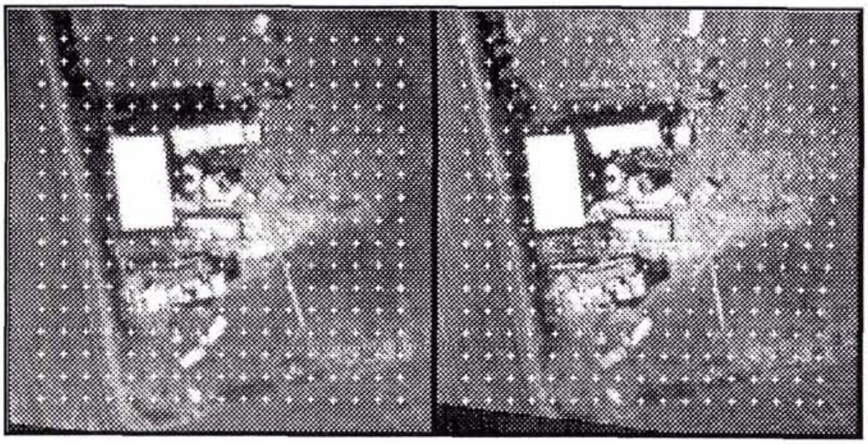

FIG. 1. Multipoint matching. All the grid points (the white crosses) are matched from the left image to the right image simultaneously. The neighborhood grid points are connected by a bilinear surface model.

minimization of energy functional using continuation methods (Witkin *et al.*, 1987).

Here the problem is cast as a minimization problem, i.e., to minimize the differences between the two matching images under the smoothness constraints. The problem is solved by a least-squares estimation technique. For every point (pixel) (x,y) from the images, we can formulate the following observation equation:

$$g_1(x,y) - g_2(x + p,y) = n(x,y) \tag{1}$$

where g_1 and g_2 are gray values of the first and second matching images, $n(x,y)$ is the noise, and p is the parallax at this point, which is interpolated bilinearly from the parallaxes of its four neighborhood grid points $p_{i,j}$ (the cross points in Figure 1) (see Rosenholm, 1986a). In Equation 1, we assume the normal case image geometry, i.e., no y-parallaxes. $p_{i,j}$s are the unknown parameters and to be determined in the least-squares estimation.

The smoothness constraints are formulated as fictitious observations with special weights $w_{i,j}$: i.e.,

$$\begin{cases} p''_{xx} = 2p_{i,j} - p_{i,j+1} - p_{i,j-1} = 0, \; w_{xx_{i,j}} \\ p''_{yy} = 2p_{i,j} - p_{i+1,j} - p_{i-1,j} = 0, \; w_{yy_{i,j}} \end{cases} \tag{2}$$

$$\begin{cases} p'_{x} = p_{i,j} - p_{i,j+1} = 0, \; w_{x_{i,j}} \\ p'_{y} = p_{i,j} - p_{i+1,j} = 0, \; w_{y_{i,j}} \end{cases} \tag{3}$$

Equations 1, 2, and 3 form the basic observation equation systems for the least-squares estimation. As Equation 1 is nonlinear, it needs to be linearized. The linearized equation systems together with Equations 2 and 3 in matrix form look like this:

$$\mathbf{AX} = \mathbf{L} + \mathbf{V}, \mathbf{W}. \tag{4}$$

Minimization of $\mathbf{V}^T \mathbf{WV}$ leads to the following normal equation:

$$(\mathbf{A}^T \mathbf{W} \mathbf{A}) \mathbf{X} = \mathbf{A}^T \mathbf{WL} \text{ or } \mathbf{N} \mathbf{X} = \mathbf{U}. \tag{5}$$

The rest is a standard least-squares estimation problem. Due to the linearization of the nonlinear observation equation (Equation 1), the solution needs to be iterated (a Newton-Gauss iteration) until certain criterion is satisfied.

MULTIGRID MULTIPOINT MATCHING

The multigrid method is a very well developed numerical method for solving large sparse system of linear or nonlinear equations, in particular, with grid spaced unknown points. The method was developed at the beginning for solving discrete elliptic boundary problems. Now it has been applied to many other numerical analysis problems in physics and engineering. It has also been applied to visual surface reconstruction (Terzopoulos, 1983; Terapoulos, 1984), image analysis (Terzopoulos, 1986), and digital terrain model (DTM) interpolation in photogrammetry (Ebner *et al.*, 1986). For a detailed description of the multigrid methods and applications see, e.g., Brandt (1982), Stüben *et al.* (1982), and Hackbusch (1985). We briefly describe the method in the following subsection.

THE MULTIGRID METHODOLOGY

The aim of the multigrid method is to gain computational efficiency. When one tries to solve a linear equation system of the form $\mathbf{L}_h\mathbf{X}_h = \mathbf{F}_h$ (where $\mathbf{L}_h$ is a linear operator, $\mathbf{X}_h$ is the unknown vector, and $\mathbf{F}_h$ is the right hand side vector. They are all defined on a grid domain $\mathbf{\Omega}_h$ and assuming $\mathbf{L}_h^{-1}$ exists) by classic relaxation methods, such as *Jacobi's method* or *Gauss-Seidel's method*, it is found that the convergency is very slow. The reason is that relaxation is a local operation. High frequency[1] components of the corrections (or errors) in the relaxation can be smoothed out quickly, while the low frequency part remains for a long time. They inhibit global information propagation, thus inhibiting convergency. Based on this observation, the basic idea of the multigrid method is to extend the grid structure to a number of levels in spatial resolution, in a coarse-to-fine strategy. The low frequencies at a finer level become high frequencies at a coarser level. Errors with these frequencies can be smoothed out very quickly by relaxation at this coarse level. In a successive scheme, transferring the already obtained results from level to level, error at any frequency (between a certain band) can be smoothed quickly at a certain level of the grid. Moreover, at coarse grid, the dimension of the system is reduced (by a factor of 4 usually), so the computation can also be done quickly. These features make the multigrid a very attractive and efficient tool in solving large linear systems defined by a regular grid.

There are different multigrid methods. One of them is the so-called *full multigrid* or *nested iteration* method, which suits our multipoint matching problem. The full multrigrid method can be simply understood as one to obtain good initial approximation from a coarser grid for the next finer grid. In the full multigrid, the relaxation starts from the coarsest grid level as predefined. At each level higher than the coarest level, a number of iteration steps of a suitable multigrid method is performed, i.e., performing a multigrid cycle from the current level

[1]Here high and low frequencies of the correction or error are related to the grid spacing. For errors whose wavelengthes are larger than the grid space are regarded as low frequency errors, and vice vera.

to the coarsest level. The result is then transferred to the next finer grid by an interpolation operator. In our case, the multigrid cycle is replaced by a number of Newton-Gauss iterations of direct solution within the same level.

Application for Multipoint Matching

Multipoint matching is formulated as a least-squares estimation problem. In our case the multigrid method cannot be applied directly because, firstly, the multipoint matching problem is not a determined linear or nonlinear system of equations, which is the case for the multigrid method. And secondly, the normal matrix **N** in Equation 5 is usually not a diagonally dominant matrix, although it is a sparse system. This means that classical relaxation methods, such as the Jacobi method or Gauss-Seidel method, are not suitable for solving the system. A direct solution is more suitable in this case. Further, the first derivative of the second image gray value function $g'_{2_x}(x,y)$ can be replaced by the first derivative of the first image gray value function $g'_{1_x}(x,y)$, and this replacement has a great advantage in obtaining computational efficiency, because $g'_{1_x}(y,x)$ is independent of the unknown **X** (Li, 1989). So for each Newton iteration, the design matrix **A** is not changed and neither is the normal matrix **N**. Once the inverse $\mathbf{N}^{-1}$ is computed, it is valid for all the iterations. For each Newton iteration, only the right hand side needs to be reformulated and one matrix multiplication $\mathbf{N}^{-1}\mathbf{U}$ needs to be performed. This saves much computational work and makes the direct solution more efficient than relaxation methods. The more iterations needed, the more efficient is this solution than the relaxation solution.

But on the other hand, multipoint matching does have some special characteristics, which makes it suitable for the multigrid solution, i.e., the regular grid-spaced matching points. The multigrid structure extends the grid to a number of levels from coarse to fine, so the spatial resolution of the grid crosses a wide range and the matching is performed in a strategy of *coarse-to-fine*. The coarse grid levels allow direct information communications over a large range, achieving global propagation of information, while fine grid levels treat more local and detailed information flow in obtaining accuracy. In this way the overall efficiency of the matching is increased. Here we use the similar idea as the full multigrid method, i.e., to obtain good initial approximate values for a fine grid from a coarse grid in an efficient way so as to accelerate the processing. But we replace the multigrid cycle over a number of levels in the full multigrid by direct solution of a number of Newton iterations within the same level. The information flow in this case is the unknown **X** from coarse level to fine level. The coarse-to-fine transfer operator is a simple bilinear interpolation as

$$\mathbf{I}_H^h = \frac{1}{4}\begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}_{2h}^{h} . \tag{6}$$

Let the number of levels in the multigrid structure be K, the grid point changing ratio from level to level be 2:1, and the grid point number be in the order of $(2^k + 1) \times (2^k + 1)$, where k is the level number. The grid spacing h, in terms of pixels, changes in a ratio of 1:2 from coarse to fine level, while the image resolution is unchanged. Such a multigrid structure is shown in Figure 2 with an example of four levels. The matching starts from the coarsest level Ω_1 ($k = 1$), with grid point number of 3 by 3 and zero initials. The grid spacing for level k is $h_k = 2^{K-k}h$ (h is the grid spacing of the finest grid level K). The coarse to fine transfer of the unknown **X** uses the bilinear interpolation operator in Equation 6. At each level higher than the coarsest level, a standard multipoint procedure is performed with the initial value obtained from the previous level. After convergence is reached, the result is transferred to the next finer level. This processing proceeds successively to the finest level.

MULTIPOINT MATCHING WITH MULTIRESOLUTION

The multiresolution structure has been widely used in image analysis and vision related problems, such as feature detection, motion estimation, image matching, and visual surface reconstruction, etc. Terzopoulos (1983, 1984) developed a method of multilevel reconstruction of a visual surface. Terzopoulos (1986) also used multiresolution of images in connection with the multigrid method to compute lightness, shape-from-shading, and optic flow. For feature detection, e.g., edges, the pyramidal structure is also often used, (see, e.g., Rosenfeld (1983)). In

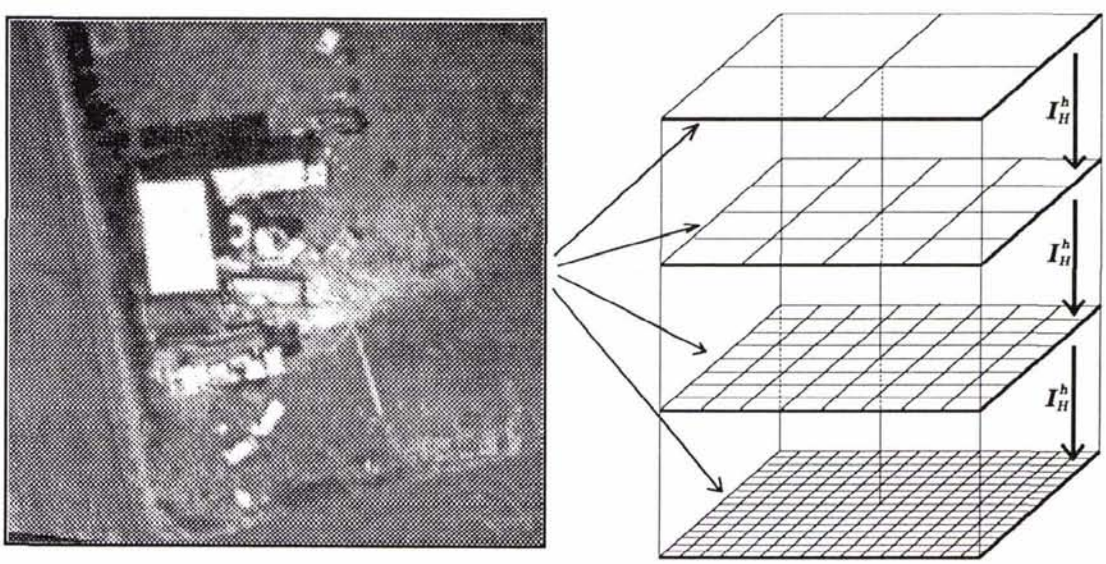

Fig. 2. The 2:1 multigrid structure of multipoint matching with four levels. The left shows the image,and the right shows the multigrid structure of the matching points. The same image (resolution) is used at different grid levels, and the result is transferred from level to level by the interpolation operator $\mathbf{I}_H^h$.

image matching, multiresolution or pyramid is often used in a hierarchical structure, typically to derive approximate values from lower resolution for higher resolution. Examples are too numerous to list here.

PYRAMID - MULTIRESOLUTION STRUCTURE

Multiresolution structure of image or image pyramid is a data structure which consists of a sequence of images of the same object presented at successively reduced resolutions. Such a structure contains no information that is not implicitly presented in the finest version of the image in the sequence (the original image), but it has great potential of gaining computational efficiency in making some of this information explicit. It also has the property that only a small overhead in memory space relative to the input image is required. These two characteristics make the multiresolution structure a very efficient tool in image analysis.

In general, the way of generating an image pyramid from a given image can be described as follows (see, e.g., Wong *et al.* (1978), Burt *et al.* (1983), and Meer *et al.* (1987)). First, the image $g_{k-1}(x,y)$ at level $k-1$ of the pyramid is low-pass filtered. This is equivalent to convolving the image with a local symmetrical weight function or generating kernel $\mu(x,y)$. After the convolution, the low-pass filtered image is resampled with a reduced sample density $s(x,y)$. So

$$g_k(x,y) = (\mu(x,y) \otimes g_{k-1}(x, y)) \oplus s(x, y) \tag{7}$$

where $\otimes$ denotes the convolution operation and $\oplus$ denotes the resampling. Usually there are some constraints on the generating kernel, such as separability, symmetry, normalization, etc. For the choice of the optimal generating kernel with respect to minimum information loss and computational efficiency, see Burt *et al.* (1983) and Merr *et al.* (1987). One of the most commonly used convolution kernels is the unimodal Gaussian-like function. A pyramid generated by such a kernel is often called a *Gaussian pyramid*. In contrast to the Gaussian pyramid, the *Laplacian pyramid* is also sometimes used. A Laplacian pyramid is a sequence of error images, which are the differences between two neighboring levels in the sequence of a Gaussian pyramid (Burt *et al.*, 1983). For fast computation of a Laplacian pyramid, also see Crowley *et al.* (1984).

The simplest way of building a resolution pyramid is the 2 by 2 block averaging method. Starting from the original image (the finest version of the image or the base of the pyramid), each new level is generated by averaging a 2 by 2 non-overlapping pixel block from the last level. Repeating the same procedure, the operation successively traces down to the root node of the pyramid. One can see that the image size in the pyramid generated in this way is reduced by a factor of 4. The generating kernel μ is the following:

$$\mu = \frac{1}{4}\begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix}. \tag{8}$$

It can be shown that only a third of the original image memory is needed for additional images in the pyramid representation.

MULTIRESOLUTION APPLIED IN MULTIPOINT MATCHING

Multiresolution representation of images offers advantages in obtaining computational efficiency. Object has scale. Large objects (in the image gray value structure) can be more easily recognized from the image or matched from one image to another at a lower resolution (large scale). This is due to the fact that, firstly, at lower resolution, information to be processed is much less than at higher resolution, so the computational work can be done more efficiently. Secondly, sometimes too detailed information (fine structure) makes the recognition or matching confusing, while at lower resolution the detailed information is reduced very much, which reduces the confusing factors. And thirdly, the techniques of generating image pyramids as discussed in the previous subsection have the effect of removing noise and improving the signal-to-noise ratio of the lower levels of the pyramid. In this sense, the matching at lower levels of the pyramid can be done more reliably. Also at lower resolution, the parallax is reduced, which means that for the same parallax it is much smaller at the lower resolution in terms of pixels than at the higher resolution. This scaling in parallax reduces the search space and so increases the pull-in effect of the matching.

On the other hand, the precision of matching at lower resolution is limited as the spatial information preserved at lower resolution is limited. As the corresponding dimension of a pixel in object space is increased at lower resolution, the obtained precision of matching is subject to this resolution. Generally, the lower the resolution, the lower the precision of the matching. On the other hand, if the resolution is higher than a certain level, increasing the resolution would not increase the precision significantly. If we want to have a certain precision, a certain level of resolution (at the finest level) has to be used.

This discussion motivates the use of multiresolution in a successive scheme for image matching. The low resolution allows the matching to treat primitive information fast and reliably, and that makes a coarse but fast and reliable correspondence between the two images based on primitive information. This coarser correspondence can then be used to guide to finer matching. By transferring the results from a lower resolution level to a higher resolution level, the higher resolution now allows the matching to concentrate on more detailed information and so achieve higher matching precision. In this successive schedule over a number of resolution levels, the matching is performed in a more efficient way. The number of levels in the pyramidal structure is, of course, problem-dependent, it need not be fixed.

Again, let the number of levels of resolution be K where we define the finest level as K. We let the image size at the finest level be 2^n by 2^n. As we only change resolution from level to level, the number of the grid points is kept constant in all levels, say $(2^K + 1)$ by $(2^K + 1)$ grid points. This means that the grid spacing, h, changes from level to level. If we use the 2:1 resolution reduction ratio from level to level, the h reduction ratio is also 2:1. Figure 3 shows the structure of multipoint matching with multiresolution.

The matching starts from the lowest resolution level R_1 with zero initials. At level k of the structure (for $1 < k \leq K$), with the image size of 2^{n-K+k} by 2^{n-K+k} and the grid spacing of $h_k = h/2^{K-k}$, a standard multipoint matching, but with initial approximation values from the last level, is performed. The results (parallaxes at the grid points) are then transferred to the next level. The transfer operation in this case is a multiplication by a factor of 2 due to the resolution reduction such as

$$\mathbf{I}_R^r = (2)_{2r}^r. \tag{9}$$

COMBINATION OF MULTIGRID AND MULTIRESOLUTION

So far we have formulated the multipoint matching with multigrid structure and with multiresolution independently. Multigrid and multiresolution, in the sense of structure, could be equivalent in some special cases of image analysis problems, e.g., in the Terzopoulos (1986) case, but they are different in our multipoint matching case. They have different effects on the matching processing. The multigrid takes advantage of the special structure of the matching problem, i.e., the regular grid spaced points for the matching, and allows the distribution of grid points to cross a number of spatial resolution levels, from coarse-to-fine. Coarse grid allows the process to communicate

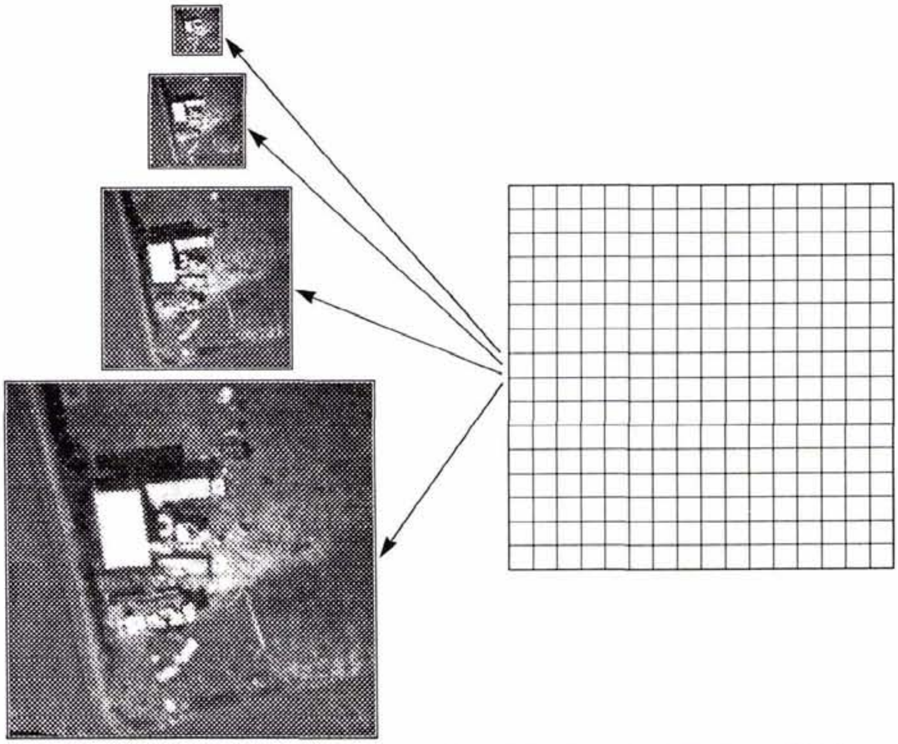

FIG. 3. Structure of multipoint matching with multiresolution. The left shows the multiple image resolution structure from lower to higher level (top-down). The right shows the grid points structure, which is the same for all levels.

directly between points over a large area and so global information flow is achieved, while fine grid treats more local communication between points which increases the model fidelity of the matching. In this way it improves the efficiency. The multiresolution structure improves the efficiency of the matching due to the fact that image gray value structures vary in resolution. The coarse resolution treats primitive information for efficiency and fine resolution treats detail information for precision.

Both multigrid and multiresolution suffer from difficulties, but in a different way, in obtaining more efficient computation. In the multigrid case the grid points are reduced from level to level, so the size of the normal equation matrix is reduced. This leads to processing on the grid point level, i.e., the matrix operation is faster at the lower level. But the pixels are not reduced from level to level, as the matching involves processing on the pixel level; this handicaps the computational efficiency of the multigrid method in this case. On the other hand, the multiresolution is just working in an opposite way, i.e., at the lower level pixels are reduced so processing on the pixel level is done fast, but processing on the grid level (matrix inverse and multiplication) is not reduced.

The above analysis of multigrid and multiresolution performances leads, naturally, to the combination of them into the multipoint matching at the same time, in such a way that advantages of both structures are promoted as much as possible while disadvantages are avoided as much as possible. The combination of them is, of course, not unique. There are many different ways to do so. But they should be combined in such a way that both computational efficiency and precision can be achieved. So the general rule is a *lower grid level with lower image resolution*, and a higher grid level with higher image resolution. In this way the matching processing is performed fast on both grid level and pixel level at the lower level of the matching, and at the higher level of the matching the precision on grid level and pixel level can be improved. In the following we will formulate a structure (combination) of multigrid and multiresolution for the multipoint matching. In this structure the grid point and pixel keep homogeneity in resolution increasing or decreasing.

Similar to previous ones, we let the number of levels be K, the original image size be 2^n by 2^n and the finest grid points be in the order of $(2^m + 1)$ by $(2^m + 1)$, and the grid spacing be h. We use the 2:1 coarsening ratio for the grid points from level to level and the same ratio for the resolution reduction. In this way the grid spacing h in terms of pixel is unchanged from level to level. Figure 4 shows the structure of multipoint matching with multigrid and multiresolution.

The matching starts from the lowest level, L_1, with zero initials. At level k of the structure (for $1 < k \leq K$), with the image size of 2^{n-K+k} by 2^{n-K+k} and the grid points of $(2^{m-K+k}+1)$ by $(2^{m-K+k}+1)$, a standard multipoint matching is performed but with initial approximate values from the previous level. The transform operation now is a combination of $\mathbf{I}^h{}_H$ as given in Equation 6 and $\mathbf{I}^r{}_R$ as given in Equation 9, that is,

$$\mathbf{I}^{h+r}_{H+R} = \mathbf{I}^h_H\, \mathbf{I}^r_R = \frac{1}{2}\begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}^{h+r}_{2(h+r)}. \qquad (10)$$

EXPERIMENT AND RESULTS

The multipoint matching with different structures as formulated in the previous sections have been tested on various image data sets, mostly concerning computational efficiency. For the purpose of comparison the following tests have been performed: (*i*) single level multipoint matching (the original algorithm), (*ii*) multigrid multipoint matching, (iii) multipoint matching with multiresolution, and (*iv*) multipoint matching

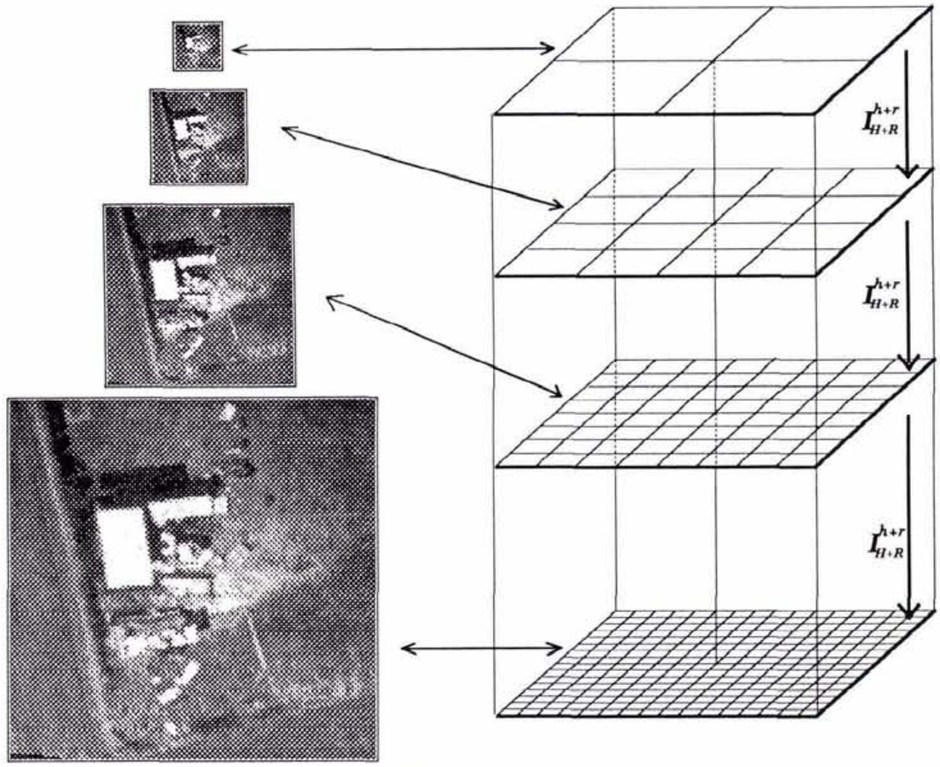

FIG. 4. Structure of multipoint matching with multigrid and multiresolution. The left shows the multi-image-resolution structure from coarser to finer and the right shows the multigrid structure from coarser to finer (top-down).

with combination of multigrid and multiresolution. Three sets of image data are used for testing the algorithms. They represent different types of images and applications of image matching. All the images have the size of 300 by 300 pixels. Data sets A are three pairs of digitized aerial photographs, B are three pair of close-range images, and C are three pair of synthetic images.

Results of Multigrid Multipoint Matching

The multipoint matching in multigrid structure is applied to all the test images. Four levels are used in the test. The grid points from level 1 to level 4 are 3×3, 5×5, 9×9, and 17×17, respectively. The grid spacings are 128×128, 64×64, 32×32, and 16×16 pixels respectively from level 1 to level 4. The matching always starts from zero initial approximate values at level 1. The results are presented in Table 1 together with the results of the original single level matching.

From Table 1 we can see that multipoint matching with multigrid structure does not reduce the computational work, if the single level does not require much computational work. For example, when the two matching images are not too far away, i.e., the parallax-range is rather small. As shown in Table 1 that for image set A and B (except *Bb*), although the number of iterations at the last level is reduced very much, the total number of iterations is not reduced, and neither is the total CPU time. This is expected, because the computational work at the lower level is not reduced very much at each iteration. The reduction is only in the matrices computations; the rest is the same at all levels. On the other hand, the multigrid structure has improvements in efficiency compared to single level multipoint matching for image pairs with a large parallax range. As in examples *Bb,Cb,Cc*, single level does not get to the correct match even after a large number of iterations, while multigrid matching does match correctly and with less computational work. In this sense, multigrid structure also improves reliability.

Results of Multipoint Matching with Multiresolution

In this section we give some examples of results of multipoint matching with multiresolution only, i.e., image resolution changes from level to level, while grid points do not change. Here the grid points are 17 by 17 at all levels for all the data sets shown here. Table 2 shows the results. In Table 2, level 4 is the finest level with the original resolution. From the table, we can see the following. The total computational work (CPU time or in terms of seconds per grid point *s/p*) for the matching is not reduced in cases where the single level does not require very much work, e.g., the parallax range is not to large. But if the single level needs a large number of iterations, e.g., with large parallax range, the multiresolution reduces the computational work. In the examples given here, data set A is the case where multiresolution does not improve efficiency, while, for example *Bb* and set C, multiresolution improves efficiency. This is similar to the results of multigrid multipoint matching but more efficient (compare the reduction of *s/p* in Table 1 and Table 2 for these examples). For examples with which the CPU is increased, the reasons for the increase of CPU time are different. Here the increase in CPU time is mainly caused by the computation of matrix inverse of the normal equation (a 289 by 289 matrix inverse needs about 1 minute and 11 seconds CPU time, which is the major part of the matching at the lower levels), which is the same for all levels. The more levels used, the less efficient. Here we used four successive resolution levels. It has been shown that it is not necessary to do so (Li, 1989). Sometimes, two levels may give better results.

TABLE 1. COMPUTATIONAL WORK OF MULTIGRID MULTIPOINT MATCHING.

data	level1 3 × 3		level2 5 × 5		level3 9 × 9		level4 17 × 17		total		one-level 17 × 17			red.
set	it	CPU	it	CPU	it	CPU	it	CPU	CPU	s/p	it.	CPU	s/p	(%)
Aa	12	3'03"	4	1'11"	3	1'13"	3	2'12"	7'39"	1.59"	17	6'19"	1.31"	−21.37
Ab	15	4'45"	3	1'11"	4	1'31"	3	2'28"	9'55"	2.06"	20	6'51"	1.42"	−45.07
Ac	11	2'49"	4	1'13"	4	1'13"	5	2'40"	7'55"	1.64"	15	5'42"	1.18"	−38.98
Ba	8	2'09"	2	43"	3	1'00"	3	2'12"	6'04"	1.26"	9	4'21"	0.90"	−40.00
Bb	73	17'21"	2	44"	2	44"	2	1'58"	20'48"	4.32"	247*	62'28"	12.97"	66.69
Bc	11	2'55"	3	58"	3	1'00"	3	2'14"	7'07"	1.48"	19	6'44"	1.40"	−5.71
Ca	10	2'36"	1	29"	1	30"	1	1'41"	5'16"	1.09"	20	4'50"	1.42"	23.24
Cb	39	9'22"	52	12'25"	5	1'27"	3	2'10"	25'24"	5.27"	247*	62'04"	12.90"	58.91
Cc	42	10'09"	7	1'54"	6	1'43"	4	2'28"	16'14"	3.37"	167*	42'46"	8.9"	61.80

Note: *it* = iterations spent on each level or total iterations; *CPU* = CPU time in minute (') and seconds (") spent on each level or total CPU time; *s/p* = CPU time counted in seconds per grid point; *One level* means the original single level multipoint matching. The last column (*red.* (%)) is the reduction of CPU time in (*s/p*), where a minus sign(−) indicates that the CPU time is not reduced but increased. *For image sets *Bb*, *Cb*, and *Cc*, the results are still not yet correct, the matching falls into local minimum.

TABLE 2. COMPUTATIONAL WORK OF MULTIPOINT MATCHING WITH MULTIRESOLUTION. SEE TEXT NOTE IN TABLE 1.

data	level1 17 × 17		level2 17×17		level3 17×17		level4 17 × 17		total		one-level 17 × 17			red.
set	it	CPU	it	CPU	it	CPU	it	CPU	CPU	s/p	it.	CPU	s/p	(%)
Aa	16	1'28"	10	1'30"	4	1'34"	3	2'11"	7'15"	1.51"	17	6'19"	1.31"	−15.27
Ab	6	1'59"	5	1'22"	5	1'40"	3	2'23"	7'08"	1.48"	20	6'51"	1.42"	−4.23
Ac	13	1'25"	15	1'38"	10	2'01"	4	2'28"	8'04"	1.67"	15	5'42"	1.18"	−41.53
Ba	15	2'01"	6	1'25"	3	1'31"	2	1'59"	6'56"	1.44"	9	4'21"	0.90"	−60.00
Bb	25	1'37"	18	1'44"	4	1'36"	3	2'17"	7'46"	1.61"	247	62'28"	12.97"	87.59
Bc	20	2'05"	8	1'27"	4	1'34"	4	2'28"	7'34"	1.57"	19	6'44"	1.40"	−10.56
Cb	61	2'10"	6	1'26"	3	1'32"	2	1'59"	7'08"	1.48"	247	62'04"	12.90"	88.53
Cc	15	1'31"	8	1'32"	3	1'38"	2	2'07"	6'50"	1.42"	167	42'46"	8.9"	84.05

RESULTS OF COMBINATION OF MULTIGRID AND MULTIRESOLUTION

Results from the previous subsections show us that neither multigrid or multiresolution alone in multipoint matching do not always improve computational efficiency for reasons we have explained, which suggests the use of multigrid and multiresolution at the same time (lower grid level with lower image resolution), and vice versa, in such a way that the computational work can be reduced. Now we will present the results of the matching with the combination of multigrid and multiresolution. The grid points are 3 × 3, 5 × 5, 9 × 9, and 17 × 17 from lower level to higher level and the image resolution is reduced by a factor of 2 from higher level to lower level. The computational work for each level is shown in Table 3 together with the results of one-level multi-point matching for comparison. From Table 3 we can see the following:

(1) In multigrid multipoint matching with multiresolution, the total CPU time, counted in seconds per point, is drastically reduced compared with single level matching for all the image sets shown here. The reduction of CPU time (*s/p*) ranges from 26 percent to 95 percent, depending on the data, with an average reduction of 60 percent for all the examples.

(2) The CPU time is also reduced compared with multipoint matching with multigrid or multiresolution alone for all the examples. As we can see, the first two levels need only a few seconds compared with a few minutes in both multigrid and multiresolution alone. This is due to the fact that the pixels and normal equations are both reduced at the lower levels. This reduces the amount of computational work for each iteration as well as the number of iteration.

(3) Multigrid together with multiresolution has a very strong impact on convergency for high frequency texture and large parallax range. As we have shown in the previous two subsections, both multigrid and multiresolution alone show improvement in efficiency in images with poor initial value or large parallax range and high frequency texture, but by combining the multigrid and multiresolution, the improvement is extremely high (see Table 3, example *Bb*, *Cb*, and *Cc*). These three examples are matched, as are all other examples, without any problem while, in the two previous subsections, these three examples need more iterations than all other examples.

(4) The reliability is improved with the combination of multigrid and multiresolution compared with single level matching (see example *Bb* and *C*). This is similar to the multigrid and multiresolution alone.

DISCUSSION AND CONCLUSIONS

Multigrid and multiresolution structures for multipoint matching have been developed and investigated. They have also been tested on a variety of images. Results are encouraging and they show that the hierarchical structures have advantages with respect to computational efficiency and reliability over single level multipoint matching, especially the combination of multigrid and multiresolution. In a sense, this hierarchy also improves the pull-in range of the matching, in particular for images with high frequency texture. This is similar to the frequency related average (FRA) approach by Korten *et al.* (1988). The hierarchical structures also improve the reliability of the matching in such cases, as has been shown by examples.

The multigrid makes use of the structure of the matching problem, i.e., regular grid spaced points[2], while the multiresolution makes use of the image resolution. Both structures can been viewed from the *scale-space* point of view (see, e.g., Witkin (1983), Lindeberg (1990), and Lindeberg and Eklundh (1990)). Objects and image structures have scales. At large (course) scale, dominant objects or image structure can be detected more easily, while at finer scale more detailed objects or image structure can be detected. The multigrid structure of the matching points

[2]Note here we do not use the multigrid algorithm, but rather, the structure of multigrid. So it may be better to say that the multigrid structure is a multiresolution representation of the geometric object surface. See the late discussion of scale-space.

TABLE 3. COMPUTATIONAL WORK OF MULTIPOINT MATCHING WITH MULTIGRID AND MULTIRESOLUTION. SEE TEXT NOTE IN TABLE 1.

data	level 1 3 × 3		level 2 5 × 5		level 3 9 × 9		level 4 17 × 17		total		one-level 17 × 17			red.
set	it	CPU	it	CPU	it	CPU	it	CPU	CPU	s/p	it.	CPU	s/p	(%)
Aa	5	3″	6	8″	4	22″	4	2′30″	3′33″	0.73″	17	6′19″	1.31″	42.28
Ab	5	4″	3	6″	5	32″	4	2′46″	3′58″	0.82″	20	6′51″	1.42″	42.25
Ac	5	3″	6	8″	7	33″	5	2′41″	3′55″	0.81″	15	5′42″	1.18″	31.36
Ba	8	3″	4	6″	3	18″	3	2′14″	3′11″	0.66″	9	4′21″	0.90″	26.67
Bb	8	4″	6	8″	3	19″	3	2′16″	3′17″	0.68″	247	62′28″	12.97″	94.76
Bc	5	3″	5	7″	3	19″	3	2′16″	3′15″	0.68″	19	6′44″	1.40″	51.43
Ca	5	2″	4	5″	2	16″	2	2′00″	2′26″	0.51″	20	4′50″	1.42″	64.08
Cb	7	3″	6	7″	4	22″	3	2′12″	2′45″	0.57″	247	62′04″	12.90″	95.58
Cc	5	2″	4	6″	5	27″	4	2′28″	3′03″	0.65″	167	42′46″	8.9″	92.70

is a scale-space representation of the geometric surface of the object. Multiresolution of image is a scale-space representation of image gray values. In both scale-space from course to fine, we treat dominant objects first, and then use this information as constraints to guide the matching of fine detailed objects.

The concepts of multigrid and multiresolution and their applications in image analysis are not new to us, but the combination of them developed in this paper is a new approach. Multigrid methods have been used in image analysis by Terzopoulos (1986). The use of multiresolution structure in image analysis is too numerous to list here. Our approach, in general, is an optimization problem combining similarity measure and smoothness constraints. The hierarchy is formulated in *double-scale-space*, i.e., in geometrical surface scale-space and image resolution scale-space. Some other similar approaches, but not quite the same, can also be found (see, for example, Poggio *et al.* (1985)). Poggio (1985) proposed an approach for stereo matching utilizing *regularization*. This algorithm has been implemented by March (1988). In this approach, the similarity is as a penalty functional and the smoothness is as a stabilizing functional in the context of regularization. Horn (1986) also formulates the matching problem in a very similar way. Both these approaches have not been implemented in scale-space. Witkin *et al.* (1987) also formulated the problem as an optimization of energy functional. Here the energy functional consists of two parts: similarity functional, which is a normalized cross-correlation, and smoothness functional. They solve the problem by continuation method over scale-space to avoid local minimum. Barnard (1989) described another approach – stochastic optimization. There the problem is posed as computational analogy to a thermodynamic physical system, i.e., simulated annealing. Barnard used hierarchy in image resolution scale-space.

The structures and algorithm developed in this investigation have some generalities and can be generalized to many other image analysis problems, where the problem is formulated in such a way that the points in question are located in a regular grid form. For example, to the object space correlation problem by, e.g., Wrobel (1987, 1988), Helava (1988), and Ebner *et al.*, (1988). Here the matching is performed in object space directly on two surfaces, the geometric surface and the radiometric surface. The surfaces are reconstructed discretely on regular grid points over a large area. This is similar to the multipoint matching but in object space and with two surfaces. The multigrid and multiresolution structures as well as the combination of them can be straightforwardly applied to the problem.

The currently obtained results are encouraging, but there are still possibilities to improve the strength and efficiency of the method. As we pointed out earlier, we did not make full use of the multigrid methods in this investigation due to the characteristics of our problem. A further consideration would be the full use of the multigrid method in the matching problem. For this, matching through regularization (see, e.g., Horn (1986) and March (1988)) is a typical example, where the multigrid could be fully exploited to achieve high computational efficiency. This high efficiency has been demonstrated by Terzopoulos (1986) by applying the multigrid method to other similar image analysis problems than matching.

So far we have not given much consideration to the problem of computational efficiency of matrix computation, in particular the matrix inverse, which is a considerable part of the computation. In this aspect, the array algebra techniques developed by Rauhala (1987) are very attractive. This technique subdivides a large matrix into several small submatrices, and then inverts them. As the computational operations of inverting vary as $O(n^3)$ (n is is the dimension of the matrix), this technique can reduce computational work incredibly. How this technique can be used in the multipoint matching case, or in general in image matching, needs to be investigated.

The main conclusions we can draw from this study are that the multipoint matching formulated in multigrid and multiresolution structure is a very efficient approach for stereo matching, in particular, for surface reconstruction and DTM generation. The hierarchy improves efficiency dramatically. It also improves reliability of the matching compared with single level matching. Our method has high flexibility and generality and can be applied to many other similar problems in photogrammetry and computer vision.

ACKNOWLEDGMENTS

The work reported in this paper was done when the author was at the Department of Photogrammetry, while the report was completed when the author was with the Computational Vision and Active Perception (CVAP) laboratory, both at the Royal Institute of Technology in Stockholm. The author wishes to thank all his colleagues at the Department of Photogrammetry, and especially, he wishes to thank Prof. Kennert Torlegård for his enlightening discussions and encouragement.

REFERENCES

Ackermann, F., 1984. Digital Image Correlation: Performance and Potential Application in Photogrammetry. *Photogrammetric Record*, 11(64), pp. 429–439.

Barnard, S. T., 1989. Stochastic Stereo Matching over Scale. *International Journal of Computer Vision*. 3, pp. 17–32.

Bertero, M., T. Poggio, and Torre, V., 1988. Ill-Posed Problems in Early Vision. *Proceedings of the IEEE*, Vol. 76, No. 8, pp. 869–889.

Brandt, A., 1982. Guide to Multigrid Development. *Multigrid Methods* (Hackbusch and Trottenberg, eds.), Springer-Verlag, pp. 220–312.

Burt, P. J. and E. H. Adelson, 1983. The Laplacian Pyramid as a Compact Image Code. *IEEE Transaction on Communications*, Vol. COM-31, No. 4, pp. 532–540.

Crowley, J. L. and R. M. Stern, 1984. Fast computation of the Difference of Low-Pass Transform. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-6, No. 2, pp. 212–222.

Ebner, H., and D. Fritsch, 1986. The Multigrid Method and its Application in Photogrammetry. *International Archives for Photogrammetry and Remote Sensing*, Vol. 26, Com. III, Rovaniemi.

Ebner, H., and C. Heipke, 1988. Integration of Digital Image Matching and Object Surface Reconstruction. *International Archives for Photogrammetry and Remote Sensing*, Vol. 27, Com. III, Kyoto, pp. B11.III.534–545.

Förstner, W., 1982. On the Geometric Precision of Digital Image Correlation. *International Archives for Photogrammetry and Remote Sensing*, Vol. 24, Com. III, Helsinki, pp. 176–189.

Grün, A., 1985. Adaptive Least Squares Correlation: A Powerful Image Matching Technique. *South African Journal of Photogrammetry, Remote Sensing and Cartography*, 14(3), pp. 175–187.

Hackbusch, W., 1985. *Multigrid Methods and Applications*. Berlin, Spring-Verlag.

Helava, U. V., 1988. Object Space Least Squares Correlation. *International Archives for Photogrammetry and Remote Sensing*. Vol. 27, Com. III, Kyoto, ppB3.321–331.

Horn, B. K. P., 1986. *Robot Vision*. MIT Press, Cambridge, Massachusetts.

Korten, Th., B. Wrobel, M. Franek, and M. Weisensee, 1988. Experiments with Facets Stereo Vision (FAST-Vision) for Object Surface Computation. *International Archives for Photogrammetry and Remote Sensing*, Vol. 27 Com. III, Kyoto, pp.B3.396–404.

Kölbl, O. and I. Silva, 1988. Derivation of a Digital Terrain Model by Dynamic Programming. *International Archives for Photogrammetry and Remote Sensing*, Vol. 27 Com. III, Kyoto, pp. III.77–86.

Li, M., 1989. *Hierarchical Multi-Point Matching and Simultaneous Detection and Location of Breaklines*. Doctorial Dissertation. Department of Photogrammetry, The Royal Institute of Technology, Stockholm.

Lindeberg, T., 1990. Scale-Space for Discrete Signals. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-12, No. 3, pp. 230–254.

Lindeberg, T., and J. O. Eklundh, 1990. *Guiding Early Visual Processing with a Scale-Space Primal Sketch*. Tech. Report TRITA-NA-P9013, Dept. of Numerical Analysis and Computing Science, The Royal Institute of Technology, Stockholm. Submitted. A short version also in *Proc. 3rd ICCV*, Osaka, Japan, 1990.

March, R., 1988. Computation of Stereo Disparity Using Regularization. *Pattern Recognition Letters*, 8, Oct., pp. 181–187.

Meer, P., E.S. Baugher, and A. Rosenfeld, 1987. Frequency Domain Analysis and Synthesis of Image Pyramid Generating Kernels. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-9, No. 4, pp.512–522.

Poggio, T., 1985. Early Vision: From Computational Structure to Algorithms and Parallel Hardware. *Computer Vision, Graphics and Image Processing*, 31, pp. 139–155.

Poggio, T., V. Torre, and C. Koch, 1985. Computational Vision and Regularization Theory. *Nature*, Vol. 317, 26.

Rauhala, U.A., 1987. Fast Compiler Positioning Algorithms and Techniques of Array Algebra in Analytical and Digital Photogrammetry. *Proceedings of Intercommission conference of ISPRS on Fast Processing of Photogrammetric Data*, Interlaken.

Rosenfeld, A., 1983. Pyramids: Multiresolution Image Analysis. *Proceedings of the Third Scandinavian Conference on Image Analysis*, Copenhagen, pp. 23–28.

Rosenholm, D., 1986a. Accuracy Improvement in Digital Matching. *Fotogrammetriska Meddelanden*, No. 52, Dept. of Photogrammetry, the Royal Institute of Technology, Stockholm.

———, 1986b. Accuracy Improvement of Digital Matching for Evaluation of Digital Terrain Models. *International Archives for Photogrammetry and Remote Sensing*, Vol. 26, Com. III, Rovaniemi, pp. 573–587.

———, 1987. Multi-point Matching Using the Least Squares Technique for Evaluation of Three-Dimensional Models. *Photogrammetric Engineering & Remote Sensing*, Vol. 53, No. 6, pp. 621–626.

———, 1988. Multi-point Matching Along Vertical Lines in SPOT Images. *International Journal of Remote Sensing*, Vol. 9, Nos. 10 & 11, pp. 1687–1703.

Stüben, K., and U. Trottenberg, 1982. Multigrid Method: Fundamental Algorithms, Model Problem Analysis and Applications. *Multigrid Method* (Hackbusch and Trottenberg, eds.), Springer-Verlag, pp. 1–176.

Terzopoulos, D., 1983. Multilevel Computational Processes for Visual Surface Reconstruction. *Computer Vision, Graphics, and Image Processing*, 24, pp. 52–96.

———, 1984. Multilevel Reconstruction of Visual Surfaces: Variational Principles and Finite-Element Representations. *Multiresolution Image Processing and Analysis* (A. Rosenfeld ed.), Springer-Verlag, Berlin, pp. 237–310.

———, 1986. Image Analysis Using Multigrid Relaxation Methods. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-8, No.2, pp. 129–139.

Witkin, A., 1983. Scale Space Filtering. *Proceedings of the International Joint Conference on Artificial Intelligence*, Karlsruhe, pp. 1019–1021.

Witkin A., D. Terzopoulos, and M. Kass, 1987. Signal Matching Through Scale Space. *International Journal of Computer Vision*, pp. 133–144.

Wong, R. Y., and E. L. Hall, 1978. Sequential Hierarchical Scene Matching. *IEEE Transactions on Computers*, Vol. C-27, No. 4, pp. 359–366.

Wrobel, B., 1987. Facets Stereo Vision (FAST Vision) – A New Approach to Computer Vision and to Digital Photogrammetry. *Proceedings of Intercommission conference of ISPRS on Fast Processing of Photogrammetric Data*, Interlaken.

———, 1988. Least Squares Methods for Surface Reconstruction from Images. *International Archives for Photogrammetry and Remote Sensing*, Vol. 27 Com. III, Kyoto, pp. B3 806–821.

(Received 10 August 1990; accepted 13 December 1990; revised 15 February 1991)