PARALLEL APPROACH TO BINOCULAR STEREO MATCHING

Herbert Jahn
DLR, Institute of Space Sensor Technology and Planetary Exploration, Berlin, Germany
Herbert.Jahn@dlr.de

Commission III, WG III/8

KEY WORDS: Vision Sciences, Stereoscopic Matching, Real-time Processing, Dynamic Networks

ABSTRACT:

An approach for parallel-sequential binocular stereo matching is presented. It is based on discrete dynamical models which can be implemented in neural multi-layer networks. It is based on the idea that some features (edges) in the left image exert forces on similar features in the right image in order to attract them. Each feature point (i,j) of the right image is described by a coordinate x(i,j). The coordinates obey a system of time discrete Newtonian equations of motion, which allow the recursive updating of the coordinates until they match the corresponding points in the left image. That model is very flexible. It allows shift, expansion and compression of image regions of the right image, and it takes into account occlusion to a certain amount. To obtain good results a robust and efficient edge detection filter is necessary. It relies on a non-linear averaging algorithm which also can be implemented using discrete dynamical models. Both networks use processing elements (neurons) of different kind, i.e. the processing function is not given a priori but derived from the models. This is justified by the fact that in the visual system of mammals (humans) a variety of different neurons adapted to specific tasks exist. A few examples show that the problem of edge preserving smoothing can be solved with a quality which is sufficient for many applications (various images not shown here have been processed with good success). A certain success was also achieved in the main problem of stereo matching but further improvements are necessary.

1. INTRODUCTION

Real-time stereo processing which is necessary in many applications needs very fast algorithms and processing hardware. The stereo processing capability of the human visual system together with the parallel-sequential neural network structures of the brain (Hubel, 1995) lead to the conjecture that there exist parallel-sequential algorithms which do the job very efficiently. Therefore, it seems to be natural to concentrate effort to the development of such algorithms.

In prior attempts to develop parallel-sequential matching algorithms (Jahn, 2000a; Jahn, 2000b) some promising results have been obtained. But in some image regions serious errors occurred which have led to a new attempt to be presented here.

If one de-aligns both our eyes by pressing one eye with the thumb then one has the impression, as if one of the images is pulled to the other until matching is achieved.

This has led to the idea that prominent features (especially edge elements) of one image exert forces to corresponding features in the other image in order to attract them. A (homogeneous) region between such features is shifted together with the region bounding features whereas it can be compressed or stretched, because corresponding regions may have different extensions. Therefore, an adequate model for the matching process seems to be a system of Newtonian equations of motion governing the shift of the pixels of one image. Assuming epipolar geometry a pixel (i,j) of the left image corresponds to a pixel (i',j) of the right image of the same image row. If a mass point with coordinate $x(i',j)$ and mass m is assigned to that pixel then with appropriate forces of various origins acting on that point it can be shifted to match the corresponding point (i,j). To match points inside homogeneous regions, the idea is to couple neighboured points by springs in order to shift these points together with the edge points. The model then resembles a little bit the old model of Julesz which he proposed in (Julesz, 1971) for stereo matching.

To obtain good results a robust and efficient edge detection filter is necessary. The filter used here is based on a non-linear edge preserving smoothing algorithm which can be implemented with the same type of parallel-sequential networks, the so-called discrete dynamical networks (Serra, Zanarini, 1990) which can be described (in 2D notation) by

$$\mathbf{z}_{i,j}(t+1) = \mathbf{f}_{i,j}\left(\mathbf{z}(t), \mathbf{P}, \mathbf{K}_{i,j}(t)\right) \qquad (i = 1,\ldots,N_1, \quad j = 1,\ldots,N_2) \tag{1}$$

Here, $\mathbf{z}_{i,j}$ is a state vector defined in each image point (i,j) ($\mathbf{z}(t)$ denotes the matrix of the $\mathbf{z}_{i,j}(t)$), $\mathbf{K}$ is an external force vector, and $\mathbf{P}$ is a parameter vector. The initial state $\mathbf{z}_{i,j}(0)$ is given by a feature vector which is derived from the given image data. Then, according to (1), the feature vector is updated recursively leading to a final state (hopefully a fix point) at $t \to \infty$ (or approximately at $t = t_{max}$). That final state is the result of the image processing task.

The algorithm (1) is of complexity O(N) ($N = N_1 \cdot N_2$) if the number of iterations is limited (= t_{max}). In each iteration step it needs a constant number n of calculations for every image point (i,j). Then the total number of operations is $N \cdot n \cdot t_{max}$. Therefore, it is very fast if it is implemented in a multi-layer network structure. Here, each neural layer is assigned to a discrete time t of (1), and the state of neuron (i,j) in layer t is given by $\mathbf{z}_{i,j}(t)$. Via the (nonlinear) function $\mathbf{f}_{i,j}$ each neuron (i,j) of layer t+1 is coupled with neurons (k,l) of layer t.

In chapter 2 algorithm (1) is specified to edge preserving smoothing. Then, chapter 3 is dedicated to stereo matching within the same framework. Some results are shown. Finally, in the conclusions some ideas for future research are presented.

2. EDGE PRESERVING SMOOTHING

Edge preserving smoothing is a pre-processing step which is often necessary in order to alleviate or even to make possible the following steps of stereo processing, object recognition etc..
In the past a huge amount of methods for edge preserving smoothing have been developed (see e.g. (Klette, Zamperoni, 1992)) but here a method is presented which fits the discrete dynamical network (1).
That method (Jahn, 1999a) which has a certain relation to the anisotropic diffusion approach (Perona, Shiota, Malik, 1994) is more general than edge preserving smoothing but here it is applied only to that special problem. We consider M points $P_k = (x_k,y_k)$ (k=1,...,M). These points are the pixel positions (i,j) in case of edge preserving smoothing. We now assign to each point P_k the points P_l of the Voronoi neighbourhood $N_V(P_k)$ which is the 4- neighbourhood in case of raster image processing. For simplifying, the notation N(k) instead of $N_V(P_k)$ is used in the following. Furthermore, to each point P_k a feature vector $\mathbf{f}_k$ is assigned (in case of edge preserving smoothing the (scalar) features are the grey values $g_{i,j}$).
To derive a feature smoothing algorithm the feature vector $\mathbf{f}_k$ is averaged over the neighbourhood N(k):

$$\langle \mathbf{f}_k \rangle = \frac{1}{n_k + 1} \cdot \left(\mathbf{f}_k + \sum_{k' \in N(k)} \mathbf{f}_{k'} \right) \tag{2}$$

Here, n_k is the number of Voronoi neighbours of point P_k (n_k = 4 in case of raster image processing).
An equivalent (recursively written) notation of (2) is

$$\mathbf{f}_k(t+1) = \mathbf{f}_k(t) + \frac{1}{n_k + 1} \cdot \sum_{k' \in N(k)} [\mathbf{f}_{k'}(t) - \mathbf{f}_k(t)] \tag{3}$$
$$(t = 0, 1, 2, \ldots)$$

The initial condition is $\mathbf{f_k}(0) = \mathbf{f}_k$ ($=g_{i,j}$).
Because of its linearity, the recursive algorithm (3) with increasing recursion level (or discrete time) t diminishes the resolution of the image and blurs the edges more and more. But here we do not want to blur edges and to smooth out image details. Therefore, the feature differences in (3) must be weighted properly to prevent that. Introducing weights $w_{k,k'}$ the following scheme is obtained:

$$\mathbf{f}_k(t+1) = \mathbf{f}_k(t) + \frac{1}{n_k + 1} \cdot \sum_{k' \in N(k)} w_{k,k'}(t) \cdot [\mathbf{f}_{k'}(t) - \mathbf{f}_k(t)] \tag{4}$$

The weight $w_{k,k'}$ is chosen as a function of the edge strength between features $\mathbf{f}_k$ and $\mathbf{f}_{k'}$. Averaging of both features is only possible if the edge strength is weak. To choose the weights the edge strength is introduced according to

$$x_{k,k'} = \frac{t_{k,k'}}{\|\mathbf{f}_k - \mathbf{f}_{k'}\|} \tag{5}$$

In (5) $\|\mathbf{f}\|$ is the norm of the vector $\mathbf{f}$ ($\|\mathbf{f}\| = |f|$ in case of an 1D feature f), and $t_{k,k'}$ is an (adaptive) threshold.
Now, the weights can be introduced via

$$w_{k,k'} = s(x_{k,k'}) \tag{6}$$

where s(x) is a non-increasing function with s(0) = 1 and s(∞) = 0. Good results are obtained with the function

$$s(x) = \frac{1}{1 + x^2} \tag{7}$$

but other functions are possible too.
The algorithm (4) is of type (1) and thus represents a special discrete dynamical network. We learn from (4) that in contrast to commonly used neural networks not the signals $\mathbf{f}_k$ are weighted and summed but their differences $\mathbf{f}_k$ - $\mathbf{f}_{k'}$ of neighboured neurons. Furthermore, the non-linearity, here given by the function s(x) (7), differs from the sigmoid function.
Figures 1 to 3 show the capabilities of the algorithm.
In algorithm (4) the averaging was confined to the (small) 4-neighbourhood. Therefore, many iterations (typically 20 – 30) are necessary to obtain sufficient smoothing. To reduce the number of iterations bigger neighbourhoods can be considered (Jahn, 1999b).

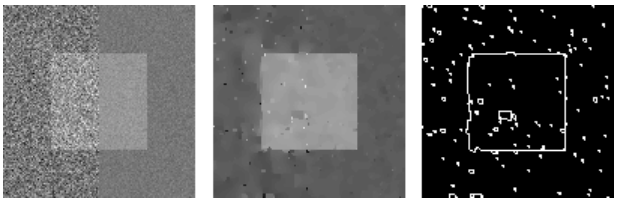

Fig. 1. left: simulated image with additive noise (S/N = 1 and S/N = 4, resp.)
center: smoothed image
right: edges in smoothed image

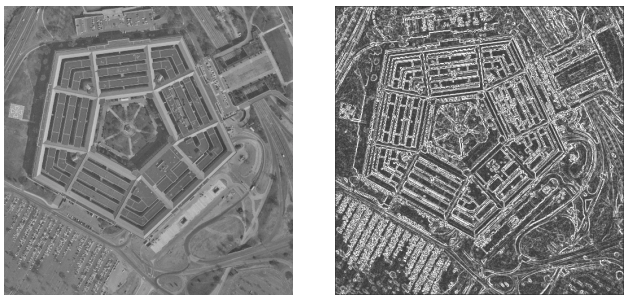

Fig. 2. Pentagon image — edge image

Fig. 3. smoothed image — edge image

3. STEREO MATCHING

Stereo matching (Klette et al., 1998) is an important problem with a broad range of applications. Considerable efforts have been made to enhance the matching quality and to reduce the processing time. Fast algorithms (Gimel'farb, 1999) and also a few approaches to parallel and neural algorithms, e.g. (Goulermas, Liatsis, 2000), (Pajares, de la Cruz, 2001), exist for the case of epipolar geometry but these geometry often is fulfilled only approximately. Therefore, methods are needed which can be generalized to the non-epipolar case and which have real-time processing capability. Here an attempt is made to develop such a method basing on the discrete dynamical networks (1). To start that research again epipolar geometry is used but it is obvious how the algorithm can be generalized to the non-epipolar case.

We consider a left image g_L(i,j) and a right image g_R(i,j) (i,j = 0,…,N-1). Corresponding points (i,j) of g_L and (i',j) of g_R on epipolar lines j are connected by $i' = i+s$ where $s(i',j)$ is the disparity. Here, $s(i',j)$ is assigned to the coordinates of the right image, but it is also possible to assign it to the left image or to a centered (cyclopean) image. In corresponding points the following equation approximately holds:

$$g_L(i,j) \approx g_R(i+s,j) \tag{8}$$

In most images there are points which are absent in the other image (occluded points). Those points (for which (1) of course does not hold) must be considered carefully in order to avoid mismatch.

Now, to each pixel (i',j) of the right image a coordinate $x(i',j)$, a velocity $v(i',j)$, and a mass m are assigned in order to describe the motion of such a point. Let (i_e,j) be an edge point of the left image and (i_e',j) an edge point of the right image, respectively. Then, the edge point in the left image exerts a force $K(i_e, i_e',j)$ on the edge point in the right image in order to attract that point. We consider that force as an external force. Furthermore, on mass point (i',j) there can be acting internal forces such as the spring type forces $K_{spring}(i'-1, i',j)$, $K_{spring}(i'+1, i',j)$ and a force $-\gamma v(i',j)$ describing the friction with the background. Other internal forces such as friction between neighboring image rows j, $j\pm1$ can also be included. More general, the forces, here denoted as $K(i',j, j\pm k)$, can also depend on edges and grey values in other image rows $j\pm k$ of both images which means a coupling of different image rows.

With those forces Newton's equations are:

$$m \cdot \ddot{x}_t(i',j) = -\gamma \cdot \dot{x}_t(i',j) + K_t(i',j,j\pm k) \tag{9}$$

Introducing the velocities $v = \dot{x}$, the system of differential equations (9) of second order can be converted into a system of first order equations

$$\begin{aligned} \dot{x}_t(i',j) &= v_t(i',j) \\ m \cdot \dot{v}_t(i',j) &= -\gamma \cdot v_t(i',j) + K_t(i',j,j\pm k) \end{aligned} \tag{10}$$

Here, $\mathbf{z} = \begin{pmatrix} x \\ v \end{pmatrix}$ is the state vector of the system.

Now, approximating $\dot{\mathbf{z}}_t$ by $\frac{\mathbf{z}_{t+\Delta t} - \mathbf{z}_t}{\Delta t}$, the system of differential equations turns into a system of difference equations or discrete time state equations:

$$\begin{aligned} x_{t+\Delta t}(i',j) &= x_t(i',j) + \Delta t \cdot v_t(i',j) \\ m \cdot v_{t+\Delta t}(i',j) &= (m - \gamma \cdot \Delta t) \cdot v_t(i',j) + K_t(i',j,j\pm k) \end{aligned} \tag{11}$$

That system which is of type (1) allows to calculate the system state $\mathbf{z_t}$ recursively. The initial conditions are:

$$\begin{aligned} x_0(i',j) &= i' \\ v_0(i',j) &= 0 \end{aligned} \tag{12}$$

When that recursive system of equations has reached its final state $x_{t_{\max}}(i',j)$, then the disparity can be calculated according to

$$s(i',j) = i' - x_{t_{\max}}(i',j) \tag{13}$$

which is the final shift of $x_t(i',j)$ from its initial position $x_0(i',j) = i'$.

The recursive calculation of the disparity according to (11) allows the incorporation of some countermeasures against ambiguities. In particular, the so-called ordering constraint (Klette et al., 1998) can be included: Let

$$\Delta_t x(i',j) = \Delta t \cdot v_t(i',j) \tag{14}$$

be the increment of $x_t(i',j)$. Then, the initial order $x_t(i'+1,j) > x_t(i',j)$ of the pixels of the right image can be guaranteed if the following limitation of $\Delta x_t(i',j)$ is used:

$$\Delta x_t(i',j) = \begin{cases} d_+/2 & \text{if} \quad \Delta t \cdot v_t(i',j) > d_+/2 \\ -d_-/2 & \text{if} \quad \Delta t \cdot v_t(i',j) < -d_-/2 \\ \Delta t \cdot v_t(i',j) & \text{elsewhere} \end{cases} \tag{15}$$

Here, $d_+ = x_t(i'+1,j) - x_t(i',j)$, $d_- = x_t(i',j) - x_t(i'-1,j)$.

Conditions such as (15) can be checked easily in each step of recursion.

We come now to the calculation of the forces K. First, it must be acknowledged that essential stereo information is only present in image regions with significant changes of grey level, and especially near edges. Furthermore, in the epipolar geometry assumed here only the x – dependence of the grey values, i. e. $\nabla_x g(i,j) = g(i,j) - g(i-1,j)$ is essential. Let's assume that there is a step edge between (i,j) and (i-1,j) with $|\nabla_x g(i,j)| > threshold$. That means that ($i,j$) belongs to an image segment and (i-1,j) to another one. When e. g. such an edge is at the border of a roof of a building then often left or right of that edge we have occlusion. Then, if the pixel (i,j) has a corresponding pixel in the other stereo image this may be not the case for pixel (i-1,j) or vice versa. Therefore, both pixels (i.e. pixels left hand and right hand of an edge) must be considered separately. They can have different disparities or even worse: in one of them

cannot be calculated a disparity at all. To such pixels only with prior information or by some kind of interpolation a (often inaccurate) disparity can be assigned. With respect to our attracting forces that means the following: If there is an edge in the left image between (i-1,j) and (i,j) and another one between (i'-1,j) and (i',j) in the right image then there is a force $K_R(i,i',j)$ originating from (i,j) and attracting (i',j) and another force $K_L(i\text{-}1,i'\text{-}1,j)$ acting from (i-1,j) to (i'-1,j). This is necessary for coping with occlusion.

Let's consider the external force $K_R(i,i',j)$. Then, first, that force depends on the difference $|g_L(i,j)\text{-}g_R(i',j)|$ or, more general, on a certain mean value of that difference. That mean value should be calculated only over pixels which are in the same image regions as pixels (i,j) (in left image) and (i',j) (in right image) in order to exclude problems with occlusion. To guarantee this, the averaging is performed only over image points (i_k,j_k) with $|g_L(i,j)\text{-}g_L(i_k,j_k)| \le threshold$ and (i'_l,j_l) with $|g_R(i',j)\text{-}g_R(i'_l,j_l)| \le threshold$, respectively. We denote that mean value as

$$\Delta g(i,i';j) = \left\langle \left| g_L(i,j) - g_R(i',j) \right| \right\rangle \qquad (16)$$

Secondly, pure radiometric criteria are not sufficient. Therefore, geometric deviations are taken into account too. To do that, we consider two region border lines (one in the left image and the other in the right image) which contain the points (i,j) and (i',j), respectively. The situation is shown in figure 4.

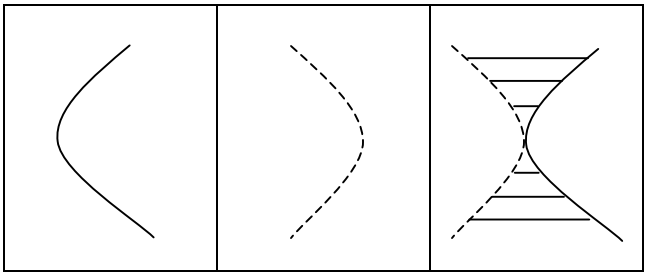

Fig. 4. Borderlines in left image, in right image, and overlaid

We see that both borderlines are different and do not match. A useful quantity for measuring that mismatch is the sum of the border point distances along the horizontal lines drawn in figure 4. Be $(i_k,j+k)$ a point on the left borderline and $(i'_k,j+k)$ a point on the right borderline, respectively. For $k = 0$ the points are identical with the points (i,j) and (i',j). Be $d_o = |i - i'|$. Then, a useful border point distance is

$$d_w(i,i';j) = \sum_{k=-w}^{w} \left(\left| i_k - i_k' \right| - d_0 \right) \qquad (17)$$

The distance d_w accomplishes a certain coupling between epipolar image rows which are no longer independent. This sometimes can reduce mismatches efficiently.

With Δg and d_w the total distance is

$$d(i,i';j) = \alpha_1 \cdot d_w(i,i';j) + \alpha_2 \cdot \Delta g(i,i';j) \qquad (18)$$

The smaller that distance between edge points (i,j) and (i',j) is, the bigger is the force $K_R(i,i',j)$. Therefore,

$$K_R(i,i';j) \propto \exp\left[-d(i,i';j) \right] \qquad (19)$$

seems to be a good measure for the force K_R. The calculation of K_L is fulfilled analogously.

Now, the (external) force $K_{ext}(i',j)$ acting on point (i',j) can be computed as the maximum of all forces $K_R(i,i',j)$ with different i or as a weighted sum of these forces. Here, we take into account only points (i,j) with $|\ i - i'| \le Max\_disparity$. The maximum disparity used here is often known a priori. The introduction of *Max_disparity* is not necessary. One can also use distance depending weighting and calculate the resulting force as

$$K_{R,ext,t}(i',j) = \sum_i K_R(i,i',j) \cdot f\left(\left| i - x_t(i',j) \right| \right) \cdot sign\left(i - x_t(i',j) \right) \qquad (20)$$

with $f(|\ i - x_t(i',j)|)$ being a certain weighting function which decreases with increasing distance $|\ i - x_t(i',j)|$. Here, we use the special function

$$f(x) = \begin{cases} |x| & if \quad |x| \le Max\_disparity \\ 0 & elsewhere \end{cases} \qquad (21)$$

Up to now we have considered only forces which act only on image points (i',j) near edges. But we must assign a disparity to each point of the right image. Therefore, the disparity information from the edges must be transfered into the image regions. Within the model presented here, it is useful to do this by means of adequate forces, which connect the edge points with interior points (i.e. points inside regions). Local forces of spring type have been studied for that purpose. Let $x_t(i',j)$ and $x_t(i'+1,j)$ be two neighboured mass points which we assume to be connected by a spring. Then, point $x_t(i'+1,j)$ exerts the following (attracting or repulsive) force on point $x_t(i',j)$:

$$K_{spring,t}(i'+1,i',j) = \kappa \cdot \left[x_t(i'+1,j) - x_t(i',j) - 1 \right] \qquad (22)$$

The same force, but with the opposite sign, acts from $x_t(i',j)$ on $x_t(i'+1,j)$ according to Newton's law of action and reaction.

Experiments with those and other local forces (e.g. internal friction) have not been fully satisfying up to now. Of course, the stereo information is transferred from the edges into the regions, but very slowly. One needs too many recursions until convergence. Therefore, one result of these investigations is that local forces are not sufficient. We need far-field interaction between points $x_t(i',j)$ and $x_t(i'+k,j)$ which can easily be introduced into our equations of motion. First experiments with such forces have given some promising results but that must be studied more detailed in future.

The algorithm is applied here to the standard Pentagon stereo pair because that image pair is a big challenge because of the many similar structures and the many occlusions. Figure 5 shows a section of the smoothed image pair (see figure 3 for the whole left-hand image).

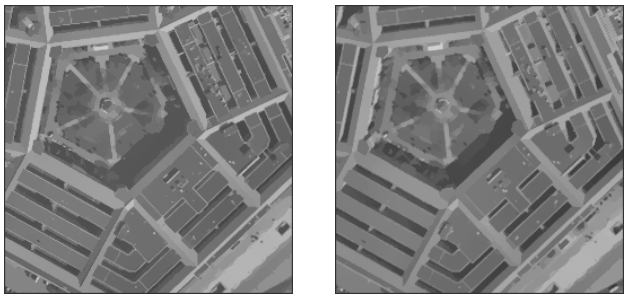

Fig. 5. Binocular image pair

Several computer experiments have been carried out in order to choose proper parameters of the algorithm. It turned out that the number w in (17) should have a size between 5 and 10 (but of course, that value depends on the structure of the image) whereas the best values of parameters α_1 and α_2 of equation (18) seem to be in the vicinity of $\alpha_1 = \alpha_2 = 1$. But these investigations are only preliminary. It is also not clear which the best law of the external forces (19) – (21) is.

Red-green overlays of left images (red) and right edge images (green) showing the inherent disparities and the quality of edge matching, respectively, are presented in figures 6 and 7. Finally, figure 8 shows the disparity image. The result shows that in most image points good matching was achieved. But it shows also that there are left some mismatches resulting in wrong disparities.

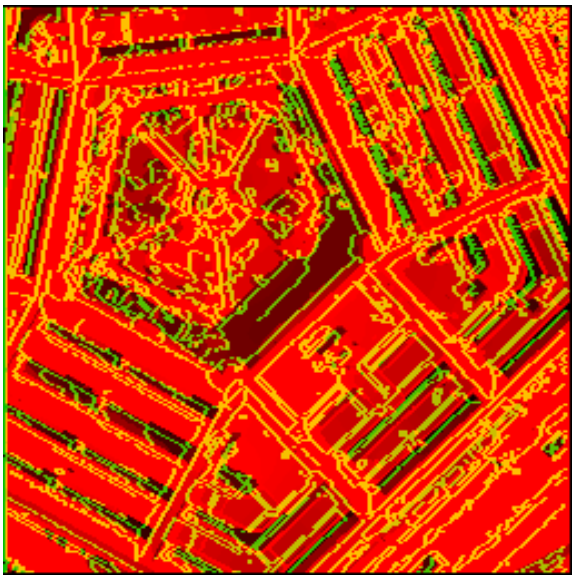

Fig. 6. Overlay of left images and right edge images (original image pair)

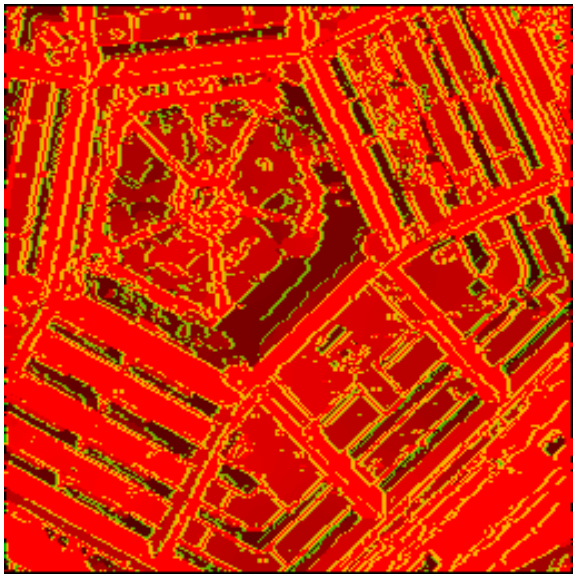

Fig. 7. Overlay of left images and right edge images (image pair after 20 iterations)

Fig. 8. Disparity map

4. CONCLUSIONS

The results show that the introduced parallel-sequential model based on Newton's equations of motion and attracting forces between edges may be a promising approach to real-time stereo processing. The algorithm gives the right disparities in most image points but there remain errors. Therefore, new efforts are necessary to enhance the quality of the approach. Some ideas for improvement are the following: First, far - field forces should be introduced. Secondly, the assumed force law (16) – (21) must be optimized or changed. When the right position $x_{tmax}(i',j)$ is reached then the external forces should reduce to zero in order to avoid oscillations which are small but not zero now.
Finally, it must be mentioned that the model can be extended to the non-epipolar case introducing coordinates $y(i',j)$ and forces acting in y – direction.

References from Journals:

Goulermas, J.Y., Liatsis, P., 2000. A new parallel feature-based stereo-matching algorithm with figural continuity preservation, based on hybrid symbiotic genetic algorithms. Pattern Recognition **33**, pp. 529-531

Pajares, G., de la Cruz, J. M., 2001. Local stereo vision matching through the ADALINE neural network. Pattern Recognition Letters **22**, pp. 1457-1473.

References from Books:

Hubel, D., 1995. *Eye, Brain, and Vision*. Scientific American Library, New York.

Julesz, B., 1971. *Foundations of Cyclopean Perception*. The University of Chicago Press, Chicago, pp. 203-215.

Klette, R., Schlüns, K., Koschan, A., 1998. *Computer Vision*. Springer, Singapore.

Klette, R., Zamperoni, P., 1992. *Handbuch der Operatoren für die Bildverarbeitung*, Vieweg, Braunschweig.

Serra, R., Zanarini, G., 1990. *Complex Systems and Cognitive Processes*. Springer, Berlin.

References from Other Literature:

Gimel'farb, G., 1999. Stereo Terrain Reconstruction by Dynamic Programming. In: *Handbook of Computer Vision and Applications*, Vol. 2, Academic Press, San Diego, pp. 505-530.

Jahn, H., 1999a. Feature Grouping Based on Graphs and Neural Networks. In: *Lecture Notes in Computer Science* **1689**, Springer, Berlin, pp. 568-577.

Jahn, H., 1999b. Unsupervised Learning of Local Mean Gray Values for Image Pre-processing. In: *Lecture Notes in Artificial Intelligence* **1715**, Springer, Berlin, pp. 64 – 74.

Jahn, H., 2000a. Stereo Matching for Pushbroom Stereo Cameras. In: *Int. Archives of Photogrammetry and Remote Sensing*, Amsterdam, Vol. XXXIII, Part B3, pp. 436-443.

Jahn, H., 2000b. Parallel Epipolar Stereo Matching. In: 15th Int. Conf. on Pattern Recognition, Barcelona, Vol. 1, pp. 402-405.

Perona, P., T. Shiota, T., Malik, J., 1994. Anisotropic diffusion, in: B. ter Haar Romeny (Ed.), Geometry-Driven Diffusion in Computer Vision, pp. 73 - 92, Kluwer Academic Publishers, Dordrecht