We are IntechOpen, the world's leading publisher of Open Access books Built by scientists, for scientists

4,800
Open access books available

122,000
International authors and editors

135M
Downloads

Our authors are among the

154
Countries delivered to

TOP 1%
most cited scientists

12.2%
Contributors from top 500 universities

CLARIVATE ANALYTICS BOOK CITATION INDEX INDEXED

WEB OF SCIENCE™

Selection of our books indexed in the Book Citation Index in Web of Science™ Core Collection (BKCI)

Interested in publishing with us?
Contact book.department@intechopen.com

Numbers displayed above are based on latest data collected.
For more information visit www.intechopen.com

21

Global Techniques for Edge based Stereo Matching

Yassine Ruichek, Mohamed Hariti, Hazem Issa
University of Technology of Belfort-Montbéliard
France

1. Introduction

Depth perception is one of the most active research areas in computer vision. Passive stereo vision is a well known technique for obtaining 3-D depth information of objects seen by two or more video cameras from different viewpoints (Hartley & Zisserman, 2000|Brown et al., 2003). The difference of the viewpoint positions causes a relative displacement of the corresponding features in the stereo images. Such relative displacement, called disparity, encodes the depth information, which is lost when the three dimensional scene is projected on an image. The key problem, which is difficult to solve and computationally expensive (Barnard & Fisher, 1982), is hence to compare each feature extracted from one image with a number, generally large, of features extracted from the other image in order to find the corresponding one, if any. Once the matching process is established and the stereo vision system parameters are known, the depth computation is reduced to a simple triangulation (Jane & Haubecker, 2000|Dooze, 2001).

This chapter presents some recent research works proposed to solve the stereo matching problem. The presented methods are based on a global approach, which can be viewed as a constraint satisfaction problem where the objective is to highlight a solution for which the matches are as compatible as possible with respect to specific constraints. These methods are tested and evaluated for real-time obstacle detection in front of a vehicle using linear stereo vision.

2. Related Works

Many approaches have been proposed to solve the stereo matching problem. According to the considered application, the existing techniques are roughly grouped into two categories: area-based and feature-based (Haralick & Shapiro, 1992). Area-based methods use correlation between brightness patterns in the local neighbourhood of a pixel in one image with brightness patterns in the local neighbourhood of the other image (Scharstein & Szeliski, 2002|Saito & Mori, 1995|Han et al., 2001|Tang et al., 2002). These methods, which lead to a dense depth map, are generally used for 3D scene reconstruction applications. Feature-based methods use zero-crossing points, edges, line segments, etc. and compare their attributes to find the corresponding features (Lee & Leou, 1994|Lee & Lee, 2004|Nasrabadi, 1992|Tien, 2004|Pajares & de la Cruz, 2004|Candocia & Adjouadi, 1997|Starink & Backer, 1995). These methods, which lead to a sparse depth map, are

Source: Scene Reconstruction, Pose Estimation and Tracking, Book edited by: Rustam Stolkin,
ISBN 978-3-902613-06-6, pp.530, I-Tech, Vienna, Austria, June 2007

generally used to ensure environment perception, as for obstacle detection. Feature-based methods can be used also for 3D scene reconstruction by interpolating the sparse depth map. To resolve matching ambiguities, feature-based and area-based methods use some constraints like epipolar, uniqueness, smoothness and ordering (Wang & Hsiao, 1999|Zhang et al., 2004).

In the robot vision domain, the stereo matching problem is generally simplified by making hypotheses about the type of objects being observed and their visual environment so that structural features, such as corners or vertical straight lines, can be more or less easily extracted (Kriegman et al., 1989). Indoor scenes, including a few rigid objects scattered without occlusions against a featureless background, are much easier to analyze than natural outdoor scenes of the real world (Nitzan, 1988). With such restrictive assumptions, the number of candidate features for matching is substantially reduced so that computing times become acceptable for real-time processing without an important loss of useful information. Unfortunately, none of these hypotheses can be used in outdoor scenes, such as road environments, for detecting and localizing obstacles in front of a moving vehicle, because the features are too numerous to allow a reliable matching within an acceptable computer time (Bruyelle & Postaire, 1993).

Considering these difficulties, some authors have proposed to use linear cameras instead of matrix ones (Bruyelle & Postaire, 1993|Inigo & Tkacik, 1987|Colle, 1990). With these cameras, the information to be processed is drastically reduced since their sensor contains only one video line, typically 2 500 pixels, instead of, at least, 250 000 pixels with standard raster-scan cameras. Furthermore, they have a better horizontal resolution than video cameras. This characteristic is very important for an accurate perception of the scene in front of a vehicle.

To solve the problem of matching edges extracted from stereo linear images, a classical approach is to use correlation techniques (Bruyelle & Postaire, 1993). In order to improve this basic approach, it has been proposed to explore the edges of the two linear images sequentially, from one end to the other. A majority of candidate edges can be matched without ambiguities by means of this scheme, performed forward and backward (Burie et al., 1995). However, this sequential procedure can leave some unmatched edges, and may lead to false matches, which are difficult to identify.

This chapter is concerned with the stereo matching problem for obstacle detection using linear cameras. The proposed approach is based on a global formulation of the stereo matching problem. Considering only possible matches that respect local constraints, the principle of this approach consists in searching a solution for which the matches are as compatible as possible with respect to global constraints. Thus, the stereo matching problem can be viewed as a constraint satisfaction problem. This approach is turned in different methods, which are evaluated and compared for obstacle detection using linear stereo vision.

3. Linear Stereo Vision

3.1 How to Build a Linear Stereo Vision Set-up

A linear stereo set-up is built with two line-scan cameras, so that their optical axes are parallel and separated by a distance E (see Figure 1). Their lenses have identical focal lengths f. The fields of view of the two cameras are merged in one single plane, called the optical plane, so that the cameras shoot the same line in the scene. A specific calibration

method has been developed to adjust the parallelism of the two optical axes in the common plane of view attached to the two cameras (Bruyelle, 1994). This calibration technique necessitates a specific planar calibration chart (see Figure 2), which bears two horizontal and two vertical calibration marks. The stereo set-up is calibrated when the vertical calibration marks are seen by the two cameras and when the horizontal calibration marks are at the centre of the two linear images. A set of oblique lines is provided so that the user knows if he is adjusting the positions of the cameras in the right direction.

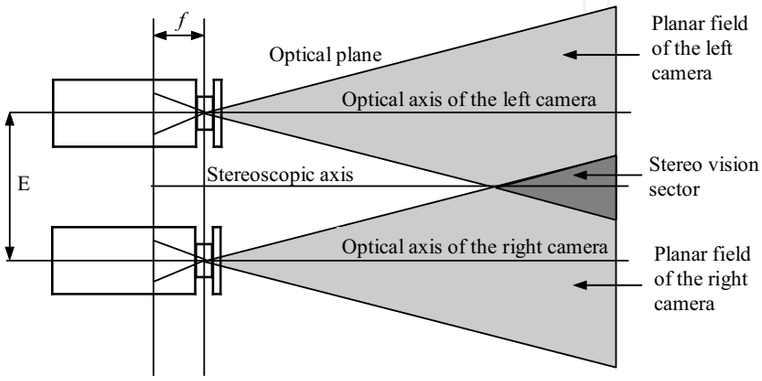


Figure 1. Geometry of the cameras

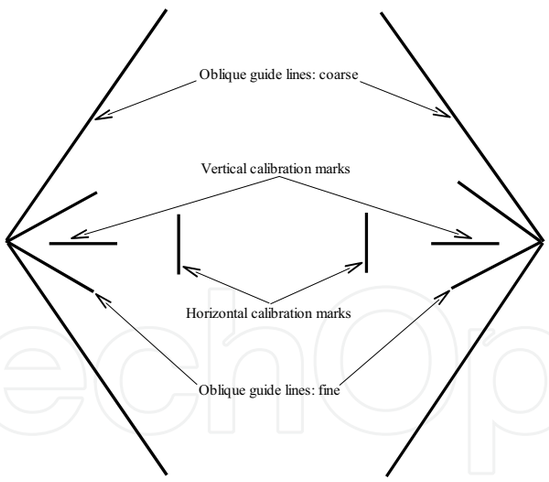


Figure 2. Calibration chart of the linear stereo set-up

If any object intersects the stereo vision sector, which is the common part of the two fields of view in the optical plane, it produces a disparity between the two linear images and, as a consequence, can be localized by means of triangulation.

Let the base-line joining the perspective centers O_l and O_r be the X-axis, and let the Z-axis lie in the optical plane, parallel to the optical axes of the cameras, so that the origin of the *{X,Z}* coordinate system stands midway between the lens centers (see Figure 3). Let us consider a point $P(X_p,Z_p)$ of coordinates X_p and Z_p in the optical plane. The image coordinates x_l and x_r represent the projections of the point P in the left and right imaging sensors, respectively. This pair of points is referred to as a corresponding pair. Using the pin-hole lens model, the coordinates of the point P in the optical plane can be found as follows:

$$z_P = \frac{E \cdot f}{d} \tag{1}$$

$$X_P = \frac{x_l \cdot z_P}{f} - \frac{E}{2} = \frac{x_r \cdot z_P}{f} + \frac{E}{2} \tag{2}$$

where f is the focal length of the lenses, E is the base-line width and $d = |x_l - x_r|$ is the disparity between the left and right projections of the point P on the two sensors.

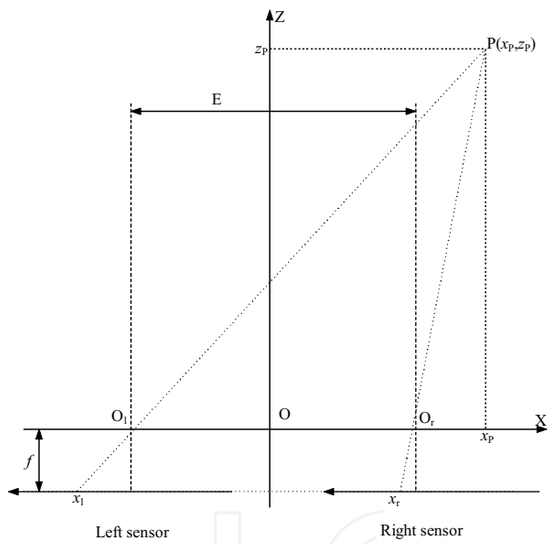


Figure 3. Pin-hole lens model

3.2 Feature Extraction

The low-level processing of a couple of two stereo linear images yields the features required in the correspondence phase. Edges appearing in these simple images, which are one-dimensional signals, are valuable candidates for matching because large local variations in the gray-level function correspond to the boundaries of objects being observed in a scene.

Edge detection is performed by means of the Deriche's operator (Deriche, 1990). After derivation, the pertinent local extrema are selected by splitting the gradient magnitude signal into adjacent intervals where the sign of the response remains constant (Burie et al., 1995) (see Figure 4). In each interval of constant sign, the maximum amplitude indicates the

position of a unique edge associated to this interval when, and only when, this amplitude is greater than a low threshold value *t*. The application of this thresholding procedure allows to remove non significant responses of the differential operator lying in the range *[-t,+t]*. The adjustment of *t* is not crucial. Good results have been obtained with *t* adjusted at *10%* of the greatest amplitude of the response of the differential operator.

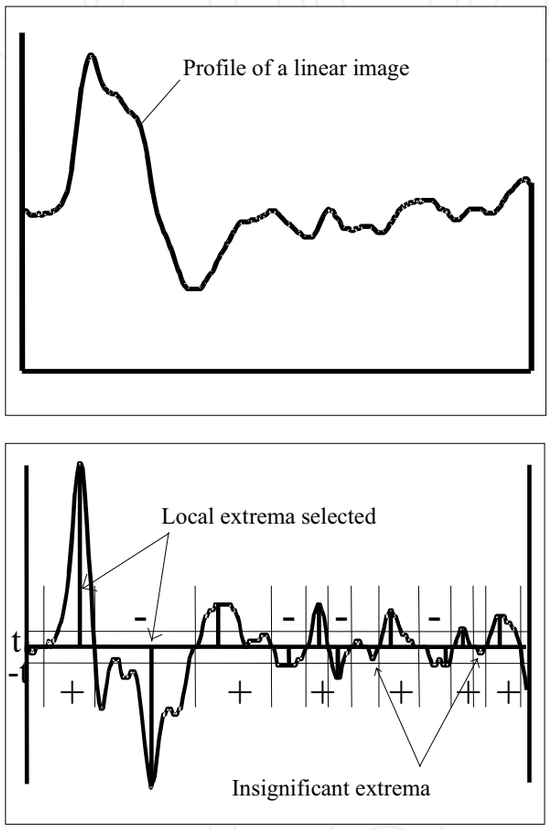


Figure 4. Edge extraction

Applied to the left and right linear images, this edge extraction procedure yields two lists of edges. Each edge is characterized by its position in the image, the amplitude and the sign of the response of the Deriche's operator.

4. Hopfield Neural Network Based Stereo Matching

4.1 Problem formulation

Let *L* and *R* be the left and right lists of the edges, respectively. The matching between *L* and *R* is first formulated as an optimization problem where an objective function, which represents the constraints on the solution, is to be minimized. The objective function,

defined such that the best matches correspond to its minimum value, is then mapped onto a Hopfield neural network for minimization.

The objective function is defined from three global constraints. The first one is the uniqueness constraint, which assumes that one edge in L matches only one edge in R (and vice-versa). The second global constraint is the ordering constraint, which is used to preserve the order, in the images, between the matched edges. This means that if an edge l in L is matched with an edge r in R, then it is impossible for an edge l' in L, such that $x_{l'} < x_l$, to be matched with and edge r' in R for which $x_{r'} > x_r$, where x denotes the position of the edge in the image. The third constraint is the smoothness constraint, which assumes that neighboring edges have similar disparities.

Combining the three global constraints, the objective function, representing the stereo correspondence problem, is defined so that its minimum value corresponds to the best solution. It can be expressed as:

$$\begin{aligned} H = & \frac{K_u}{2}\sum_{l\in L}\left(1-\sum_{r\in R/(l,r)\in\Omega}E_{lr}\right)^2+\frac{K_u}{2}\sum_{r\in R}\left(1-\sum_{l\in L/(l,r)\in\Omega}E_{lr}\right)^2 \\ & +\frac{K_o}{2}\sum_{(l,r)\in\Omega}\sum_{(l',r')\in\Omega}O_{lrl'r'}E_{lr}E_{l'r'}-\frac{K_s}{2}\sum_{(l,r)\in\Omega}\sum_{(l',r')\in\Omega}S_{lrl'r'}E_{lr}E_{l'r'} \end{aligned} \tag{3}$$

where K_u, K_o, and K_s are weighting positive constants, which are set experimentally to *5*, *1* and *1*, respectively. E_{lr} represents the matching state between the edge l in L and the edge r in R: if $E_{lr} = 1$, then the edges l and r are matched; otherwise they are not matched. Ω is the set of all possible matches between the edges in L and those in R, i.e. the set of all pairs of edges (l,r) that satisfy the two following local constraints. Resulting from the sensor geometry, the first one is the geometric constraint, which assumes that a couple of edges l and r appearing in L and R, respectively, represents a possible match only if the constraint $x_l > x_r$ is satisfied. The second local constraint is the slope constraint, which means that only edges with the same sign of the gradient are considered for a possible matching.

$$\Omega = \left\{(l,r)\in L\times R/(l,r)\ \text{satisfy the local constraints}\right\} \tag{4}$$

The two first terms of the objective function correspond to the uniqueness constraint. It can be seen that these terms tend to increase when multiple matches occur (i.e. when an edge in L (respectively R) has more than one corresponding edge in R (respectively L)). The first term (respectively second term) tends to a minimum value when the sum of the matching states of all possible matches of an edge in L (respectively R) is equal to *1*.

The third term allows the ordering constraint to be respected. The coefficient $O_{lrl'r'}$ indicates whether the order between the two pairs (l,r) and (l',r') is respected:

$$O_{lrl'r'} = \left|s(x_l - x_{l'}) - s(x_r - x_{r'})\right| \tag{5}$$

with:

$$s(a) = \begin{cases} 1 & if \quad a > 0 \\ 0 & otherwise \end{cases} \tag{6}$$

The fourth term of the objective function is used to enforce the smoothness constraint. The coefficient $S_{lrl'r'}$ indicates how compatible the two pairs *(l,r)* and *(l′,r′)* are:

$$S_{lrl'r'} = S(X_{lrl'r'}) = \frac{2}{1 + e^{\alpha \cdot (X_{lrl'r'} - \omega)}} - 1 \tag{7}$$

where $X_{lrl'r'}$ is the absolute value of the difference of disparities of the pairs *(l,r)* and *(l′,r′)*. The nonlinear function *S(X)* scales the compatibility measure smoothly between *-1* and *1*. The parameter ω is adjusted so as to allow some tolerance with respect to noise and distortion. It is chosen such that a high compatibility is reached for a good match when *X* is close to *0*, while a low compatibility corresponds to a bad match when *X* is very large. A satisfying value of this parameter is experimentally selected as $\omega = 20$. The parameter α controls the slope of the function *S(X)* when $X = \omega$. This parameter is set experimentally to *0.1*.

4.2 Objective Function Mapping onto a Hopfield Neural Network

After the mathematic formulation of the stereo correspondence problem as an optimization task, the next step is to map the objective function onto a Hopfield neural network for minimization. The neural network consists of a set of neurons mutually interconnected: each neuron is connected to all the other ones, except itself (Hopfield & Tank, 1985). A neuron n_{lr} of the network represents a possible match between the edges *l* and *r* appearing in *L* and *R*, respectively (see Figure 5). Note that only the pairs satisfying the local constraints are represented in the Hopfield neural network. The output of the neuron n_{lr} corresponds to the matching state E_{lr}.

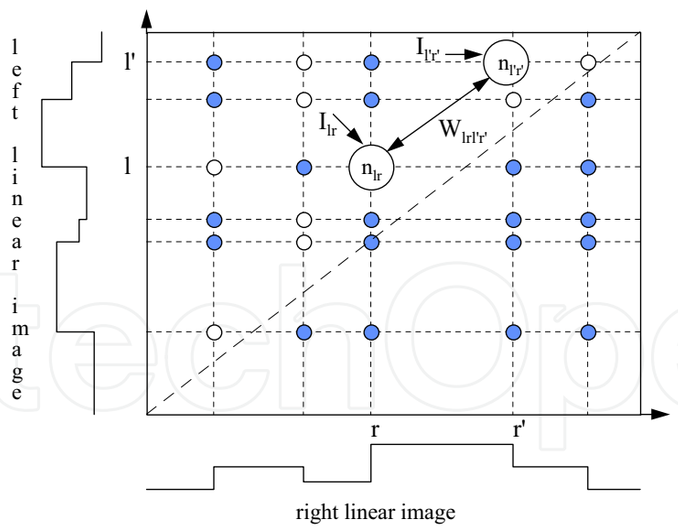


Figure 5. Hopfield neural network architecture. The white circles correspond to the neurons representing possible matches whereas the black ones correspond to the neurons representing impossible matches with respect to the local constraints

To determine the connection weights $\{W_{lrl'r'}\}$ between the neurons and the external inputs $\{I_{lr}\}$, the objective function is rearranged in the form of the energy function of the Hopfield neural network:

$$H = \sum_{(l,r)\in\Omega}\sum_{(l',r')\in\Omega} W_{lrl'r'}E_{lr}E_{l'r'} - \sum_{(l,r)\in\Omega} I_{lr}E_{lr} \tag{8}$$

with:

$$\begin{aligned} W_{lrl'r'} = &-K_u \cdot \left[\delta_{ll'}(1-\delta_{rr'}) - \delta_{rr'}(1-\delta_{ll'})\right] \\ &- K_o \cdot O_{lrl'r'}(1-\delta_{ll'})(1-\delta_{rr'}) \\ &+ K_s \cdot S_{lrl'r'}(1-\delta_{ll'})(1-\delta_{rr'}) \end{aligned} \tag{9}$$

$$\text{and} \quad I_{lr} = 2K_u$$

where:

$$\delta_{ij} = \begin{cases} 1 & if \quad i > j \\ 0 & otherwise \end{cases} \tag{10}$$

Once the objective function has been mapped onto the Hopfield neural network, the minimization process is achieved by letting the so-defined network evolve so that it reaches a stable state (i.e. when no change occurs in the state of its neurons during the updating procedure). For the neural network relaxation, we have chosen a continuous dynamic evolution in which the output of the neurons is allowed to vary continuously between *0* and *1*. It has been shown that continuous Hopfield neural networks perform better than discrete ones in which the neuron outputs are restricted to the binary values *0* and *1*. With a continuous Hopfield network, the output of a neuron can be interpreted as a matching probability or quality, which is quantified continuously from *1* for a correct match to *0* for a wrong match. To start the network evolution, the neural states are set to *0.5*, i.e. all the possible matches are considered with the same probability. During the network evolution, the state of neurons representing good matches converges toward *1* and the state of neurons representing bad matches converges toward *0*.

The equation describing the time evolution of a neuron n_{lr} in a continuous Hopfield neural network is:

$$\frac{du_{lr}}{dt} = -\frac{u_{lr}}{\tau} + \sum_{(l',r')\in\Omega} W_{lrl'r'}E_{l'r'} + I_{lr} \tag{11}$$

where u_{lr} is the internal input of the neuron n_{lr} and τ is a time constant, which is set to *10*. The internal input u_{lr} and the output E_{lr} of the neuron n_{lr} are coupled as:

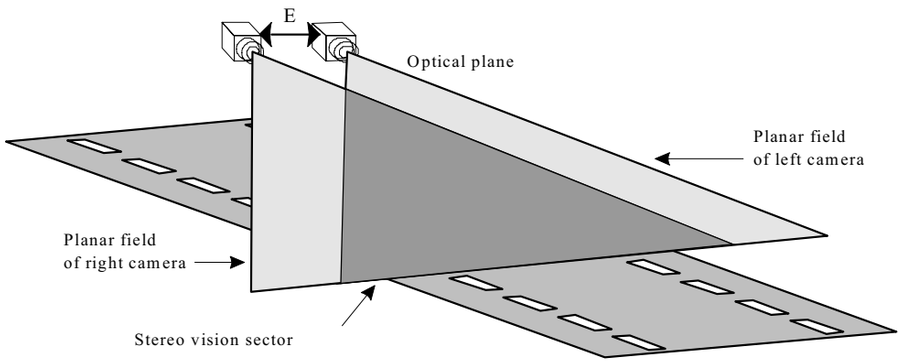
$$E_{lr} = \frac{1}{2}\left(1 + \tanh\left(\frac{u_{lr}}{\lambda}\right)\right) \tag{12}$$

where λ is a parameter, which determines how close the final state of the neurons is to the binary values *0* and *1*. This parameter is set experimentally to *0.01*.

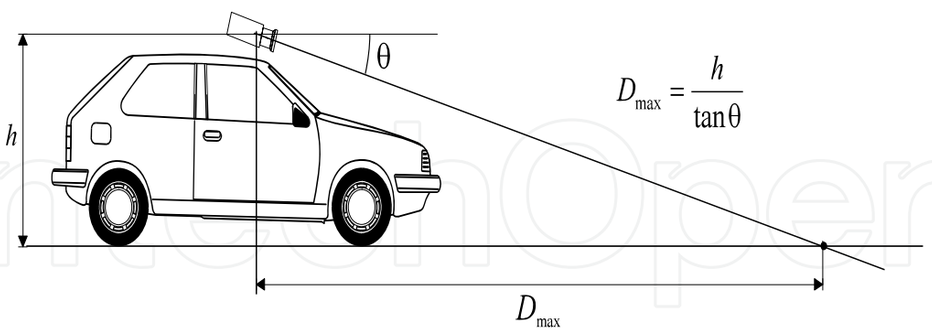
To extract the pairs of corresponding edges from the final state of the network, a procedure is designed to select the neurons for which the output is maximum in each row and each column of the network. This is achieved by selecting, in each row of the network, the neuron with the largest response. However, this procedure can select more than one neuron in a same column. To discard this configuration, which corresponds to multiple matches, the same procedure is applied to each column of the network. The neurons selected by this two-step procedure indicate the correct matches.

4.3 Application to Obstacle Detection

A linear stereo set-up is installed on top of a car, 1.5 m above the level of the road, for periodically acquiring stereo pairs of linear images as the car travels (see Figure 6). The tilt angle is adjusted so that the optical plane intersects the pavement at a distance D_{max} = 50 m in front of the car. This configuration ensures that every object that lies on the road in front of the vehicle is seen by the two cameras, even if its height is very small.



(a)



(b)

Figure 6. Stereo set-up configuration. (a) Top view. (b) Side view

One of the sequences shot in field conditions by this set-up is shown in Figure 7. In these pictures, the linear images are represented as horizontal lines, time running from top to bottom. In this example, the left and right sequences are composed by 200 linear images each. In this sequence, a pedestrian travels in front of the car according the trajectory shown in Figure 8. On the images of the sequence, we can clearly see the white lines of the pavement. The shadow of a car, located out of the vision plane of the stereoscope, is visible on the right of the images as a black area.

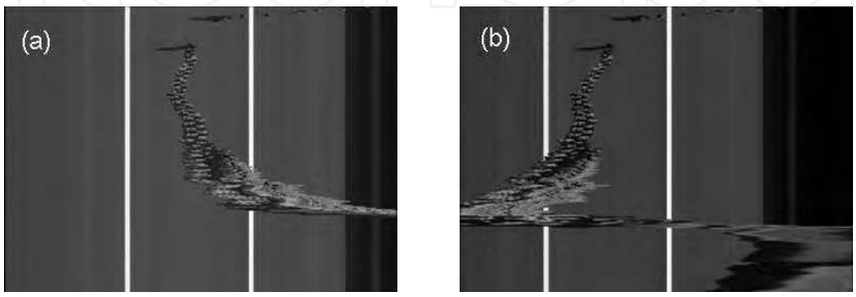


Figure 7. Stereo sequence. (a) Left sequence. (b) Right sequence

The neural processing of this stereo sequence allows determining the matched edge pairs. The disparities of all matched edges are used to compute the horizontal positions and distances of the object edges seen in the stereo vision sector. The results are shown in Figure 9 in which the horizontal positions are represented along the horizontal axis and the distances are represented by color levels, from the red, which corresponds to the farther distance, to the blue, which corresponds to the closer distance. As in Figure 7, time runs from top to bottom. The edges of the two white lines have been correctly matched and their detection is stable along the sequence. Indeed, the positions and distances remain constant from line to line. The pedestrian is well detected as he comes closer and closer to the car. The transition between the pavement and the area of shadow is also well detected. The presence of a few bad matches is noticed when occlusions occur (i.e. when the pedestrian hides one of the white lines to the left or right camera). These errors are caused by matching the edges of the white line, seen by one of the cameras, with those representing the pedestrian. Using an *AMD Athlon XP 2800+ PC* with *1.67 GHz* and *512 Mo RAM*, the processing rate is about *90* pairs of stereo linear images per second.

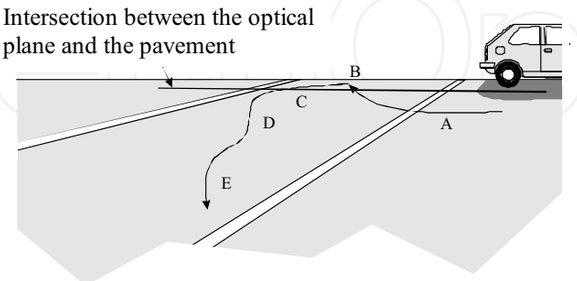


Figure 8. Trajectory of the pedestrian during the sequence

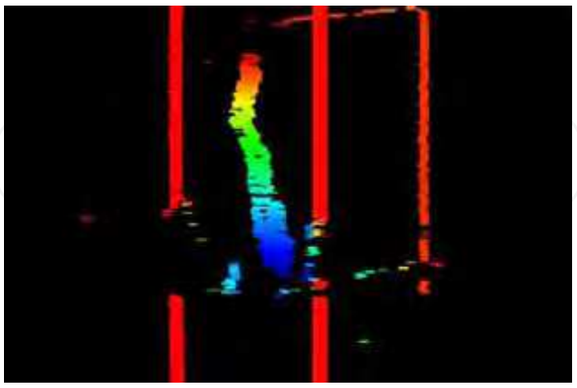


Figure 9. Neural stereo reconstruction

5. Genetic Algorithm Based Stereo Matching

It is known that Hopfield neural networks can perform only a local optimization process, and thus, they do not always guarantee to reach the global optimum, which corresponds to the best matching solution.

Genetic Algorithms (GAs) are randomized searching and optimization techniques guided by the principles of evolution and natural genetics (Goldberg, 1989). They are efficient, adaptive and robust search processes, and they are not affected by the presence of spurious local optimum in the solution space. Indeed, GAs span the solution space and can concentrate on a set of promising solutions that reach the global optimum or converge near the optimal solution. GAs have been applied successfully in many fields such as image processing, pattern recognition, machine learning, etc. (Goldberg, 1989).

5.1 Integer Encoding Scheme

To solve the stereo correspondence problem by means of a genetic algorithm, one must find a chromosome representation in order to code the solution of the problem. Let L and R be the lists of the edges extracted from the left and right linear images, respectively. Let N_L and N_R be the numbers of edges in L and R, respectively. A classical encoding scheme is to encode all the possible matches that meet the local constraints as a binary string B (see Figure 10). Each element B_k of this binary string contains two records. The first one, which is static, represents a possible match between an edge i in the left image and an edge j in the right one. The second record, which takes binary values, indicates if the hypothesis that the edge i is matched with the edge j is valid or not. If this record is set to *1*, then the edges i and j are matched; otherwise they are not matched. This encoding scheme is referred hereafter to as a binary encoding scheme since it allows manipulating binary chromosomes. Note that a binary chromosome can be represented as a N_LxN_R array M in which each element M_{ij} validates or not the hypothesis that the edge i in the left image matches the edge j in the right image (see Figure 11). If $M_{ij} = 1$, then the edges are matched; otherwise, they are not matched. Note that only the possible matches, which respect the local constraints, are represented in this array. Figure 11 shows an example of a binary chromosome represented by an array. *NbPix* is the number of pixels in the linear images.

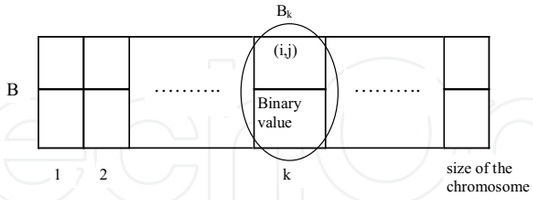


Figure 10. Binary chromosome

As we can see in Figure 11, the binary encoding scheme may produce chromosomes with many ambiguities when multiple possible matches appear simultaneously on the lines and columns of their corresponding arrays. Therefore, a genetic algorithm based on this encoding scheme will not explore efficiently the solution space and, as a consequence, it will be necessary to perform a great number of iterations to reach an acceptable solution. Furthermore, handling binary chromosomes, which are large-sized chromosomes, requires an important computing effort. To overcome the limitations that appear when handling classical binary chromosomes, we propose a new encoding scheme, which produces compact chromosomes with less matching ambiguities.

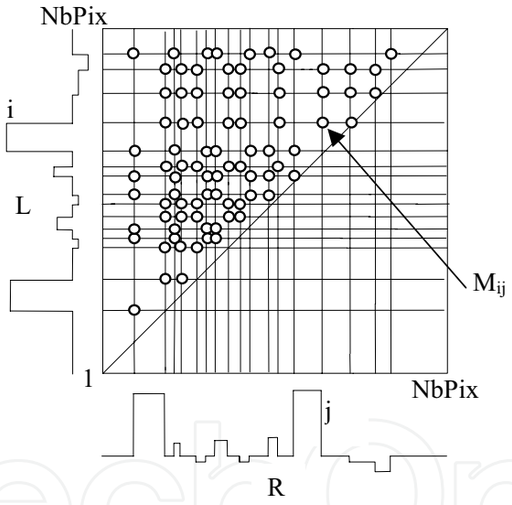


Figure 11. Array representation of a binary chromosome

Let $T_{max} = \{1,2,\ldots,N_{max}\}$ and $T_{min} = \{1,2,\ldots,N_{min}\}$ be the edge lists to be matched, where $N_{max} = max\ (N_L,N_R)$ and $N_{min} = min\ (N_L,N_R)$ are the sizes of T_{max} and T_{min}, respectively. This means that if $N_{max} = N_L$, then $T_{max} = L$ and $T_{min} = R$ (and vice-versa). The new encoding scheme, referred hereafter to as an integer encoding scheme, consists in representing a solution as a chain C indexed by the elements of the list T_{max} and which takes its values in the list $\{0\} \cup T_{min}$. The interpretation of the new encoding scheme is as follows. If $C_i = 0$, then the edge i in T_{max} has no corresponding edge; otherwise, the edges i in T_{max} and C_i in T_{min} are

matched. As for binary chromosomes, the integer ones encode only possible matches, which respect the local constraints. Figure 12 gives an example of an integer chromosome, which represents a matching possibility between the edge lists L and R of Figure 11. In this example, $N_{max} = N_R = 17$ and $N_{min} = N_L = 15$, thus $T_{max} = R$ and $T_{min} = L$.

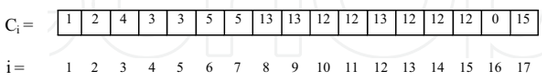


Figure 12. Chromosome based on the integer encoding scheme

Note that it is easy to represent an integer chromosome C as a $N_L x N_R$ array M'. For $i \in T_{max}$ and $j \in T_{min}$, $M'_{ij} = 1$ if $C_i = j$; otherwise $M'_{ij} = 0$. Figure 13 illustrates the array representation of the integer chromosome of Figure 12. We can see in this figure that there are no ambiguities in the columns of this array. In general, if $T_{max} = L$, then the integer encoding scheme produces chromosomes with no ambiguities in the lines of their corresponding arrays. In the opposite case, i.e., if $T_{max} = R$, as in Figure 13, the integer encoding scheme produces chromosomes with no ambiguities in the columns of their corresponding arrays.

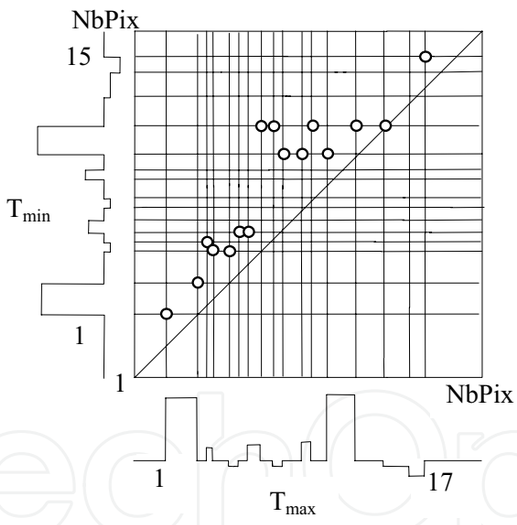


Figure 13. Array representation of the integer chromosome of Figure 12

The integer encoding scheme will therefore allow a genetic algorithm to explore more efficiently the solution space and, thus, to converge toward a better solution within a lower number of generations than when using the binary encoding scheme. Furthermore, the integer chromosomes, which are smaller than the binary ones, will require less computing time for their mutation and evaluation. In the examples given above, the integer chromosomes are 17-sized whereas the binary ones are 84-sized.

5.2 Integer Chromosome Evaluation

A genetic algorithm needs a fitness function for evaluating the chromosomes. The fitness function is defined from the global constraints so that the best matches correspond to its minimum:

$$\begin{aligned} F_{\text{int}eger}(C) = K_u & \sum_{i=1}^{N_{\max}-1} \sum_{k=i+1}^{N_{\max}} U(C_i, C_k) \\ & + K_m \left(N_{\max} - N_{\min} - \sum_{i=1}^{N_{\max}} Z(C_i) \right)^2 \\ & + K_o \sum_{i=1}^{N_{\max}-1} \sum_{k=i+1}^{N_{\max}} O(C_i, C_k) \\ & - K_s \sum_{i=1}^{N_{\max}} \sum_{k=1/k\neq i \text{ and } C_i \neq C_k}^{N_{\max}} S(C_i, C_k) \end{aligned} \tag{13}$$

where C is the integer chromosome to be evaluated. K_u, K_m, K_o and K_s are weighting positive constants, which are experimentally set to *5*, *5*, *5* and *1*, respectively.

The first term of the fitness function corresponds to the uniqueness constraint, where the quantity $U(C_i,C_k)$ represents a penalty when the constraint is not respected, i.e., when the two edges i and k have a same corresponding one. This penalty is computed as follows:

$$U(C_i, C_k) = \begin{cases} 1 \text{ if } C_i = C_k \text{ and } (C_i \neq 0, C_k \neq 0) \\ 0 \text{ otherwise} \end{cases} \tag{14}$$

The second term allows promoting chromosomes with an important number of matches. This term tends to a minimum when the number of matches is equal to N_{min}. The quantity $Z(C_i)$ is computed as follows:

$$Z(C_i) = \begin{cases} 1 \text{ if } C_i = 0 \\ 0 \text{ otherwise} \end{cases} \tag{15}$$

The third term is used to respect the ordering constraint. The quantity $O(C_i,C_k)$ represents a penalty when the order between the two pairs of edges (i,C_i) and (k,C_k) is not respected:

$$O(C_i, C_k) = \begin{cases} 1 \text{ if } C_k < C_i \text{ and } (C_i \neq 0, C_k \neq 0) \\ 0 \text{ otherwise} \end{cases} \tag{16}$$

The fourth term supports the smoothness constraint. The quantity $S(C_i,C_k)$ indicates how compatible are the two pairs of edges (i,C_i) and (k,C_k) with respect to the smoothness constraint. This compatibility measure is computed as follows:

$$S(C_i, C_k) = Q(X_{C_iC_k}) = \frac{2}{1 + e^{\alpha \cdot (X_{C_iC_k} - \omega)}} - 1 \tag{17}$$

where $X_{C_iC_k}$ is the absolute value of the difference between the disparities of the pairs of edges (i,C_i) and (k,C_k), expressed in pixels. The non-linear function Q is identical to the one

described in section 4.1 (see Equation 7). The parameters α and ω are experimentally set to *1* and *20*.

5.3 Genetic Stereo Matching Algorithm

The genetic algorithm for the edge stereo correspondence problem consists first in generating randomly an initial population of chromosomes representing possible matches that satisfy the local stereo constraints. The evolution process is then performed during some generations thanks to reproduction and selection operations in order to highlight the best chromosome, which minimizes the fitness function.

Starting from a current population in which each chromosome is evaluated, particular chromosomes are chosen with a selection probability proportional to the fitness value. These selected chromosomes are first reproduced using a single point crossover operation, i.e., two chromosomes are divided at a random position, and a portion of each chromosome is swapped with each other. The offspring chromosomes that are the result of the crossover operation are then submitted to a mutation procedure, which is randomly performed for each gene. The mutation of a gene number i of an integer chromosome C is performed by replacing its value by a new one chosen randomly in $\{0\} \cup T_{min} - \{C_i\}$ (see section 5.1).

After the crossover and mutation phases, a new population is obtained by means of two selection procedures: a deterministic selection and a stochastic one. These two selection procedures are applied to the set containing the chromosomes of the current population and those produced by the crossover and mutation operations. Based on an elitist strategy, the deterministic procedure is applied to select the best chromosomes, which represent 10% of the population. The remainder of the new population is obtained thanks to the stochastic selection, which is based on the same principle used to select chromosomes to be reproduced, i.e., with a selection probability proportional to the fitness value.

The algorithm is iterated until a pre-specified number of generations is reached. Once the evolution process is completed, the optimal chromosome, which corresponds to the minimum value of the fitness function, indicates the pairs of matched edges.

5.4 Genetic Parameter Setting

It is known that the convergence time of a genetic algorithm depends generally on the size of the population and the number of generations. To obtain good matching results, the values of these two parameters are chosen by taking into account the complexity of the problem, i.e., the sizes of the stereo edge lists to be matched. Thus, if the complexity of the problem is important, it is necessary to set these parameters to large values in order to reach a good solution. Furthermore, our experiment tests show that when there is a large difference between the sizes of the stereo edge lists, large values are required for these two parameters to converge toward a good solution. Considering these two observations, we propose the following empirical expressions for setting the size of the population *SizePop* and the number of generations N_{gen}:

$$SizePop = W_1 \cdot (N_L + N_R) + W_2 \cdot |N_L - N_R| \tag{18}$$

$$N_{gen} = W_3 \cdot SizePop \tag{19}$$

where N_L and N_R are the total numbers of edges in the left and right images, respectively. W_1, W_2 and W_3 are weighting positive constants, which are experimentally set to *5*, *2* and *1*, respectively. Concerning the other genetic parameters, the crossover probability is set to *0.6* and the mutation probability is equal to the inverse of the number of genes in a chromosome.

5.5 Performance Analysis of the Integer Encoding Scheme

To compare the performances of the integer and binary encoding schemes, two genetic algorithms are run separately. The first one, named integer genetic algorithm (IGA), uses integer chromosomes. The second one, named binary genetic algorithm (BGA), manipulates binary chromosomes. The chromosome evaluation is performed using a common fitness function, which is adapted to the array representation (see section 5.1). This fitness function allows evaluating both integer and binary chromosomes. It is constructed from the global constraints so that the best matches correspond to its minimum:

$$\begin{aligned} F_{array}(M) = {} & K_u \sum_{i \in L} \left(1 - \sum_{j \in R / (i,j) \in \Omega} M_{ij} \right)^2 \\ & + K_u \sum_{j \in R} \left(1 - \sum_{i \in L / (i,j) \in \Omega} M_{ij} \right)^2 \\ & + K_m \left(N_{\min} - \sum_{(i,j) \in \Omega} M_{ij} \right)^2 \\ & + K_o \sum_{(i,j) \in \Omega} \sum_{(k,l) \in \Omega} O_{ijkl} M_{ij} M_{kl} \\ & - K_s \sum_{(i,j) \in \Omega} \sum_{(k,l) \in \Omega} S_{ijkl} M_{ij} M_{kl} \end{aligned} \tag{20}$$

where M is the array representation of the integer or binary chromosome to be evaluated. K_u, K_m, K_o and K_s are weighting positive constants. Ω is the set of all possible matches between the edges in the lists L and R, i.e., the set of all pairs of edges *(i,j)* that satisfy the local constraints (see Equation 4).

The two first terms of the fitness function correspond to the uniqueness constraint. These terms tend to a minimum when the sum of the elements lying in each line and each column of the array is equal to *1*. The third term is used to enforce an important number of matches in the array. This term tends to a minimum when the number of matches is equal to $N_{min} = min(N_L,N_R)$. The fourth term is introduced to respect the ordering constraint. The coefficient O_{ijkl} indicates if the order between the pairs of edges *(i,j)* and *(k,l)* is respected (see Equation 5). The last term is used to support the smoothness constraint. The quantity S_{ijkl} indicates how compatible are the two pairs of edges *(i,j)* and *(k,l)* with respect to the smoothness constraint (see Equation 7).

The two genetic algorithms, i.e., the binary chromosome-based algorithm (BGA) and the integer chromosome-based algorithm (IGA), are applied to a couple of stereo linear images. There are 27 and 21 edges in the left and right linear images, respectively. Let us recall that

during their evolution, the two algorithms evaluate the chromosomes by using the fitness function F_{array}, associated with the array representation of the chromosomes (see Equation 20).

Figure 14 illustrates the evolution of the fitness function F_{array} using the integer genetic algorithm. Figure 15 and 16 show the evolution of the fitness function F_{array} using the binary genetic algorithm, with different values for the population size and the number of generations. Table 1 gives the fitness function values corresponding to the best chromosome obtained by the integer (IGA) and binary (BGA) genetic algorithms. With a population of *100* chromosomes, the fitness function reaches a minimum of *-128* after 300 generations using the integer genetic algorithm. Using the same values for the population size and the number of generations, the fitness function reaches a minimum of *112* when applying the binary genetic algorithm. By increasing the population size and the number of generations to *300* and *600*, respectively, the binary genetic algorithm leads the fitness function to a minimum value of *-99* .

As a conclusion to this discussion, the integer encoding scheme allows the genetic algorithm to explore more efficiently the solution space and, thus, to converge toward a better solution within a lower population size and a lower number of generations than when using the binary encoding scheme.

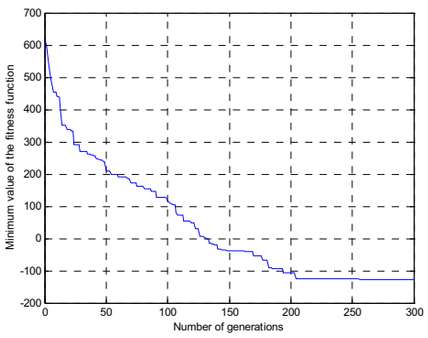


Figure 14. Evolution of the fitness function using the integer genetic algorithm

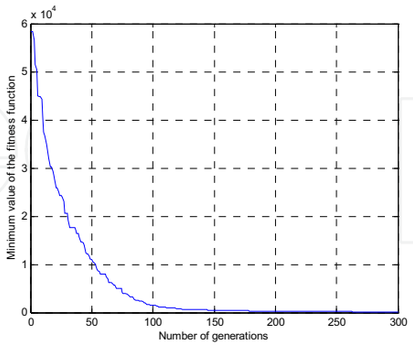


Figure 15. Evolution of the fitness function using the binary genetic algorithm, with a population of 100 chromosomes and 300 generations

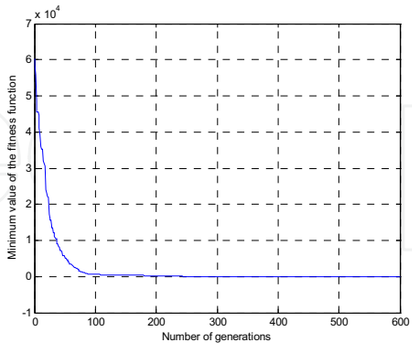


Figure 16. Evolution of the fitness function using the binary genetic algorithm, with a population of 300 chromosomes and 600 generations

Algorithm	Population size and number of generations	Fitness value
BGA	100 chromosomes and 300 generations	112
	300 chromosomes and 600 generations	-99
IGA	100 chromosomes and 300 generations	-128

Table 1. Performance comparison between the integer and binary encoding schemes

5.6 Genetic Stereo Matching Result

The integer genetic processing of the stereo sequence of Figure 7 provides the reconstructed scene represented in Figure 17. When compared to the binary genetic algorithm, the integer genetic algorithm allows reducing significantly the computing time. Indeed, the processing rate is about 2.7 stereo linear images per second instead of 0.1 stereo linear images per second (see Table 2).

The stereo matching results are comparable with those obtained by the neural stereo matching procedure (see Figures 9 and 17). However, the processing rate of the genetic stereo matching procedure is much lower when compared to the processing rate of the neural stereo matching algorithm (see Table 2).

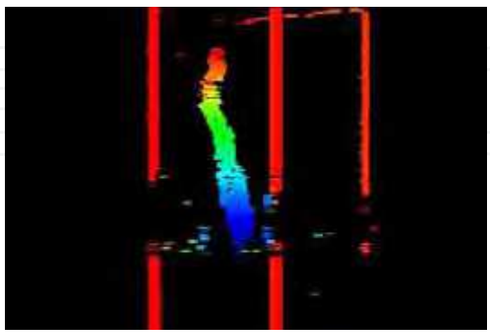


Figure 17. Integer genetic stereo reconstruction

Algorithm	Processing rate
BGA	0.1 pairs per second
IGA	2.7 pairs per second
Neural algorithm	90 pairs per second

Table 2. Processing rate comparison between the neural algorithm, integer genetic algorithm and binary genetic algorithm

6. Stereo Matching using a Multilevel Searching Strategy

Stereo matching is a combinatorial problem. To reduce the combinatory (resulting from the number of the edges considered in the stereo images) we propose a multilevel searching strategy, which decomposes hierarchically the problem into sub-problems with reduced complexities. The hierarchical decomposition performs edge stereo matching at different levels, from the most significant edges to the less significant ones. At each level, the process starts by selecting the edges with the larger gradient magnitudes. These edges are then matched and the obtained pairs are used as reference pairs for matching the most significant edges in the next level.

The multilevel searching strategy starts from level *1* in which all the left and right edges are considered. Let $L_1^0 = L$ and $R_1^0 = R$ be the lists of the edges extracted from the left and right images, respectively. We define from L_1^0 and R_1^0 a $NL_1^0 \times NR_1^0$ array MA_1^0 in which are represented all the possible matches that satisfy the local constraints (see Figure 18). NL_1^0 and NR_1^0 are the total numbers of edges in L_1^0 and R_1^0, respectively.

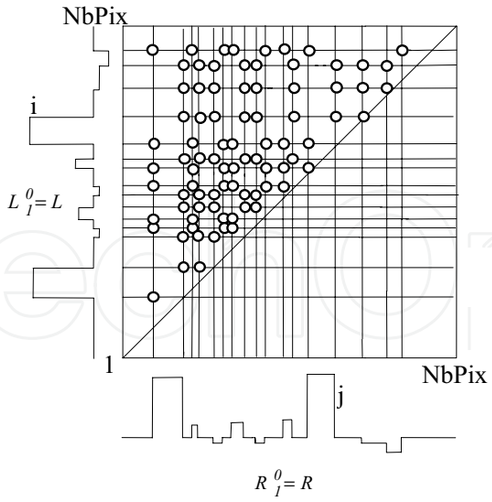


Figure 18. Array representation taking into account all the edges in the left and right images

The first step of the multilevel searching strategy consists of selecting, from L_1^0 and R_1^0, the edges with significant gradient magnitudes (see Figure 19). These selected edges are then matched and the obtained pairs, called reference pairs, define new sub-arrays, which are processed with the same searching strategy to match the most significant edges in level 2 (see Figure 20). In the example of Figure 20, four reference pairs are obtained from the first level. Thus, five sub-arrays, namely MA_2^0, MA_2^1, MA_2^2, MA_2^3 and MA_2^4, are considered in the second level.

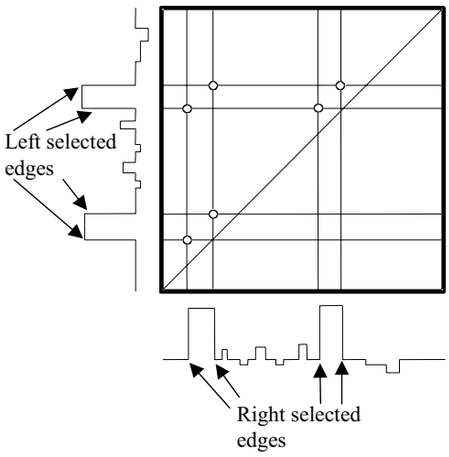


Figure 19. Level 1 of the multilevel searching strategy

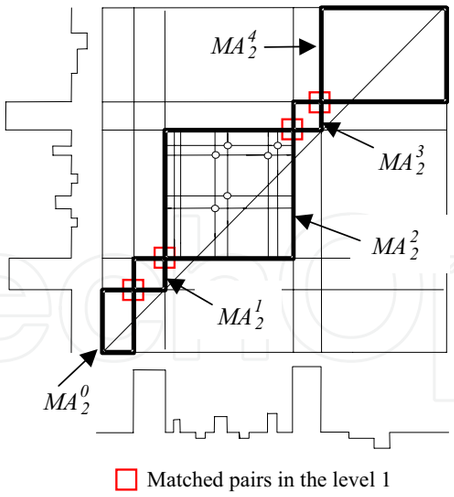


Figure 20. Level 2 of the multilevel searching strategy

Let MA_n^q be the sub-array number q in the level n. Let L_n^q and R_n^q be the left and right edge lists from which the sub-array MA_n^q is defined. The significant edges considered for the matching in the level n are selected in L_n^q and R_n^q such that their gradient magnitudes satisfy the following conditions:

$$\frac{\min_l}{2^{n-1}} \le mg_i \le \frac{\min_l}{2^n}$$
$$\text{or} \qquad i \in L_n^q \tag{21}$$
$$\frac{\max_l}{2^n} \le mg_i \le \frac{\max_l}{2^{n-1}}$$

$$\frac{\min_r}{2^{n-1}} \le mg_j \le \frac{\min_r}{2^n}$$
$$\text{or} \qquad j \in R_n^q \tag{22}$$
$$\frac{\max_r}{2^n} \le mg_j \le \frac{\max_r}{2^{n-1}}$$

where min_l and max_l (respectively min_r and max_r) are the smallest and largest gradient magnitudes of the edges extracted from the left (respectively right) image. mg_i and mg_j are the gradient magnitudes of the left edge i and right edge j, respectively.

Let K_n^q be the number of the matched pairs obtained from the matching of the selected edges in L_n^q and R_n^q. These pairs, called reference pairs, define new sub-arrays MA_{n+1}^0, $MA_{n+1}^1,\ldots, MA_{n+1}^{K_n^q}$, which are processed in the level $n+1$ using the same principle for matching the most significant edges in this level. In order to optimize its running, the multilevel searching strategy is implemented recursively.

The performance of the multilevel searching strategy is analyzed using the integer genetic algorithm, described in section 5. The using of the integer genetic algorithm with the multilevel searching strategy is referred hereafter to as a multilevel genetic algorithm (MiGA). On the other hand, the integer genetic algorithm performing stereo matching without the multilevel searching strategy, i.e., applied to all the edges extracted from the left and right linear images, is referred hereafter to as a basic genetic algorithm (BiGA).

Applied to the stereo sequence of Figure 7 (see section 4.3), the multilevel genetic algorithm provides the reconstructed scene shown in Figure 21. When we compare the reconstructed scenes obtained from BiGA (see Figure 17) and MiGA (see Figure 21), we can see that the matching results are globally similar. To evaluate quantitatively the performances of these two algorithms, we compare the matching solutions by means of an objective function constructed from the three terms corresponding to the uniqueness, ordering and smoothness constraints in the fitness function, which is defined by Equation 13 (see section 5.2). Figure 22 shows, for each genetic matching algorithm (BiGA and MiGA), the objective function values corresponding to the solutions obtained for each stereo pair of linear images of the sequence of Figure 7 (see section 4.3). We can see that the two algorithms behave almost identically except for the stereo pairs in the range [105,145] for which the multilevel scheme is less robust than the basic one. As we can see in Figure 7, some occlusions appear

in these stereo pairs (i.e. when the pedestrian hides one of the white lines to the left or right camera). The partial fail of the multilevel scheme is due probably to the edge selection procedure, which is performed before stereo matching at each level. Indeed, the selection procedure can select an edge from the left image while the corresponding one is not selected from the right image (and vice-versa). This situation occurs frequently in presence of occlusions.

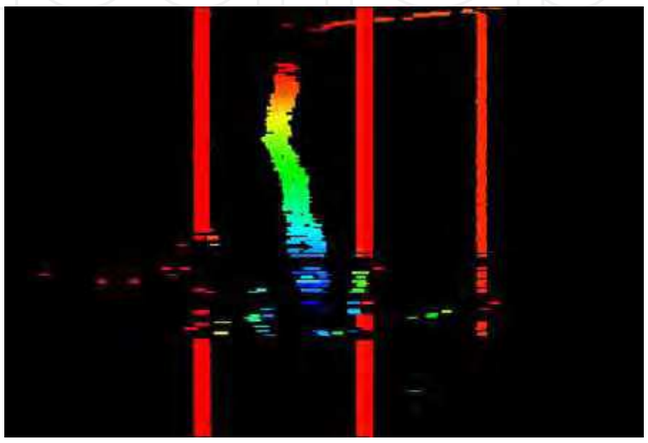


Figure 21. The reconstructed scene using the multilevel genetic algorithm

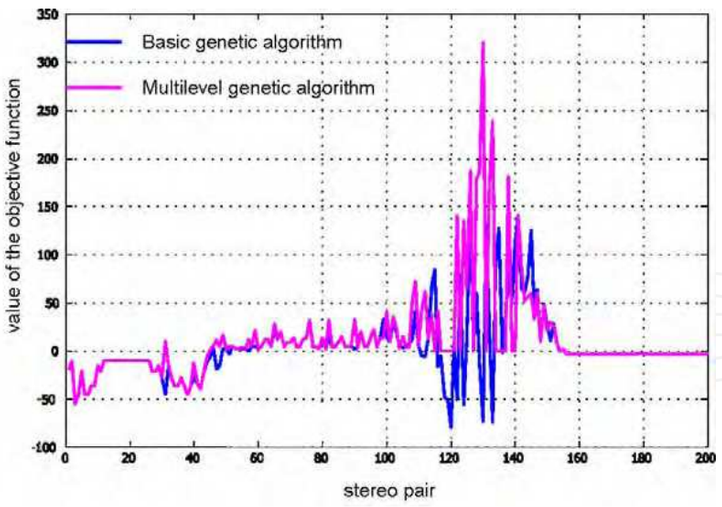


Figure 22. Quantitative analysis between the basic and multilevel genetic algorithms

This minor loss of robustness is the cost of the improvement in terms of processing time (see Table 3). With the multilevel searching strategy, the processing rate becomes 83 stereo pairs

per second, while it was only equal to 2.7 stereo pairs per second with the basic scheme. The same observations are noted when the multilevel searching strategy is associated with the neural stereo processing (see Figure 23 and Table 3).

Method	Processing rate	
	Without the multilevel searching strategy	With the multilevel searching strategy
Genetic algorithm	2.7 stereo pairs per second	83 stereo pairs per second
Neural algorithm	90 stereo pairs per second	260 stereo pairs per second

Table 3. Processing rate comparison between the basic and multilevel schemes

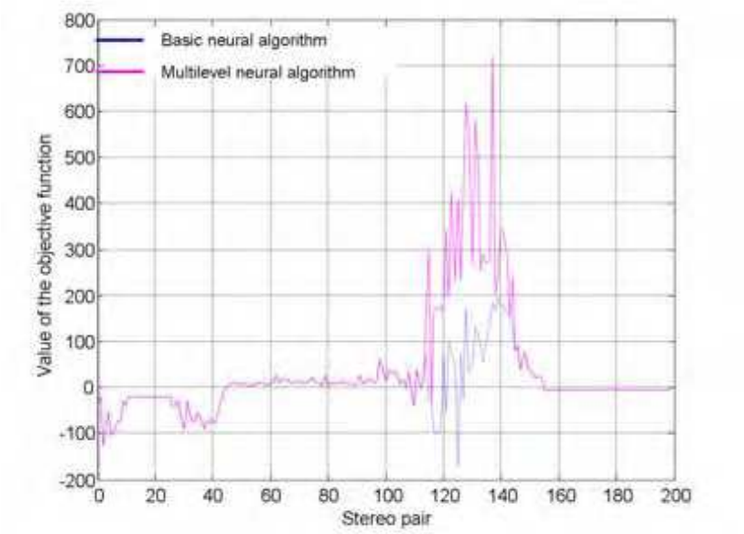


Figure 23. Quantitative analysis between the basic and multilevel neural algorithms

7. Voting Method Based Stereo Matching

The multilevel searching strategy allows improving significantly the stereo processing time. However, it may cause some difficulties because of the selection procedure applied at each level. Indeed, this procedure may select an edge from an image while the true corresponding one is not selected from the other image. Considering these difficulties, we propose an alternative stereo matching approach, which is based on a voting strategy.

7.1 Problem Mapping

Let L and R be the lists of the edges extracted from the left and right linear images, respectively. Let N_L and N_R be the numbers of edges in L and R, respectively. The edge stereo matching problem is mapped onto a N_LxN_R array M, called matching array, in which an element M_{lr} explores the hypothesis that the edge l in the left image matches or not the

edge *r* in the right image (see Figure 24). We consider only the elements representing the possible matches that met the position and slope constraints. For each element M_{lr} representing a possible match *(l,r)*, we associate a score SM_{lr}, which is set initially to zero.

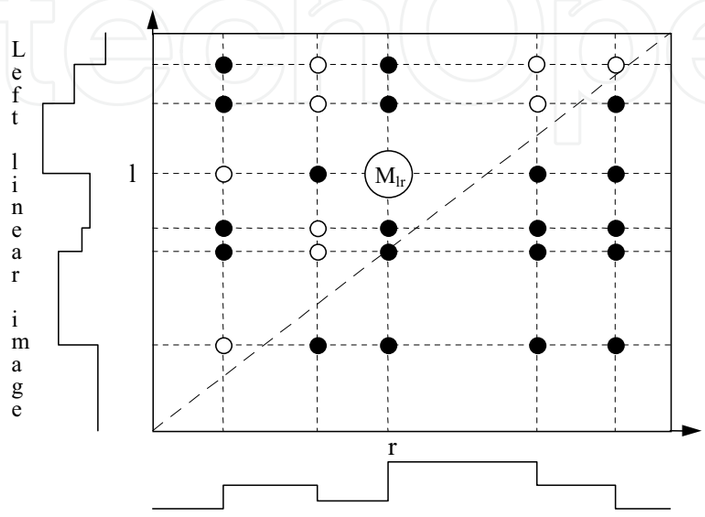


Figure 24. Matching array. The white circles represent the possible matches that met the position and slope constraints. The black circles represent the impossible matches that do not respect the position and slope constraints

7.2 Score Based Stereo Matching

The stereo matching process is performed thanks to a score-based procedure, which is based on the global constraints. The procedure is applied to all the possible matches that meet the local constraints, i.e., the position and slope constraints. This procedure consists in assigning for each possible match a score, which represents a quality measure of the matching regarding the global constraints. Let M_{lr} be an element of the matching array, representing a possible match between the edges *l* and *r* in the left and right images, respectively. The stereo matching procedure starts by determining among the other possible matches those that are authorized to contribute to the score SM_{lr} of the possible match M_{lr}. The contributor elements are obtained by using the uniqueness and ordering constraints: an element $M_{l'r'}$ is considered as a contributor to the score of the element M_{lr} if the possible matches *(l,r)* and *(l',r')* verify the uniqueness and ordering constraints (see Figure 25). The contribution of the contributors to the score of the element M_{lr} is then performed by means of the smoothness constraint. For each contributor $M_{l'r'}$, the score updating rule is defined as follows:

$$SM_{lr}(new) = SM_{lr}(previous) + G(X_{lrl'r'}) \tag{23}$$

where $X_{lrl'r'}$ is the absolute value of the difference between the disparities of the pairs *(l,r)* and *(l',r')*, expressed in pixels. *G* is a non linear function, which calculates the contribution of

the contributors. This function is chosen such that a high contribution corresponds to a high compatibility between the pairs *(l,r)* and *(l′,r′)* with respect to the smoothness constraint, i.e. when $X_{lrl'r'}$ is close to 0, and a low contribution corresponds to a low smoothness compatibility, i.e. when $X_{lrl'r'}$ is very large. This function is chosen as:

$$G(X) = \frac{1}{1+X} \tag{24}$$

By considering the different steps of the score-based procedure, the final score FSM_{lr} of the possible match M_{lr} can be computed as follows:

$$FSM_{lr} = SM_{lr}(final) = \sum_{(l',r') \in \Omega_{lr}} G(X_{lrl'r'}) \tag{25}$$

where Ω_{lr} is the set of all the possible matches *(l′,r′)* such that the pairs *(l,r)* and *(l′,r′)* satisfy the uniqueness and ordering constraints.

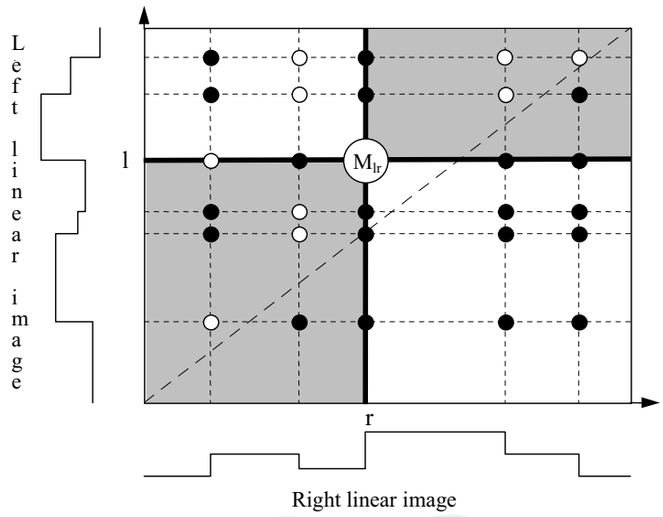


Figure 25. Determination of the contributors for calculating the score of a possible match: The contributors to the score of the possible match M_{lr} are the elements (white circles) situated in the gray area of the matching array

7.3 Extraction of the Correct Matches

To determine the correct matches, a procedure is designed to select the possible matches for which the final score is maximum. This is achieved by selecting in each row of the matching array the element with the largest score. However, this procedure can select more than one element in a same column of the matching array. To discard this configuration, which corresponds to multiple matches, the same procedure is applied to each column of the matching array. The elements selected by this two-steps procedure indicate the correct matches.

7.4 Voting Stereo Matching Result

The processing of the stereo sequence of Figure 7 (see section 4.3) using the voting procedure provides the reconstructed scene shown in Figure 26. We can see that the stereo matching results are very similar with those obtained by the neural and genetic algorithms (see Figures 9 and 17). The advantage of the voting technique is that it is fast, with a very high speed processing (see Table 4). Furthermore, the neural and genetic stereo techniques use many coefficients, which are usually difficult to adjust. The voting method is not confronted to this problem since it does not depend on any parameter.

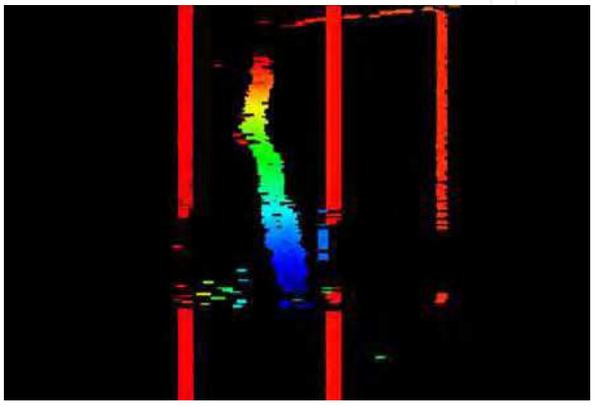


Figure 26. Reconstructed scene using the voting stereo technique

Method	Processing rate
Neural network based method	90 stereo pairs per second
Genetic algorithm based method	2.7 stereo pairs per second
Voting method	220 stereo pairs per second

Table 4. Processing rate comparison between the neural, genetic and voting techniques

8. Conclusion

We proposed global techniques for edge stereo matching. The problem is formulated as a constraint satisfaction problem where the objective is to highlight a solution for which the matches are as compatible as possible with respect to specific constraints: local constraints and global ones. The local constraints are used to discard impossible matches so as to consider only potentially acceptable pairs of edges as candidates. The global constraints are used to evaluate the compatibility between the possible matches in order to determine the best ones.

In the first technique, the problem is turned into an optimization task where an objective function, representing the global constraints, is mapped and minimized thanks to a Hopfield neural network. This neural optimization method constitutes a local searching process, and thus, it does not always guarantee to reach a global minimum. As an alternative, we suggested to use genetic algorithms, which span the solution space and can

concentrate on a set of promising solutions that reach the global optimum or converge near the optimal solution. A specific encoding scheme is presented and the analysis shows its efficiency to explore the solution space. However, the genetic technique necessitates a lot of time computation, and thus, it cannot be exploited for real-time applications such as obstacle detection in front of a moving vehicle. In order to improve the time computation, we proposed a multilevel searching strategy, which decomposes the stereo matching problem into sub-problems with reduced complexities. This searching strategy performs stereo matching from the most significant edges to the less significant ones. In each level, the procedure starts by selecting significant edges, using their gradient magnitude. The selected edges are then matched and the obtained pairs are used as reference pairs for matching the remaining edges according the same principle. The multilevel searching strategy allows improving significantly the stereo processing time. However, the limitation is that, in each level, the selection procedure may select an edge from an image while the true corresponding one is not selected from the other image. Considering this difficulty, we proposed a voting stereo matching technique, which consists to determine for each possible match a score based on the combination of the global constraints. This technique provides very similar stereo matching results with a high speed processing, compatible with real-time obstacle detection. Furthermore, unlike the neural and genetic stereo methods, the voting technique does not use any parameter, and hence, does not need any adjustment.

9. References

Hartley, R. & Zisserman, A. (2000). *Multiple View Geometry in Computer Vision,* Cambridge Univ. Press, Cambridge, U.K.

Brown, M.Z.; Burschka, D. & Hager, G.D. (2003). Advances in computational stereo. *IEEE Trans Pattern Anal Machine Intel.,* Vol. 25, (Aug. 2003) 993-1008)

Jane, B. & Haubecker, H. (2000). *Computer vision and applications,* Academic, New York

Dooze, D. (2001). *Conception et réalisation d'un stéréoscope bi-modal à portée variable: application à la détection d'obstacles à l'avant de véhicules guidés automatisés,* PhD Thesis, Université des Sciences et Technologies de Lille, France

Haralick, R.M. & Shapiro, L.G. (1992). *Image matching, computer and robot vision, Part 2,* Addison-Wesley, New York

Scharstein, D. & Szeliski, R. (2002). A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. *Int J Comput Vis,* Vol. 47, No. 1-3, (2002) 7-42

Saito, H. & Mori M. (1995). Application og genetic algorithms to stereo matching of images. *Pattern Recognit Lett,* No. 16, (1995) 815-821

Han, K.P.; Song, K.W.; Chung, E.Y.; Cho, S.J. & Ha, Y.H. (2001). Stereo matching using genetic algorithm with adaptive chromosomes. *Pattern Recognit,* No. 34, (2001) 1729-1740

Tang, L.; Wu, C. & Chen, Z. (2002). Image dense matching based on region growth with adaptive window. *Pattern Recognit Lett,* No. 23, (2002) 1169-1178

Lee, S.H. & Leou, J.J. (1994). A dynamic programming approach to line segment matching in stereo vision. *Pattern Recognit Lett,* Vol. 27, No. 8, (1994) 961-986

Lee, K. & Lee, J. (2004). Generic obstacle detection on roads by dynamic programming for remapped stereo images to an overhead view, *Proceedings of the IEEE International Conference on Networking, Sensing and Control*, pp. 897-902, 2004, Taipei

Nasrabadi, N.M. (1992). A stereo vision technique using curve-segments and relaxation matching. *IEEE Trans Pattern Anal Machine Intel,* Vol. 14, No 5., (1992) 566-572

Tien, F.C. (2004). Solving line-feature stereo matching with genetic algorithms in Hough space. *Journal of the Chinese Institute of Industrial Engineers,* Vol. 21, No. 5, (2004) 515-526

Pajares, G. & de la Cruz, J.M. (2004). On combining support vector machines and simulated annealing in stereovision matching. *IEEE Trans Man Cybern Part B,* Vol. 34, No. 4, (2004) 1646-1657

Candocia, F. & Adjouadi, A. (1997). A similarity measure for stereo feature matching. *IEEE Trans Image Processing,* No. 6, (1997) 1460-1464

Starink, J.P.P. & Backer, E. (1995). Finding point correspondences using simulated annealing. *Pattern Recognit,* Vol. 28, No. 2, (1995) 231-240

Wang, J.H. & Hsiao, C.P. (1999). On disparity matching in stereo vision via a neural network framework. *Proc Natl Sci Counc ROC(A),* Vol. 23, No. 5, (1999) 665-678

Zhang, P.; Lee, D.J. & Beard, R. (2004). Solving correspondence problems with 1-D signal matching, *Proceedings of the SPIE 5608*, pp. 207-217, 2004, Philadelphia

Kriegman, D.J.; Triendl, E. & Binford, T.O. (1989). Stereo vision and navigation in buildings for mobile robot. *IEEE Trans Robot Autom,* Vol. 5, No. 6, (1989) 792-803

Nitzan, D. (1988). Three-dimensional vision structure for robot application. *IEEE Trans Pattern Anal Machine Intel,* Vol. 10, No. 3, (1988) 291-309

Bruyelle, J.L. & Postaire, G.J. (1993). Direct range measurement by linear stereo vision for real-time obstacle detection in road traffic. *Robo Auton Syst,* No. 11, (1993) 261-268

Inigo, R.M. & Tkacik, T. (1987). Mobile robot operation in real-time with linear image array based vision, *Proceedings of the IEEE Intelligent Control Symposium*, pp. 228-233, 1987

Colle, O. (1990). *Vision stéréoscopique à l'aide de deux caméras linéaires: application à la robotique mobile,* PhD Thesis, Institut des Sciences Appliquées de Lyon, France

Burie, J.C.; Bruyelle, J.L. & Postaire, J.G. (1995). Detecting and localising obstacles in front of a moving vehicle using linear stereo vision. *Math Comput Model,* Vol. 22, No. 4-7, (1995) 235-246

Bruyelle, J.L. (1994). *Conception et réalisation d'un dispositif de prise de vue stéréoscopique linéaire: application à la détection d'obstacles à l'avant des véhicules routies,* PhD Thesis, Université des Sciences et Technologies de Lille, France

Deriche, R. (1990). Fast algorithms for low-level vision. *IEEE Trans Pattern Anal Machine Intel,* Vol. 12, No., 1 (1990) 78-87

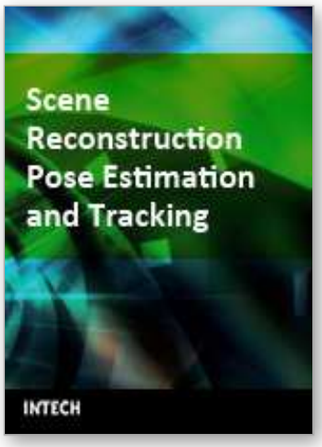
Hopfield, J.J. & Tank, T.W. (1985). Neural computation of decisions in optimization problems. *Biol Cybern,* No. 52, (1985) 141-152

Lima, P.; Bonarini, A. & Mataric, M. (2004). *Name of Book in Italics,* Publisher, ISBN, Place of Publication

Li, B.; Xu, Y. & Choi, J. (1996). Title of conference paper, *Proceedings of xxx xxx*, pp. 14-17, ISBN, conference location, month and year, Publisher, City

Siegwart, R. (2001). Name of paper. *Name of Journal in Italics,* Vol., No., (month and year of the edition) page numbers (first-last), ISSN

Arai, T. & Kragic, D. (1999). Name of paper, In: *Name of Book in Italics,* Name(s) of Editor(s), (Ed.), page numbers (first-last), Publisher, ISBN, Place of publication



Scene Reconstruction Pose Estimation and Tracking

Edited by Rustam Stolkin

ISBN 978-3-902613-06-6

Hard cover, 530 pages

Publisher I-Tech Education and Publishing

Published online 01, June, 2007

Published in print edition June, 2007

This book reports recent advances in the use of pattern recognition techniques for computer and robot vision. The sciences of pattern recognition and computational vision have been inextricably intertwined since their early days, some four decades ago with the emergence of fast digital computing. All computer vision techniques could be regarded as a form of pattern recognition, in the broadest sense of the term. Conversely, if one looks through the contents of a typical international pattern recognition conference proceedings, it appears that the large majority (perhaps 70-80%) of all pattern recognition papers are concerned with the analysis of images. In particular, these sciences overlap in areas of low level vision such as segmentation, edge detection and other kinds of feature extraction and region identification, which are the focus of this book.

InTech Europe

University Campus STeP Ri
Slavka Krautzeka 83/A
51000 Rijeka, Croatia
Phone: +385 (51) 770 447
Fax: +385 (51) 686 166
www.intechopen.com

InTech China

Unit 405, Office Block, Hotel Equatorial Shanghai
No.65, Yan An Road (West), Shanghai, 200040, China
中国上海市延安西路65号上海国际贵都大饭店办公楼405单元
Phone: +86-21-62489820
Fax: +86-21-62489821

IntechOpen