FAST AREA-BASED STEREO ALGORITHM

Master Thesis

Bratislava 2006 Michal Récky

FAST AREA-BASED STEREO ALGORITHM

MASTER THESIS

Michal Récky

FACULTY OF MATHEMATICS, PHYSICS AND INFORMATICS
COMENIUS UNIVERSITY, BRATISLAVA

Department of Applied Informatics

INFORMATICS

Supervisor : RNDr. Kateřina Dařílková

BRATISLAVA 2006

I honestly declare, that this work was elaborated independently, only with using of listed literature.

……

I would like to thank my supervisor RNDr. Kateřina Dařílková for advice and patience and my colleagues for their help.

Abstract

Area-based stereo algorithms, as the most universal of all stereovision methods, have the potential to become widely used in many industrial sectors, but their relatively low speed halt the usage. This paper describes the method which can significantly increase computing speed. This new method is called false epipolar constraint. It is an addition to epipolar constraint which can reduce the set of possible corresponding points from whole image to single line. False epipolar constraint in combination with epipolar constraint is able to reduce even this set. This method is most efficient in area-based stereo algorithms but it can be used in any application, where the epipolar constraint is used.

Keywords: epipolar, stereo, computer vision

Introduction

Stereovision is the part of computer vision inspired by nature. All higher life forms have two eyes to perceive and navigate in the 3D world. We can assume that the same can be done by computers. Thanks to the recent fast progress in digital camera technology, we have an appropriate input. All we need is an efficient algorithm. The algorithms solving the problem of extracting 3D information from multiple views are called the stereo algorithms and the field of study for this process is called stereovision. Main function of stereovision is the estimation of depth or range. It is clear that there are many applications where this estimation is required, but stereovision is not used widely. It is because simpler and mostly more accurate methods are used. Some of them were developed even before computers, like radar or sonar. Now there are even more sensors and lasers capable of performing this task. But now there are also more and more applications where stereovision takes its place. It is because the technology has finally comes to stage where digital cameras are cheap enough and computers are powerful enough to execute this task efficiently. Stereovision is most universal from range estimating methods, therefore it is used by robots performing wide array of different tasks. Main task of computer graphics is to recreate real 3D world in computer's virtual world. Commonly 3D scanners are used to transfer shapes of real object into computer's memory. Stereovision is becoming even cheaper solution, and for larger objects the only one solution. There are more applications for stereovision, but this two I consider to be the most important.

The top speed and accuracy is often required from stereo algorithms, as they have to often run in real time and even the slightest error can lead to incorrect output. Thanks to the epipolar geometry it is known that we can introduce the epipolar constraint in stereovision. This constraint can make the stereo algorithm to run faster and more accurate. A new method will be presented in this work to increase the speed and accuracy even further. This new method uses the same set of input data required for epipolar constraint to improve the performance of stereo algorithms.

Contents

Abstract iv
Introduction v
1 Overview 1
1.1 Definitions 1
1.2 Epipolar Geometry 3
1.2.1 The Parameters of the Cameras 4
1.2.2 Estimating the Fundamental Matrix 5
1.2.3 Linear Eight Point Algorithm 6
1.2.4 Fundamental Matrix Parameterization 6
1.2.5 The Uncertainty of the Fundamental Matrix 8
1.2.6 Epipolar Band 10
1.3 Solving the Correspondence Problem 12
1.3.1 Area-based Stereo Algorithms 13
1.3.2 Feature-based Stereo Algorithms 16
1.3.3 Constraints 17
2 False Epipolar Constraint 19
2.1 Method Realization 19
2.2 Method Improvement 22
2.3 False Epipolar Constraint as Weak Constrain 24
3 Points Selection 26
3.1 Critical Points Description 26
3.2 Critical Points Selection Algorithm 26
4 Algorithm Implementation 30
4.1 Input, Output 30
4.2 Implementation 30
4.2.1 Corresponding Points Detection 30
4.2.2 False Fundamental Matrix Computation 33
5 Test Results 34
6 Conclusion 39

7 References 40
Abstrakt 42

1 Overview

In this section, theory required for my research will be presented. Theme of this paper is the stereovision and stereo algorithms. All necessary concepts from this field are introduced in section 1.1 Definitions as well as the parallel between human perception and computer vision. In section 1.2 Epipolar Geometry, the model of stereo system is presented and the coherence between stereo pictures is examined more closely. In final section 1.3 Solving the Correspondence Problem the two principal groups of stereo algorithms are introduced and described.

1.1 Definitions

Process of extracting 3D information from multiple views will be described here. Usually the inputs for the process are pictures from a digital camera. In the simplest model, we need two pictures of the same scene, taken from the different positions (stereo pictures). Therefore special stereo camera or two common cameras are required. They must be synchronized to take the pictures at the same moment. To simplify our model, let's assume that both pictures have the same resolution and the parameters of the cameras are the same. It is also possible to take both pictures by one camera which has been moved to required positions and pictures were taken consequently. Restriction for this approach is that no moving objects are present in the scene, because these objects will display incorrect depth.

Let's choose a single 3D point in a real scene and let's assume that this point is visible in both taken pictures. These projections of real point are called corresponding points (see Figure 1.1).

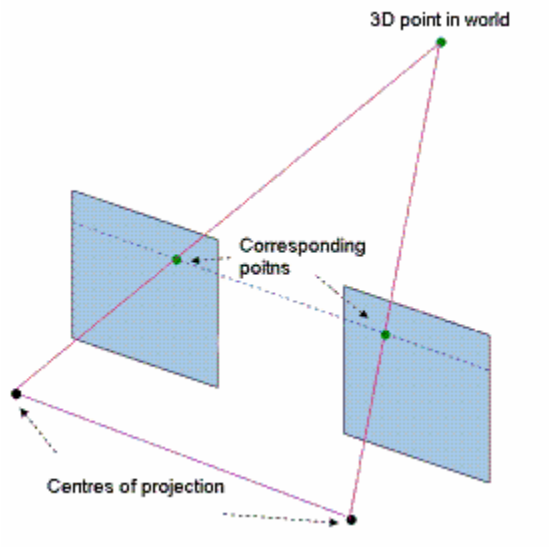

Figure 1.1: Corresponding points as projections into stereo pictures.

When these pictures are centered, we can read the position of both corresponding points in regard to the centre. Difference of these positions is called the disparity of the points. Human brain can estimate the distance of the real point from disparity and so the computer can. Theoretically when the disparity is zero, distance is infinite and as the point is closer to the observer, the disparity is rising. When we are working with digital pictures, disparities can gain only discrete values. From this feature we can see that the precision of the depth computation is limited by a picture resolution. As the resolution is rising, the precision does too, but clearly more data are required to process and the time to perform the task is rising too. It is therefore necessary to find balance between speed and precision. Commonly this balance is dependable on the given task and situation.

Before we can compute the disparity, we must find the corresponding points in both pictures. The detection of corresponding points is the critical step of all stereo algorithms, because the highest possible speed and precision is required. This task is described in section 1.3 (Solving the Correspondence Problem).

So let's assume that we can detect corresponding points correctly. Identifying the object to which these points belong is much more complicated task. This is another situation where human sight is different from the computer vision. When we look at the

scene we don't see points in space, but we can see whole 3D objects situated in space. One possible explanation for this feature will be described now. Our brain can choose some critical points from the scene, situate them into the space and create a simple geometric model. Based on the model's features, unseen parts are estimated and object is categorized. All this is done by previous experience with real world [1]. Trying to follow this approach in computer vision, we will need advanced artificial intelligence with large database of learned knowledge. Current level of technology cannot provide this, so we will have to choose different approach. We can either find as many points as possible and situate them into 3D space or we can try to find some critical points and form a mesh from them. We cannot estimate the unseen parts or segment the objects in the scene without some advanced learning algorithm. This algorithm is out of range of this article and will not be considered here.

1.2 Epipolar Geometry

Epipolar geometry was founded to examine the coherence between pair of stereo pictures and exploit the possibility to recreate 3D view from them. The only geometric constraint which can be derived from the pair of pictures is given by epipolar geometry. The fact that the point in the scene and the two optical centers lie on the same plane shows us, that for every point in first picture, its correspondence must be located on a known line in the second picture. This constraint is dependent only on the positions and the parameters of the cameras, not on the structure of the scene. Dependence was first formulated by French mathematician Chasles in 1855 who introduced seven points algorithm to obtain the geometry. Eight points algorithm was developed later to solve this problem. This approach will be described later in section 1.2.3 (Linear Eight Point Algorithm). But first let's have a closer look at the main equation derived from epipolar geometry:

$$\mathbf{x'}^{T}\,\mathbf{F}\,\mathbf{x} = \mathbf{0} \qquad (1.1)$$

Here $\mathbf{x}=(x_1, x_2, 1)^T$ represents the projective coordinates of single point from first picture and $\mathbf{x'}=(x_1', x_2', 1)^T$ are coordinates of its corresponding point from second picture (so they are the projections of the same real point in the taken scene). **F** is the 3x3 matrix called the **fundamental matrix**. We know from epipolar geometry that for each two stereo pictures, there is exactly one fundamental matrix which can be applied for every point in the pictures. Also for each point **x** from first picture there is a line in the second picture given by vector **Fx**. This line is called the epipolar line (see Figure 1.2). It is obvious from (1.1) that the corresponding point for **x** is located in its epipolar line[4].

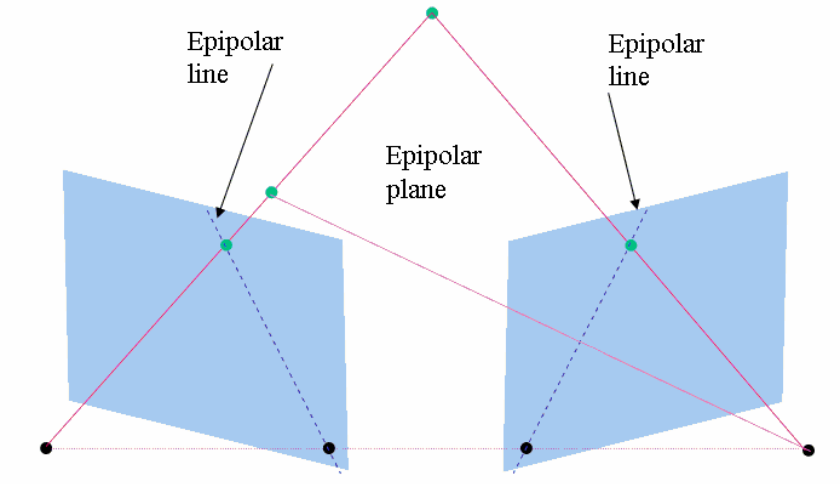

Figure 1.2: Epipolar lines as intersection of epipolar planes on images.

1.2.1 The Parameters of the Cameras

Geometric coherence of stereo pictures is dependent only on the parameters of the cameras and their relative positions, so the fundamental matrix can be derived from them. For each calibration of camera we can obtain the calibration matrix **K**:

$$\mathbf{K} = \begin{bmatrix} a_x & c & t_x \\ 0 & a_y & t_y \\ 0 & 0 & 1 \end{bmatrix}$$

a_x, a_y – defines scaling functions of the camera in x and y axis.

(t_x, t_y) – optic centre coordinates

c – distortion parameter

When the picture is taken by camera, there is a coherence for each point $\mathbf{X}=(x_1, x_2, x_3, 1)^T$ in real scene and its image point $\mathbf{x}$ in taken picture. It can be formulated by equation : $\mathbf{x} = \mathbf{PX}$

$\mathbf{P}$ is a 3x4 matrix called the camera's matrix.

Let's assume that we have two cameras ready to take a pair of pictures. When the coordinates system of the first camera is identified with the world coordinates system, the cameras' matrices can be described as $\mathbf{P}=\mathbf{K}[\mathbf{I}_{3x3}|\mathbf{0}]$ and $\mathbf{P'}=\mathbf{K'}[\mathbf{R}|\mathbf{t}]$ where $\mathbf{R}$ is the rotation and $\mathbf{t}$ is the translation of the second camera in regard to the first camera. $\mathbf{K}$ and $\mathbf{K'}$ are the calibration matrices of the cameras. Then the fundamental matrix can be expressed as:

$$\mathbf{F}=[\mathbf{e'}]_x \mathbf{P'}\ \mathbf{P}^+ \qquad (1.2)$$

Where $\mathbf{e'}$ is called the epipol – the optical centre of first camera projected into plane image of the second camera. $\mathbf{P}^+$ is the pseudoinverse matrix to $\mathbf{P}$ and $[\ ]_x$ is the following transform:

$$[\mathbf{a}]_x = \begin{bmatrix} 0 & -a_3 & a_2 \\ a_3 & 0 & -a_1 \\ -a_2 & a_1 & 0 \end{bmatrix}$$

if $\mathbf{a}=(a_1, a_2, a_3)^T$ is a vector.

1.2.2 Estimating the Fundamental Matrix

In many situations, the calibration matrices of the cameras are unknown, or they cannot be identified at the moment. Other methods can be used to estimate the fundamental matrix. Usually these methods need some already recognized corresponding points. They are divided into following groups:

- linear algorithms
- algebraic minimalization algorithms

- difference minimalization algorithms

1.2.3 Linear Eight Point Algorithm

Previously mentioned eight point algorithm will be described in this section. The basic idea of linear methods is that if enough corresponding points has been detected, unknown components of fundamental matrix can be computed by substituting these points into given equations. This idea was first used by Longued – Higgins [8].
The equation (1.1) can be formulated as:

$$x_1'xF_{11} + x_1' x_2F_{12} + x_1'F_{13} + x_2'x_1F_{21} + x_2 x_2'F_{22} + x_2'F_{23} + x_1F_{31} + x_2F_{32} + F_{33} = 0 \quad (1.3)$$

where $\mathbf{F}=(F_{ij})_{i,j=1..3}$ is fundamental matrix, $\mathbf{x}=(x_1, x_2, 1)$ and $\mathbf{x'}=(x_1', x_2', 1)$ are the corresponding points. Equation is linear and homogenous in 9 unknown coefficients of fundamental matrix. Therefore we need 8 pairs of corresponding points to find a unique solution. To increase the robustness of this approach we can use more then 8 points. In this case we will have more equations then unknowns and we can use the error minimization methods [6].

1.2.4 Fundamental Matrix Parameterization

The basic idea of fundamental matrix parameterization is to use seven scalars which represents the epipolar geometry of stereo pictures. When the epipoles from both pictures has the finite coordinates, we can use these affine coordinates to represent the parameterization. In practice this method will become unstable when the epipoles are at infinity [6]. To solve this problem the normalization of projective coordinates of the epipoles by one of them can be used. In this section, we will assume that the rank of fundamental matrix **F** is 2. This constraint is often required for fundamental matrix and it can be enforced by approximation (using the closest rank 2 matrix) for example. Than we can introduce these equations:

$$\exists(j_1, j_2, j_3) \in [1, 3], \exists(\lambda_{j2}, \lambda_{j3}) \in \mathbf{R}^2; \quad \mathbf{c_{j1}} + \lambda_{j2}\mathbf{c_{j2}} + \lambda_{j3}\mathbf{c_{j3}} = 0$$
$$\exists(i_1, i_2, i_3) \in [1, 3], \exists(\mu_{i2}, \mu_{i3}) \in \mathbf{R}^2; \quad \mathbf{r_{i1}} + \mu_{i2}\mathbf{r_{i2}} + \mu_{i3}\mathbf{r_{i3}} = 0 \tag{1.4}$$

Where $\mathbf{c_1}$, $\mathbf{c_2}$, $\mathbf{c_3}$ are the columns and $\mathbf{r_1}$, $\mathbf{r_2}$, $\mathbf{r_3}$ are the rows of fundamental matrix. $\mathbf{F}=(F_{ij})_{i,j=1..3}$ can be parameterized by the 8-vector $\mathbf{f_8}=[\ \lambda_{j2},\ \lambda_{j3},\ \mu_{i2},\ \mu_{i3},\ F_{i2j2},\ F_{i2j3},\ F_{i3j2},\ F_{i3j3}]^{\mathrm{T}}$.

Function that compute $\mathbf{F}$ from $\mathbf{f_8}$ is denoted as $F_{i1j1}(\mathbf{f_8})$ and computed with $i_1=3$ and $j_1=3$ for example as:

$\mathbf{F} = F_{i1j1}(\mathbf{f_8}) =$

$$\begin{bmatrix} F_{i2j2} & F_{i2j3} & -\lambda_{i2}F_{i2j2} - \lambda_{i3}F_{i2j3} \\ F_{i3j2} & F_{i3j3} & -\lambda_{i2}F_{i3j2} - \lambda_{i3}F_{i3j3} \\ -\mu_{i2}F_{i2j2} - \mu_{i3}F_{i3j2} & -\mu_{i2}F_{i2j3} - \mu_{i3}F_{i3j3} & \begin{matrix}\mu_{i2}(\lambda_{i2}F_{i2j2} + \lambda_{i3}F_{i2j3}) + \\ \mu_{i3}(\lambda_{i2}F_{i3j2} + \lambda_{i3}F_{i3j3})\end{matrix} \end{bmatrix} \tag{1.5}$$

But the fundamental matrix is defined only up to scale factor, so only three of the four elements can define the epipolar geometry. Let $i_{23} \in \{i_2, i_3\}$ and $j_{23} \in \{j_2, j_3\}$ be such that $Fi_{23},j_{23} \neq 0$, then the parameterization can be transformed:

$$\left[\lambda_{j2}, \lambda_{j3}, \mu_{i2}, \mu_{i3}, \frac{F_{i2j2}}{F_{i23j23}}, \frac{F_{i2j3}}{F_{i23j23}}, \frac{F_{i3j2}}{F_{i23j23}}, \frac{F_{i3j3}}{F_{i23j23}}\right]^{\mathrm{T}} \tag{1.6}$$

where one of the last four elements is equal to 1. Therefore only 7 elements can be used for parameterization when four indices i_1, j_1, i_{23}, j_{23} are known:

$\mathbf{f_7} = [\ \lambda_{j2}, \lambda_{j3}, \mu_{i2}, \mu_{i3}, a, b, c]^{\mathrm{T}}$

Consequently matrix can be computed as $\mathrm{F_{i1j1i23j23}}(\mathbf{f_7})$. If $i_1=3$, $j_1=3$, $i_{23}=i_2=1$ and $j_{23}=j_3=2$ then:

$\mathbf{F} = F_{i1j1i23j23}(\mathbf{f_7}) =$

$$\begin{bmatrix} F_{i2j2} & 1 & -\lambda_{i2}F_{i2j2} - \lambda_{i3} \\ F_{i3j2} & F_{i3j3} & -\lambda_{i2}F_{i3j2} - \lambda_{i3}F_{i3j3} \\ -\mu_{i2}F_{i2j2} - \mu_{i3}F_{i3j2} & -\mu_{i2} - \mu_{i3}F_{i3j3} & \begin{matrix}\mu_{i2}(\lambda_{i2}F_{i2j2} + \lambda_{i3}) + \\ \mu_{i3}(\lambda_{i2}F_{i3j2} + \lambda_{i3}F_{i3j3})\end{matrix} \end{bmatrix} \tag{1.7}$$

1.2.5 The Uncertainty of the Fundamental Matrix

To better understand the stability of fundamental matrix, computation of uncertainty of fundamental matrix from the uncertainty of corresponding points is presented in this section. Functions needed for this computation will be introduced now:

The covariance matrix of any random vector **y** is defined as:

$$\Lambda_{\mathbf{y}} = E[(\mathbf{y} - E[\mathbf{y}])(\mathbf{y} - E[\mathbf{y}])^T], \tag{1.8}$$

where E[**y**] is the mean of the **y**.

In case of explicit function $\mathbf{y}=\varphi(\mathbf{x})$ the covariance matrix is computed as:

$$\Lambda_{\mathbf{y}} = D\varphi(E[\mathbf{x}])\ \Lambda_{\mathbf{x}}\ D\varphi(E[\mathbf{x}])^T, \tag{1.9}$$

where $D\varphi(E[\mathbf{x}])$ is the Jacobian matrix of φ evaluated at E[**x**].

The uncertainty of fundamental matrix can be presented as implicit error function. Therefore the covariance of this kind of function will be examined more closely:

Proposition 1.1: *Let an error function* $C:\mathbf{R}^m x \mathbf{R}^q \rightarrow \mathbf{R}$ *be a function of class* C^∞, $x_0 \in \mathbf{R}^m$ *be the measurement vector and* $y_0 \in \mathbf{R}^q$ *be a local extremum of* $C(x_0, z)$. *If the Hessian* **H** *of C with respect to z is invertible at* $(x, z) = (x_0, y_0)$ *then there exists an open set U' of* $\mathbf{R}^m$ *containing* x_0 *and an open set U'' of* R^q *containing* y_0 *and a* C^∞ *mapping* $\varphi:\mathbf{R}^m \rightarrow \mathbf{R}^q$ *such that for (x, y) in U'xU'' the two relations ''y is a local extremum of C(x,z) with respect to z'' and* $y=\varphi(x)$ *are equivalent. Furthermore, we have the equation for function jacobian*

$$D\varphi(\mathbf{x}) = -\mathbf{H}^{-1}\frac{\partial \mathbf{\Phi}}{\partial \mathbf{x}}, \tag{1.10}$$

where $\mathbf{\Phi} = \left(\frac{\partial C}{\partial \mathbf{z}}\right)^T$ *and* $\mathbf{H} = \frac{\partial \mathbf{\Phi}}{\partial \mathbf{z}}$.

Taking x_0=E[**x**] and y_0=E[**y**], we will get the value of covariance from (1.9) and (1.10) as:

$$\Lambda_{\mathbf{y}} = \mathbf{H}^{-1} \frac{\partial \mathbf{\Phi}}{\partial \mathbf{x}} \Lambda_{\mathbf{x}} \frac{\partial \mathbf{\Phi}^T}{\partial \mathbf{x}} \mathbf{H}^{-T} \tag{1.11}$$

The error function relevant for fundamental matrix can be expressed as C(**x**, $\mathbf{f_7}$) where **x**=($\mathbf{x_1}$, $\mathbf{x_1}$', …,$\mathbf{x_p}$, $\mathbf{x_p}$') is the set of corresponding points used for matrix computation (p=8 for 8-point algorithm) and $\mathbf{f_7}$ is the parameterization of fundamental matrix described in previous section. Covariance $\Lambda_{\mathbf{F}}$ of fundamental matrix **F** is then computed as:

$$\Lambda_{\mathbf{F}} = \frac{d\mathbf{F}_{i1j1i23j23}(\mathbf{f_7})}{d\mathbf{f}_7} \Lambda_{\mathbf{f}_7} \frac{d\mathbf{F}_{i1j1i23j23}(\mathbf{f_7})^T}{d\mathbf{f}_7} \tag{1.12}$$

Where $\frac{d\mathbf{F}_{i1j1i23j23}(\mathbf{f_7})}{d\mathbf{f}_7} = \frac{d\mathbf{F}_{i1j1}(\mathbf{f_8})}{d\mathbf{f}_8} \frac{d\mathbf{f_8}(\mathbf{f_7})}{d\mathbf{f}_7}(\mathbf{f_7})$ is the Jacobian matrix of the parameterization.

$\mathbf{f_8}(\mathbf{f_7})$ is the function that computes $\mathbf{f_8}$ from a given $\mathbf{f_7}$. For $i_{23}=i_2=1$ and $j_{23}=j_3=2$, for example, we have

$\mathbf{f_8}(\mathbf{f_7})$=[λ_{j2}, λ_{j3}, μ_{i2}, μ_{i3}, F_{i2j2}, *1*, F_{i3j2}, F_{i3j3}]$^{\mathrm{T}}$

and for i_1=3 and j_1=3: $\frac{d\mathbf{F}_{i1j1}(\mathbf{f_8})}{d\mathbf{f_8}} =$

$$\begin{bmatrix}
0 & 0 & -F_{i2j2} & 0 & 0 & -F_{i3j2} & 0 & 0 & F_{i2j2}\mu_{i2} + F_{i3j2}\mu_{i3} \\
0 & 0 & -F_{i2j3} & 0 & 0 & -F_{i3j3} & 0 & 0 & F_{i2j3}\mu_{i2} + F_{i3j3}\mu_{i3} \\
0 & 0 & 0 & 0 & 0 & 0 & -F_{i2j2} & -F_{i2j3} & F_{i2j2}\lambda_{j2} + F_{i2j3}\lambda_{j3} \\
0 & 0 & 0 & 0 & 0 & 0 & -F_{i3j2} & -F_{i3j3} & F_{i3j2}\lambda_{j2} + F_{i3j3}\lambda_{j3} \\
1 & 0 & -\lambda_{j2} & 0 & 0 & 0 & -\mu_{i2} & 0 & \lambda_{j2}\mu_{i2} \\
0 & 1 & -\lambda_{j3} & 0 & 0 & 0 & 0 & -\mu_{i2} & \lambda_{j3}\mu_{i2} \\
0 & 0 & 0 & 1 & 0 & -\lambda_{j2} & -\mu_{i3} & 0 & \lambda_{j2}\mu_{i3} \\
0 & 0 & 0 & 0 & 1 & -\lambda_{j3} & 0 & -\mu_{i3} & \lambda_{j3}\mu_{i3}
\end{bmatrix}^T \tag{1.13}$$

Equation (1.12) is required for the computation of epipolar band and is the main result of this section.

1.2.6 Epipolar Band

As described in section 1.2 (Epipolar Geometry), the vector **Fx** for point **x** and fundamental matrix **F** represents the epipolar line in second picture on which the corresponding point to **x** is located. But when the fundamental matrix is computed from the coordinates of corresponding points (by linear algorithm for example), it is clear that these values are not absolutely accurate, as the coordinates on digital pictures are discrete. Also the detection of these points may not to be accurate. Therefore the corresponding point is not always located exactly on the epipolar line. It is located inside the area around epipolar line called epipolar band. This area can be computed from the uncertainty of fundamental matrix (1.12).

At first we will define the ellipse of uncertainty:

Proposition 1.2: *If y follows a Gaussian distribution, then the probability that y lies inside the k-hyper-ellipsoid is defined by the equation*

$$\left(y - E[y]\right)^T \Lambda_y^{-1} \left(y - E[y]\right) = k^2 \tag{1.14}$$

where k is any scalar.

In two-dimensional case, the k-hyper-ellipsoid has the form of ellipse. In future text this ellipse will be called the ellipse of uncertainty.

Let $\mathbf{x}_0=(x,\ y,\ 1)^{\mathrm{T}}$ be a point in the first picture and **F** a fundamental matrix obtained from the linear algorithm. A coordinate vector of epipolar line $\mathbf{l}_0=(l_{01},\ l_{02},\ l_{03})^{\mathrm{T}}$ of $\mathbf{x}_0$ is the vector $\mathbf{l}_0= \mathbf{F}\mathbf{x}_0$. The covariance matrices $\Lambda_{\mathbf{x}0}$ of $\mathbf{x}_0$ and $\Lambda_{\mathbf{F}}$ of **F** are computed as described in previous sections (1.8), (1.12). Covariance $\Lambda_{\mathbf{l}_0}$ is then computed as:

$$\Lambda_{\mathbf{l}_0} = \frac{\partial \mathbf{l}_0}{\partial \mathbf{F}} \Lambda_{\mathbf{F}} \frac{\partial \mathbf{l}_0^{\ T}}{\partial \mathbf{F}} + \mathbf{F} \Lambda_{\mathbf{x0}} \mathbf{F}^T \tag{1.15}$$

Resulting object is a 3x3 covariance matrix of epipolar line. The ellipse of uncertainty for the epipolar line can be defined from (1.15). But first, to compute the ellipse of

uncertainty we need to reduce the dimension of covariance matrix (1.15) by one. Therefore following substitution is performed:

$$\Lambda_{\hat{\mathbf{l}}_0} = \frac{\partial \hat{\mathbf{l}}_0}{\partial \mathbf{l}_0} \Lambda_{\mathbf{l}_0} \frac{\partial \hat{\mathbf{l}}_0}{\partial \mathbf{l}_0}^T \tag{1.16}$$

where $\hat{\mathbf{l}}_0 = \left(\frac{l_{01}}{l_{03}}, \frac{l_{02}}{l_{03}} \right)^T$.

Consequently the ellipse of uncertainty can be defined:

$$\left(\hat{\mathbf{l}} - \hat{\mathbf{l}}_0\right)^T \Lambda_{\hat{\mathbf{l}}_0}^{-1} \left(\hat{\mathbf{l}} - \hat{\mathbf{l}}_0\right) = k^2 \tag{1.17}$$

The geometric interpretation of this ellipse of uncertainty is that the epipolar line **l** of point **x** has a probability defined by equation (1.14) of "falling" into the ellipse defined by equation (1.17). As ellipse is not presented in image space, the meaning of "falling" is actually accurate in dual space (defined by transform: $(l_{01}, l_{02}, l_{03})^T \rightarrow \left(\frac{l_{01}}{l_{03}}, \frac{l_{02}}{l_{03}} \right)^T$).

To use this interpretation in image space, dual conic to ellipse has to be used.

Proposition 1.3: *The conic C^* dual to the ellipse of uncertainty C is a hyperbola, a parabola or an ellipse according the fact that the point z (representing a line in dual space) is inside, on or outside C.*

In case of epipolar line $\mathbf{l}_0=(l_{01}, l_{02}, l_{03})^T$, the dual conic is hyperbola if the point $\hat{\mathbf{l}}_0 = \left(\frac{l_{01}}{l_{03}}, \frac{l_{02}}{l_{03}} \right)^T$ is located inside the ellipse of uncertainty. Inner area of this hyperbola is called the epipolar band. This area can be also defined by following definition:

Definition: *The epipolar band is the set of lines in the image that correspond to the points in the dual plane inside the ellipse of uncertainty.*

As the ellipse of uncertainty was computed from uncertainty of fundamental matrix **F**, following the vector **Fx** for epipolar line, the area of epipolar band is the location of corresponding point to **x**.

In practice, computing of epipolar band for each point is too time demanding. Instead the search area is limited by constant distance from epipolar line. The theory of epipolar band is important for investigating the stability of fundamental matrix.

The most important conclusion for my work is: *Maximal distance of corresponding point to epipolar line is directly proportional to the level of uncertainty of corresponding points from which the fundamental matrix was computed.*
The theory for false epipolar constraint described in section 2 is based upon this assumption.

1.3 Solving the Correspondence Problem

Many methods capable of corresponding points detection were developed. Commonly they can be divided into two different categories:

Area-based stereo

- For single position in first image search for the similar location in second image.
- Dense set of corresponding points is found.

Feature-based stereo

- Find objects with unique features in both pictures and match them based on features.
- Sparse set of corresponding points is found.

Many experiments have been conducted to find out which approach is used in natural human vision. The first idea was that features are detected and then the depth is estimated. An example of this is suggested by walking into room full of books. At first, each book is identified and categorized as "book", then they are matched in both images. This theory was later disproved by stereograms [3]. Stereogram is an image consisting of repeating patterns. These patterns seem to be placed randomly or in exact rows, but a hidden order invoking the disparity properties is included. Single Image Random-Dot stereogram consisting of two images (one for each eye) was used for experiment. Even there were no features and the dots were placed "randomly", the depth was recognized. The absence of any feature in stereograms gives us the clue, that human depth recognition is area-based. Let's have a closer look at both methods in computer vision approach.

1.3.1 Area-based Stereo Algorithms

Idea of area-based methods is based on an assumption that we can somehow compute the similarity of two locations in different pictures. Then we can assume that the corresponding points will have similar neighbourhoods. Measure this similarity is not trivial task, because for one location in the first picture there can be multiple similar locations in the second picture. Therefore additional problems must be solved:

What is the measure of similarity?

- Similarity is computed in local neighbourhood (see Figure 1.3).
- Extend of the neighbourhood has an impact on the accuracy of the method. If it is possible, it should be depended on the size of the objects in picture.

Where in the image can be the similarity measure carried out successfully?

- Changes in color, texture or brightness are the sources of local inhomogenity. Analysis can be successful in these regions.
- Homogenous regions are less likely to be reliable.

How are the corresponding location found from the similarity values?

- This is dependent on the method of similarity measuring.

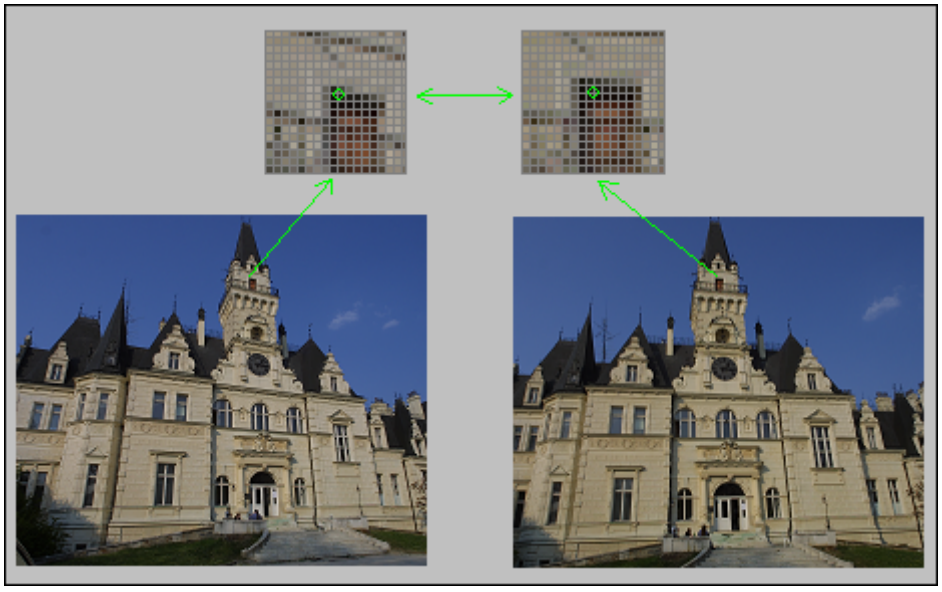

Figure 1.3: Principle of area-based stereo. Neighbourhoods of two points are displayed in grids. Corresponding points are in the centre of these grids.

Majority of the area-based methods uses the correlation measure. Algorithms in this group are called correlation-based stereo algorithms [5].

Correlation Measure:

Correlation is computed from the covariance values.

$$C = \frac{\sigma_{LR}^2}{\sqrt{\sigma_L^2 \sigma_R^2}}$$

Different variables derived from pictures properties can be substituted for covariance σ. In this diploma work, intensity difference was used in algorithm described in section 4 (Algorithm Implementation).

Intensity Difference:

Brightness (intensity) values of examined points neighbourhoods are differenced. In this approach, the similarity is computed from intensity values, so the assumption has to be made, that these values are not very different in the corresponding points neighbourhoods. This is the case when both pictures have not very different illuminations and when there are not specular reflections in the pictures. When intensity differences are substituted into correlation scheme, the similarity value for area-based algorithm is computed [7].

$$C(x,y,d,e) = \frac{\sum_{(\xi,\eta)} \left[\mathrm{I}_1(x+\xi, y+\eta) - \mathrm{I}_\mathrm{r}(x+\xi+d, y+\eta+e) \right]^2}{\sqrt{\sum_{(\xi,\eta)} \mathrm{I}_1(x+\xi, y+\eta)^2 \sum_{(\xi,\eta)} \mathrm{I}_\mathrm{r}(x+\xi+d, y+\eta+e)^2}}$$

where $\xi \in [-n, n], \quad \eta \in [-m, m]$ define the size of neighbourhood around the point, $\mathrm{I_p}(x, y)$ is the value of intensity for point (x, y) in p=1(left) or p=r(right) picture and d, e define the disparity.

As there are high time consumptions operators (square root, square) used in this equation, simplification can be made to speed up the algorithm.

$$C = \sum_{i=1}^{m} \sum_{j=1}^{n} \left(\mathrm{I}_1(i,j) - \mathrm{I}_\mathrm{r}(i,j) \right)^2$$

This modification will make the algorithm to run faster, but it will decrease the reliability of correct detection.

There has been developed some methods of corresponding points detection based on neural networks. As these methods are working with points neighbourhoods, they can be ranked as area-based methods. For example:
PCA (Principal Components Analysis) – similarity of two locations is computed through dimension reduction into eigenspace [9].

In general, area-based methods are not as fast as feature-based. But current level of technology allows some of them to run even in real time applications, which are important for robot control or virtual reality [10], [5]. The very basic idea of computing the disparities from local information (neighbourhood) makes them quite inaccurate. Output of the area-based methods is a dense set of corresponding points, which can be used to create disparity map. This map is usually 2D matrix of depth values. It can provide some information about close objects, but it is not a 3D model of the scene. Disparity map can provide enough information for robot control (see Figure 1.4).

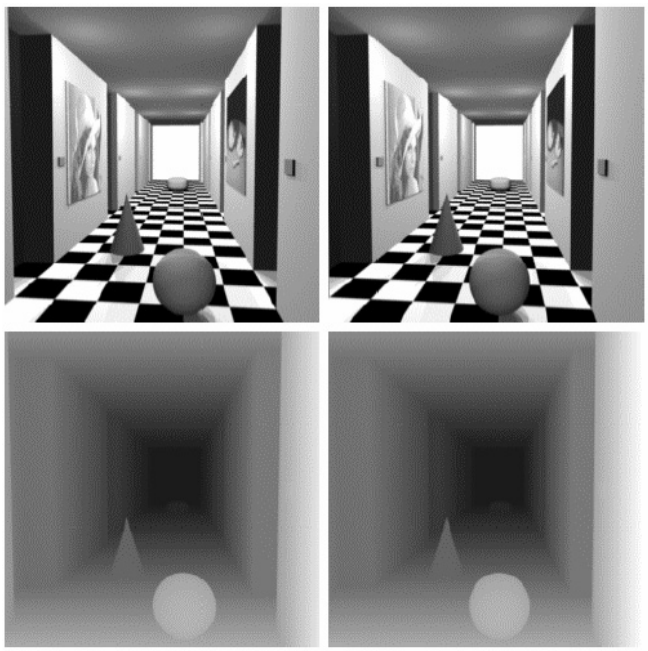

Figure 1.4: Disparity maps derived from area-based algorithm.

Also the points can be situated into space and create a 3D data visualization. This can provide the base for segmentation or even object reconstruction.

1.3.2 Feature-based Stereo Algorithms

Many properties can be used to describe objects in the scene. Some of them can stay relatively unchanged in the different views. This is the idea of the feature-based methods. Gradient magnitude, orientation of incident edges, corners shapes or curvature can be used to describe some areas in both images. Consequently objects with similar features are matched in both pictures (see Figure 1.5). The corresponding points can be computed very accurate in these objects. Feature-based method is usually even faster then area-based and the probability of the misplaced corresponding points is lower. However, only points on specific objects, which were matched, can be located, therefore the set of the corresponding points is very sparse. Usually the corresponding points are located in clusters (matched objects) [11]. Points between matched objects must be computed with another method. Commonly different kind of interpolation based on disparity gradient or the relative positions of the neighbours can be used. [6] Also the problem with many similar objects in the image can appear, as they can have similar features.

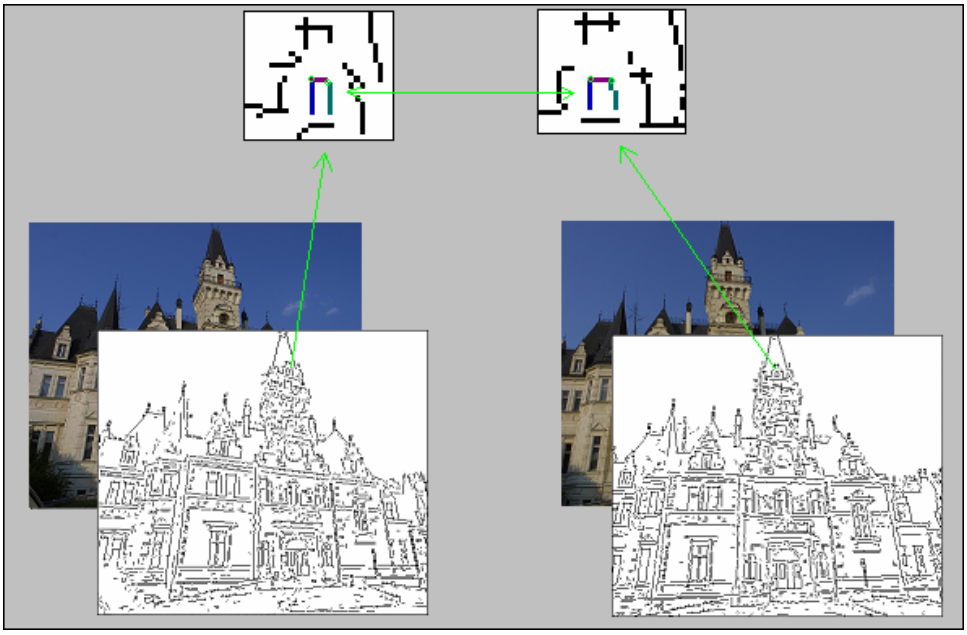

Figure 1.5: Principle of feature-based stereo algorithms. Three edges were detected (marked with different colors) and matched based on their incidence.

Type of objects and their features:

- **Edge elements** - edges incidence or orientation
- **Corners** - coordinates of corners, type of junctions that the corners correspond to (e.g. Y-junction, L-function, A-junction, etc.)
- **Line segments** - coordinates of end-points, mid-points, orientation of line segments
- **Regions** - interiors\borders – areas, bounding lines, centroids.
- **Curve segments, circles, ellipses** - used only in special cases (industrial automation)

1.3.3 Constraints

If we know some additional information about the pictures or stereo system, we can extract some special properties of corresponding points. Thanks to these properties, we can define areas in pictures where the corresponding points should be located, or

discover already detected incorrect correspondences. These properties can be usually formulated as constraints. They are used in both area-based and feature-based algorithms, but different way. Some of the constraints will be presented here:

- **Uniqueness** - Almost always, a given pixel or feature from one image can match *no more than one* pixel or feature from the other image [10].
- **Continuity** - The cohesiveness of matters suggests that the disparity of the matches should vary smoothly almost everywhere over the image.
- **Ordering** – If point *m* is *to the left* of *n* then it's corresponding point *m*' should also be *to the left* of *n*'.
- **Left-Right consistency** – Corresponding point *m'* in second picture should be confirmed by searching it's corresponding point in first picture [7].

These are the most common constraints and are called weak constraints, as they are not always valid. Their usage is limited to some special cases (like near-based stereo). There is only one strong constraint. It is derived from epipolar geometry (see section 1.2).

- **Epipolar Constraint** - Given a point *m* in the left image, the corresponding point *m*' must lie on the epipolar line.

Constraints are the most effective way how to speed up stereo algorithm and how to increase its accuracy.

2 False Epipolar Constraint

A new constraint as an addition to epipolar constraint will be presented in this section. Thanks to this method, the provable area on epipolar line with corresponding point can be highlighted. The area can vary from 1/3 to 1/10 size of original epipolar line. As a result, the algorithm does not need to search for corresponding point on the whole epipolar line but only on this smaller section. Obviously this can speed-up whole process significantly and increase the reliability because some incorrect correspondences may be eliminated from the searching area.

2.1 Method Realization

The method is based on the existence of so-called False Fundamental Matrices (FFM). While correct fundamental matrix should be computed from accurate corresponding points, FFMs are computed from error input deliberately. They can be computed by the method commonly used for numerical computation of fundamental matrix known from epipolar geometry. The 8-point algorithm described in section 1.2.3 (Linear Eight Point Algorithm) can be modified to compute FFM. In my work 9-point modification was used. The structure of the original algorithm [2]:

1) As an input we have 9 pairs of corresponding points already detected in the pictures.

2) Equation $\mathbf{x'}^{T}F\mathbf{x}=0$ can be rewritten as:

$$x_1'xF_{11} + x_1' x_2F_{12} + x_1'F_{13} + x_2'x_1F_{21} + x_2 x_2'F_{22} + x_2'F_{23} + x_1F_{31} + x_2F_{32} + F_{33} = 0$$

where $\mathbf{F}=(F_{ij})_{i,j=1..3}$ is fundamental matrix, $\mathbf{x}=(x_1, x_2, 1)^T$ and $\mathbf{x'}=(x_1', x_2', 1)^T$ are the coordinates of corresponding points.

3) After substituting all 9 corresponding points into this equation, we will get 9 equations with 9 unknown variables.

4) Solving this system (by SVD-decomposition for example [12]) will lead to fundamental matrix.

To get a false fundamental matrix, we have to do only simple input modification. We must change the coordinates of all 9 original points

$\mathbf{x}=(x, y, 1)^T$ to $\mathbf{x}_\varepsilon=(x+\varepsilon_1, y+\varepsilon_2, 1)^T$

and their corresponding points

$\mathbf{x'}=(x',y',1)^T$ to $\mathbf{x'}_\varepsilon=(x'+\varepsilon'_1, y'+\varepsilon'_2, 1)^T$

where $\varepsilon_1, \varepsilon_2, \varepsilon'_1, \varepsilon'_2$ are relative small numbers dependent on picture resolution. We can consider to bound $\varepsilon_1, \varepsilon_2, \varepsilon'_1, \varepsilon'_2$ in intervals described in Table 1.

Resolution	**Interval**
640x480	<-1, 1>
800x600	<-1.5, 1.5>
1024x768	<-2, 2>
1280x1024	<-2.3, 2.3>

Table 1: Interval – resolution dependencies for corresponding point modifications.

It is important, that for each point these variables are different and are distributed through the interval uniformly. The simplest way is to choose the variables randomly from the interval. Another solution is to predefine these variables, or design a function to compute them. Example of one possible function:

$\varepsilon_1 = \cos(2\pi / 9 * i)*d$

$\varepsilon_2 = \sin(2\pi / 9 * i)*d$

$\varepsilon'_1 = \cos(2\pi / 9 * i + \pi/2)*d$

$\varepsilon'_2 = \sin(2\pi / 9 * i + \pi/2)*d$

where ε_1, ε_2, ε'_1, ε'_2 are the variables modifications for i^{th} point (i=1..9). Variable d is the highest bound of interval described in Table 1.

Algorithm with this input modification will compute false fundamental matrix **F'** (fundamental matrix with slightly changed input). This matrix has some important properties:

1) Corresponding point for **x** should be located near false epipolar line given by vector **F'x**. The maximal distance of corresponding point from false epipolar line is directly proportional to maximal values of ε_1, ε_2, ε'_1, ε'_2 (see section 1.2.6).

2) False epipolar line should have different direction than the original epipolar line. If the direction is similar, another false fundamental matrix can be computed from different values of ε_1, ε_2, ε'_1, ε'_2.

From these two properties, we can see that the corresponding point should be located near to the intersection of both lines (see Figure 2.1.). This intersection can define an area in epipolar line, where we can search for corresponding point. A size of the area is dependent on a current task, or the algorithm requirements. Increasing the area size the reliability of algorithm is raising, but also the computing time raises.

Figure 2.1: A light line is the epipolar line and a dark one is the false epipolar line. The corresponding point is located near their intersection.

2.2 Method Improvement

The critical question is: How far the corresponding point is located from the epipolar lines intersection? As described in section 1.2.6 (The Epipolar Band), this distance can be computed precisely. In practice, the computation of this value is so complicated and time demanding, that final method speed-up will be minimal, or none. Therefore this distance should be estimated or approximated. In most cases the intersection is only few pixels far away from corresponding point, but there are usually some specific areas in the image, where intersection is more distant from the point. Existence and location

of these areas is caused by the selection of particular nine points used for matrix computation, but it is hard to predict these areas just from this selection. Another problem appears when the false epipolar line and the original epipolar line have similar directions. Corresponding point is always located in some distance from false epipolar line. As the angle of false epipolar line with correct epipolar line is decreasing (lines are becoming parallel), the distance is causes that the intersection moves away from the correct corresponding point. The best solution for these problems is to apply more false fundamental matrices at once. Some degrees of freedom were added using the algorithm from previous section – in computing the variables ε_1, ε_2, ε'_1, ε'_2. Different methods or functions can be used to compute these variables. Several different matrices can be computed this way from the same set of points so that no additional data are required for this extension. More matrices will guarantee a greater chance that at least one of the intersections is located near the corresponding point. We can also eliminate false epipolar lines which forms a low angle with original epipolar line. During the testing 5 - 10 false fundamental matrices proved to be sufficient to solve the problems. The algorithm was modified to find a left-most and a right-most intersection of the false epipolar lines with the epipolar line. These two boundaries will give us interval, where the corresponding point could be located (see Figure 2.2). The interval can be increased by some fraction to prevent any omission of the corresponding point.

Intersection of a single false epipolar line with the "correct" epipolar line is either on the left or on the right side from the corresponding point. Let's consider that it is on the left side. This matrix was computed thanks to ε_1, ε_2, ε'_1, ε'_2 parameters modification to corresponding points as described in the previous section. False fundamental matrix computed by using negated parameters: $-\varepsilon_1$, $-\varepsilon_2$, $-\varepsilon'_1$, $-\varepsilon'_2$ generates epipolar lines intersection on the other side (right side). Corresponding point should be located between these intersections. This assumption is valid in the majority of the picture area, but there can be some specific locations where the intersections are on the same side. Therefore the use of more FFMs can increase the reliability.

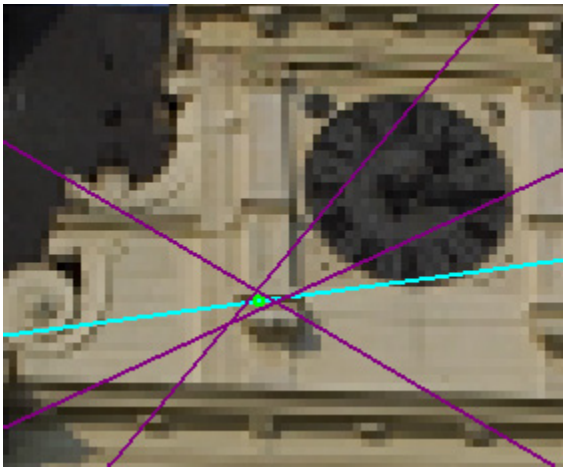

Figure 2.2: Three dark lines represent three false fundamental matrices used together. Light corresponding point is located between leftmost and rightmost intersection.

2.3 False Epipolar Constraint as Weak Constrain

Even after method's extensions, which increase the reliability of algorithm, false epipolar constraint should be considered as weak constraint. It means that it is more suitable in some cases than in another. The assumption that we can pinpoint the location of corresponding point will ultimately fail in cases when the positions of points adjacent in first image are too discontinuous in second image. (see Figure 2.3). False epipolar lines intersections on epipolar line for these points will remain continuous in second picture and will be close just to one location. As a result, all points in one of these locations will have pinpointing intersection in other (incorrect) location, thus will not be detected correctly. Which location will be pinpointed depends on the set of 9 points, form which the false fundamental matrix was computed. This problem can be possibly solved or avoided in the future. It occurs only when objects are relatively close to both cameras. This problem should not be present frequently in near-baseline stereo with cameras close each other. Increasing the searching area on epipolar line will also solve this problem because the larger area will reach both locations.

Figure 2.3: Points in the border of two objects marked in first picture have too discontinuous correspondences (marked in second picture as two circles). Intersection will be inside only one of these locations for points from both objects.

3 Points Selection

All area-based methods can detect the corresponding points thanks to the information in their local neighbourhoods. It is obvious that the detection cannot be successful, if there is not enough information in these areas to make the correct match. Therefore it is important to detect the correspondences only in those points, where the detection can be successful. These critical points should carry as much 3D information about the scene as needed. In this section, the algorithm for the critical points selection will be described.

3.1 Critical Points Description

At first, points where area-based algorithm cannot be successful has to be excluded from search. These are the points with homogeneous neighbourhood, because there is not enough information for correspondence matching. In the pictures, points like these are located usually inside a narrow planes or homogeneous areas. To recover the 3D information about these objects, points in their borders has to be selected as critical. As it is hard to define borders of the objects, in practice they are substituted by image edges.

3.2 Critical Points Selection Algorithm

Critical points have to be selected only from one of the stereo pictures. Standard corner detectors will not detect all critical points as these detectors usually cannot detect points on the borders of curved objects. As described in section above, they also should be considered as critical. Instead series of image processing methods can be used to achieve required results. The basic idea of process is to detect all borders (edges), then detect critical points on these borders.
At first, simple and fast edge detector was used.

$$Edge\_value[i,j] = Max\left(I(i+1,j+1) - I(i-1,j-1),\ I(i+1,j-1) - I(j-1,j+1)\right)$$

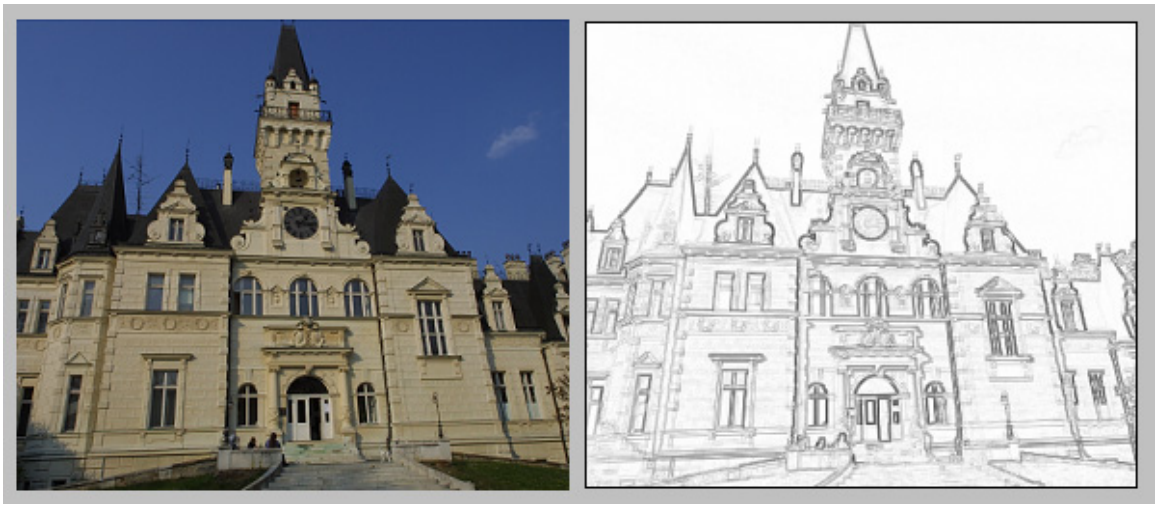

Figure 3.1: The result of edge detector.

After the edges have been detected (see Figure 3.1), threshold function was used to eliminate weak edges. At this stage, borders are displayed as thick lines. To perform next step, these lines has to be thinned to the width of one pixel. This was done by erosion operation (see Figure 3.2). Erosion operator was modified not to eliminate lines of width one.

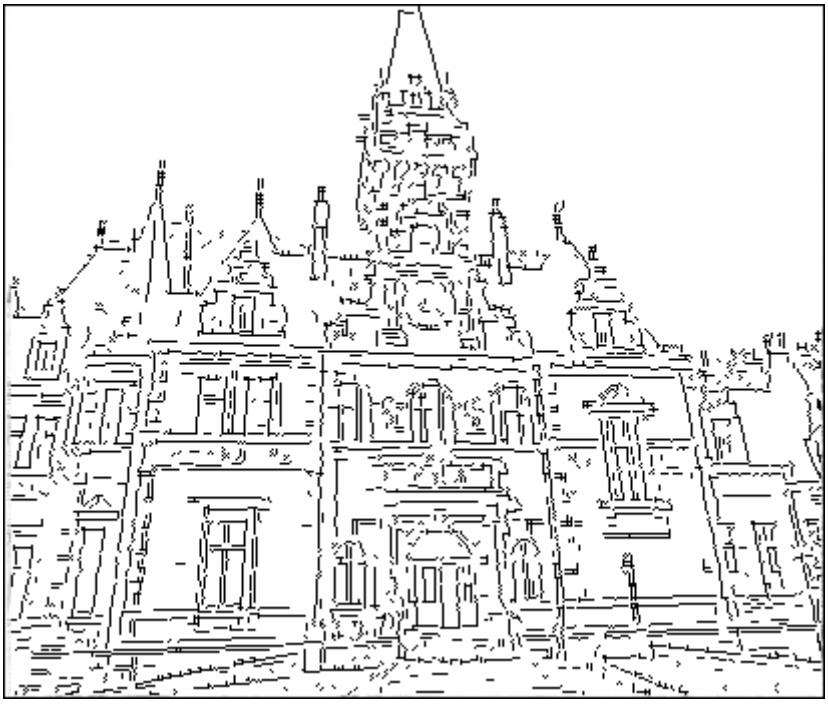

Figure 3.2: Image after the modified erosion operators.

In next step, the critical points are detected. For every edge point on the image the 1x7 mask is centered on point and this mask is rotated. For every position of the mask, the number of mask cells with edge is computed. If this value exceeds threshold value, edge point is erased.

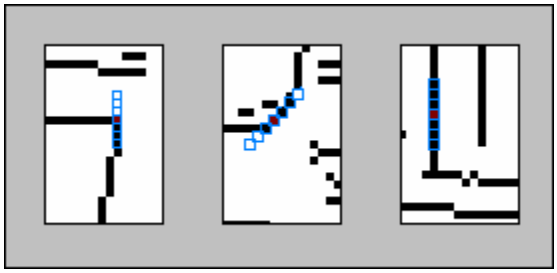

Figure 3.3: Masks for corner, curved edge and narrow edge are displayed.

In Figure 3.3 mask is displayed in various situations. If the threshold will be set to value 4, corners and points on curved edges will be detected. Points on narrow line are not critical, as the end points of this line are enough to define the whole line.

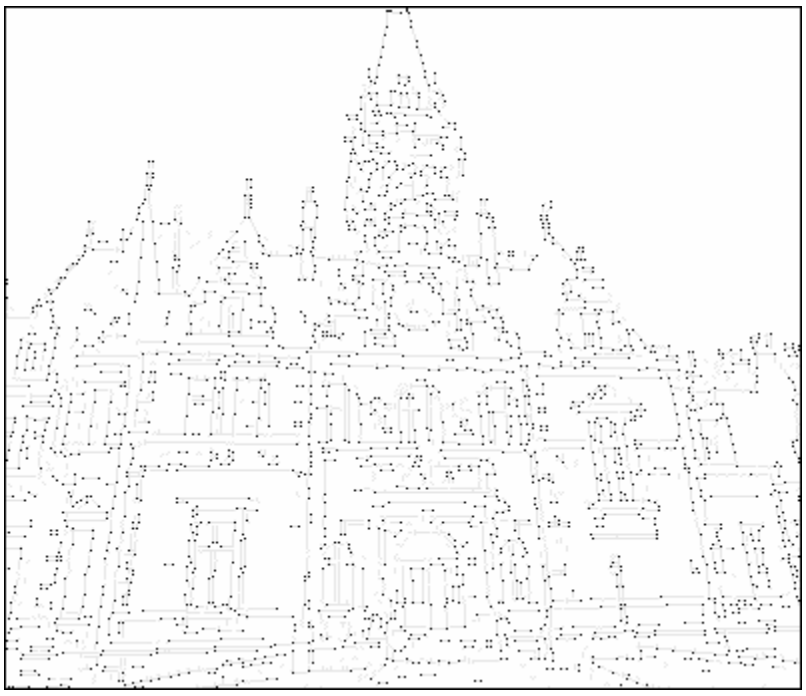

Figure 3.4: The set of critical points is detected.

In Figure 3.4, the final result is displayed. The pixels in corners and on borders of objects are displayed and pixels from narrow lines are omitted. In the case of Figure 3.4: 2182 points has been identified as critical. Mastering this technique, the number of critical points could be even lower, and should not be very resolution dependent.

4 Algorithm Implementation

To test the properties of false fundamental constraint, program has been developed. The base of the program is an area-based stereo algorithm using the intensity difference to compute the similarity of two locations (see section 1.3.1 Area Based Stereo Algorithms). Algorithm is able to use epipolar and false epipolar constraint.

4.1 Input, Output

Input for the algorithm is the pair of pictures, which represents the projections of the one scene from different positions. Both pictures has to be taken by cameras with the same intrinsic parameters, therefore the pictures has to have the same resolution. Points are selected from the first picture and their corresponding points are detected in second picture. Fundamental matrix can be imported from special file for the stereo images. Files with fundamental matrices can be created by this program as soon as the matrix is present. This file contain the matrix, but also the set of nine corresponding points from which the matrix has been computed.

The primary output of the program is the set of corresponding points detected by algorithm.

If we have at least nine corresponding points detected, algorithm is able to compute the fundamental matrix from them and export it to file.

4.2 Implementation

The form of application and computation of false epipolar constraint and epipolar constraint will be described in this section.

4.2.1 Corresponding Points Detection

When both pictures are loaded, algorithm is ready to detect corresponding points. First the point is selected manually or by function in the first picture. If fundamental matrix

is not present, algorithm must search whole picture for corresponding point. During the search, the probability map is created (see Figure 4.1).

Figure 4.1: Probability map. Value of probability is displayed in red values. Dark areas have been excluded due to different value of hue.

To speed up this process, points with different hue are excluded from the search. Point with highest probability on the map is selected as corresponding point. The number of points with similar probability as corresponding point is computed consequently. From this value, the level of success of detection is computed. Idea for this approach is that if there are many points with similar probability, there is also the greater chance, that one of them is correct corresponding point and the detection was not successful. If there is no other point with similar probability, point was possibly detected correctly. In program this level of success is visualized as the color of corresponding point in second picture. Corresponding point is marked as light green if there are no other candidates for corresponding point with similar probability. Dark green points has one other candidate, yellow points two or three candidates, orange ones - less then ten and red - more the ten other candidates. This marking is important for fundamental matrix computation, as matrix should be computed only from light green points.

When the fundamental matrix is present, algorithm can use the epipolar constraint. When point is selected in the first picture, epipolar line is computed in second picture. During the detection of corresponding point, probability map is created like before. The

probability of points which are too distant from epipolar line is set to zero (see Figure 4.2). Thanks to this, the algorithm has to search only about 0.95% of the picture for corresponding point.

Figure 4.2: Probability map is reduced to red area after the epipolar line was computed.

User can also decide if he wants to use the false epipolar constraint. This option is available only when fundamental matrix is present and is used only with epipolar constraint. Probability map is created only in area defined by false epipolar constraint (see Figure 4.3).

Figure 4.3: False epipolar constraint reduce the probability map to smaller area marked as red on the picture.

4.2.2 False Fundamental Matrix Computation

User is able to perform several types of false fundamental matrix computation, to test different kinds of approaches. At first, fundamental matrix has to be present (imported or computed). After selecting False Fundamental Matrix (FFM) -> Find Matrix, dialog window for computation of FFM is displayed. Nine points has to be selected for computation, or nine points from fundamental matrix can be used the same way. Consequently, the type of modifications to corresponding points can be selected (see Figure 4.4):

- Random Modification – modifications are randomly chosen from interval
- Function Modification – function described in section 2.1 is used
- Manual modification – every modification can be performed manually

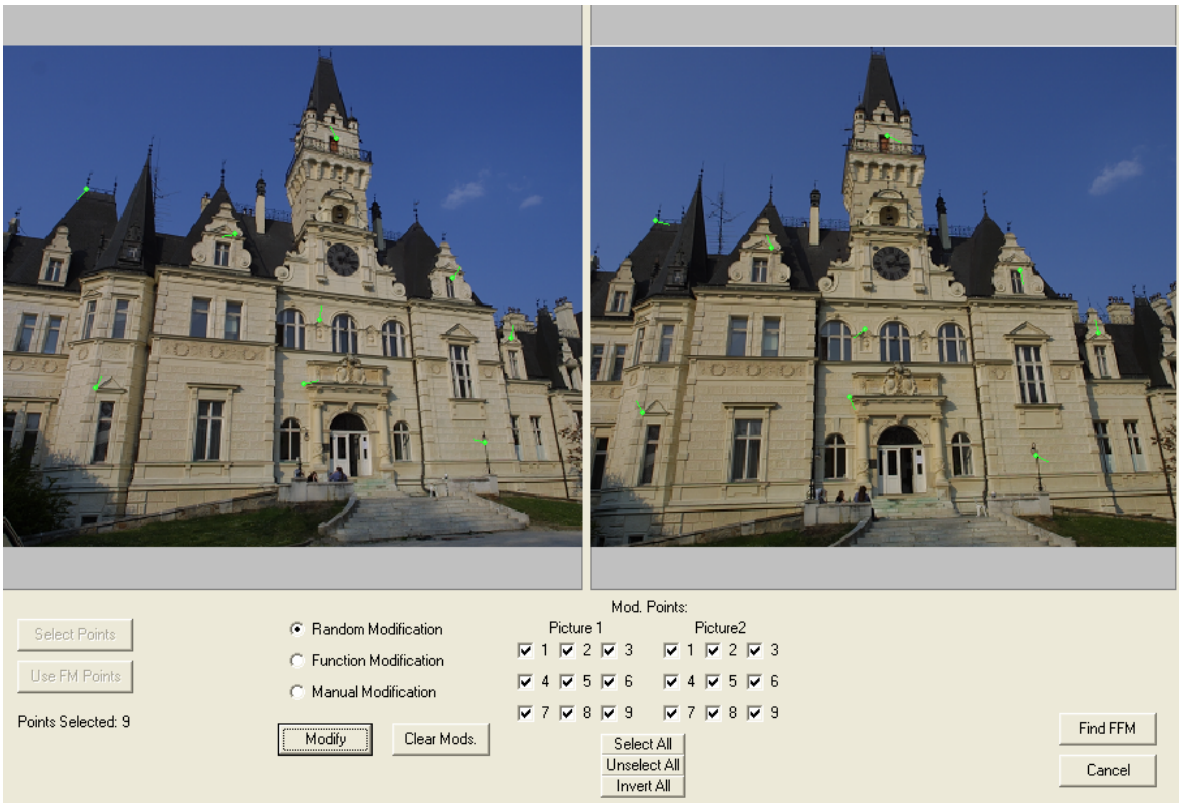

Figure 4.4: Directions of corresponding points modifications are displayed as green lines centered on points. The real magnitude of modifications is ten times smaller then are the lengths of the green lines.

User can also restrict modifications to only some chosen points. After modifications have been performed, matrix is computed. Several matrices can be used at once. Also the import/export functionality for FFM is implemented.

5 Test Results

Several experiments focused on false epipolar constraint have been conducted with program described in previous section. The most important experiment performed was to determine the overall speed up of the corresponding points detection. Assumption for the experiment was that input is from near-based stereo so the provable area highlighted on epipolar line should not be large. Several input stereo pictures with different resolutions were provided and on each set of stereo pictures 500 corresponding points was detected. Time needed for this detection was measured. Results are displayed in Table 2.

Resolution	**EC [s]**	**EC+FEC [s]**
400x300	15	10
800x600	31	11
1200x900	59	13
1600x1200	97	16

Table 2: Time needed for detecting 500 corresponding points is recorded. EC - only epipolar constraint was used; EC+FEC - results of using additional false epipolar constraint is displayed, both for different picture resolutions.
(tested on Athlon 2000+, 512 MB RAM)

As we can see in Table 2, speed increases significantly when false epipolar constraint is used for high resolution pictures. The computational acceleration is based on reduction of search area. About 0.95% of the second picture area has to be searched for corresponding point when only epipolar constraint is used. With false epipolar constraint, it is only about 0.08% of picture. This reduction is so big, that the algorithm speed dependence on the picture resolution becomes insignificant.

The speed of algorithm is slightly dependent on the number of false fundamental matrices used. The more matrices are used, the larger is the searching interval. This dependency is displayed in Table 3.

Number of matrices	5	6	7	8	9
Searching area size (% of picture)	0.071	0.076	0.079	0.080	0.081

Table 3: Dependency of average searching area and number of false fundamental matrices.

It is also obvious that the reliability of successful point detection is higher thanks to the false epipolar constraint. However this property is hard to test, as it is varying case to case. For example when there are many similar objects on the picture, reliability can increase considerably. (see Figure 4.5).

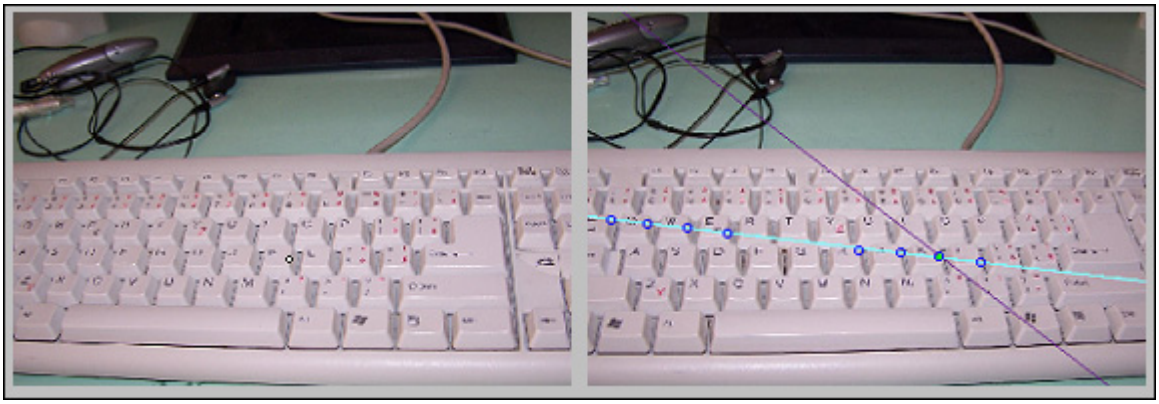

Figure 4.5: A light epipolar line is running through many similar locations in this picture (marked in circles). Usually this can cause incorrect detection, but false epipolar constraint was used here to pinpoint correct location (dark line).

When we want to estimate the optimal size of search area, it could be useful to know the average distance of corresponding point from lines intersection. This value is also dependent on many features like modifications and selection of corresponding points for matrix computation, calibrations of cameras and parameters of stereo system. As it is impossible to cover all this cases, this test was performed only for one model situation (see Figure 4.6).

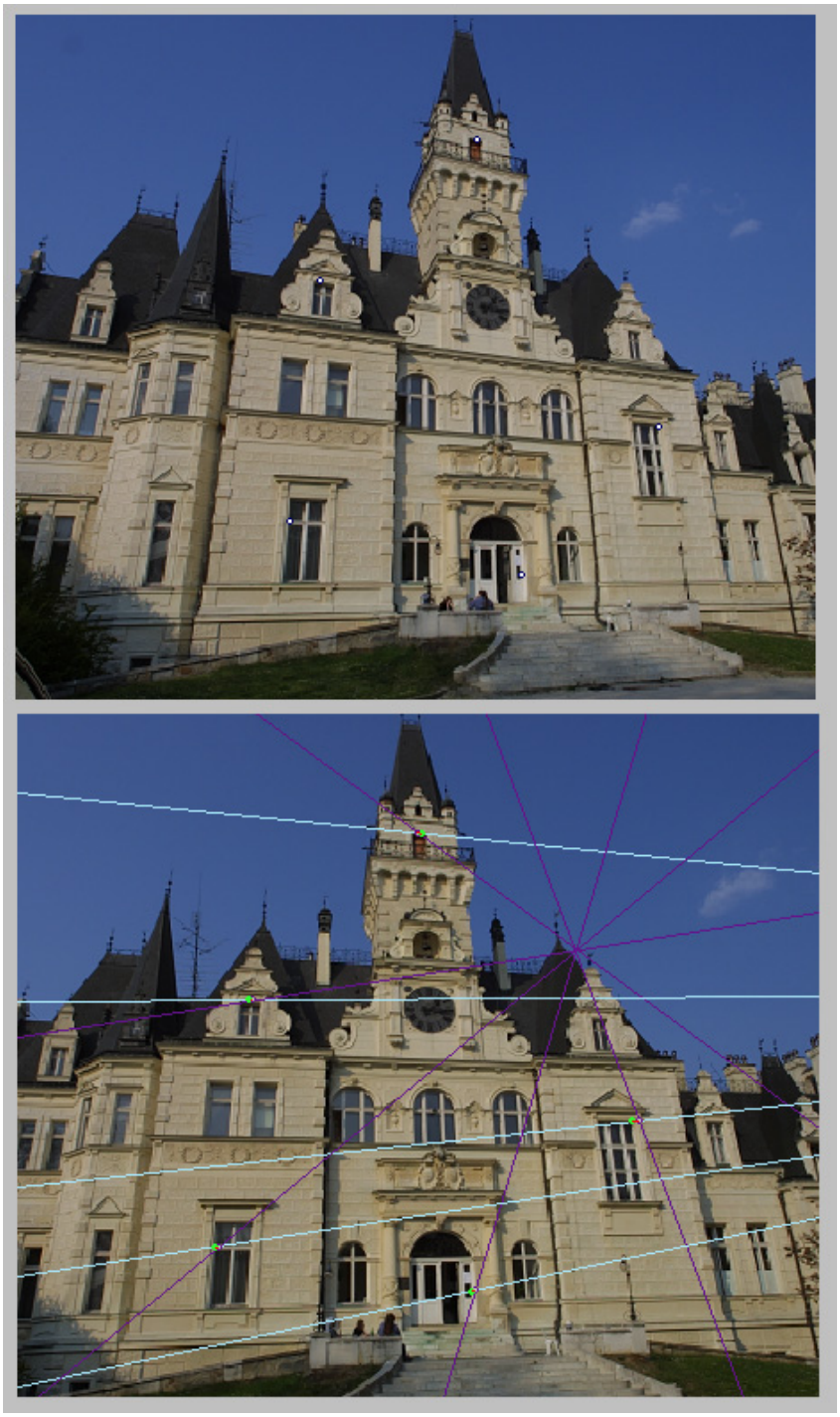

Figure 4.6: Five points from different areas of picture has been detected. Distance of intersections from these points is less then 3 pixel for all cases. Light lines are the epipolar and dark lines are false epipolar.

Two hundred equally distributed corresponding points has been detected in stereo pictures (resolution 800x600) and the distance of each point from lines intersection has been measured. Results are displayed in Table 4:

Distance in pixels	0-3	3-10	>10
Number of points	176	14	10

Table 4: Distance of 200 corresponding points from intersection.

To conclude the tests, speed of corresponding points detection can be increased up to 6 times, mainly in pictures with high resolution. The reliability of detection is increased in pictures with many similar objects.

6 Conclusion

When the false epipolar constraint is used properly, it can significantly speed up the corresponding point detection algorithm. This method allows the area-based stereo algorithm to run several times faster then before and with increased reliability. Simple implementation and low software and hardware requirements allow this constraint to be used even in most basic stereo systems. I believe that also this constraint can be one of the methods that will help the stereo algorithms to become the most universal and widely used distance measuring method.

The first program, in which this method was used is described in section 4 (Algorithm Implementation). The method extensions were consequently derived during the testing. With extended usage of this method another modification could be implemented. One of them could be the solution of the weak constraint problem described in section 2.3. As mentioned before, the location of intersection is dependent on the set of 9 points. This problem can be avoided by computing multiple false fundamental matrices, each from a different set of points. This modification can possibly solve most cases, but as a result, the computing time will increase, because the searching area in epipolar line will be more extensive.

Focus of this paper is on area-based algorithms. It should be noted, that false epipolar constraint can be used in any algorithm, which uses also epipolar constrain. For example, the feature-based algorithms use the epipolar constraint to match corresponding points or to discover incorrect correspondences. False epipolar constraint can improve both speed and accuracy of these tasks.

7 References

[1] A. Assadi. *Perceptual Simplification of Geometry of Natural Surfaces in Human Vision.* UW Mathematics

[2] K. Dařílková. *Modeling of Real 3D Object using Photographs.* WSCG‘2005, Plzen 2005; ISBN: 80-903100-9-5

[3] A. Glassner. *Principles of Digital Image Synthesis*. 1994 ISBN:1558602763

[4] R. Hartley, A. Zisserman. *Multiple View Geometry in Computer Vision.* Cambridge University Press, March 2004 ; ISBN: 0-521-54051-8

[5] O. Faugeras, B. Hotz, H Mathieu. *Real time correlation - based stereo: algorithm, implementations and applications*. AOEUT 1993.

[6] O. Faugeras, Q.T. Luong. *The Geometry of Multiple Images.* MIT Press, Cambridge, Mass. US. 2001; ISBN: 0-262-06220-8

[7] A. Fusiello, V. Roberto, E. Trucco. *Experiments with a new Area-Based Stereo Algorithm*. International Conference on Image Anlysis and Proceedings, Florence 1997

[8] H. Longuet-Higgins. *A Computer Algorithm for Reconstruction a Scene from Two Projections*. Nature, 1981 ISBN:293:133-135

[9] A. Milela, F. Pont, R. Siegwart. *Model-Based Realtive Localization for Cooperative Robots Using Stereo Vision.* EPFL Switzerland.

[10] L.D. Stefano, M. Marchionni. *A Fast Area-Based Stereo Matching Algorithm*. DEIS-ARCES, University of Bologna.

[11] E. Vincent, R. Laganiere. *Matching Feature Points in Stereo Pairs*. Univarsity of Ottava

[12] W. Wang, H.T. Tsui. *A SVD Decomposition of Essential Matrix with Eight Solutions for the Relative Positions of two Perspective Cameras*. ICPR 2000

Abstrakt

Stereo algoritmy založené na oblastiach sú považované za najuniverzálnejšiu metódu stereovízie a majú potenciál pre široké využite v praxi. Avšak ich relatívne nízka rýchlosť bráni praktickému využitiu. V tejto práci je opísaná metóda, ktorá umožňuje podstatne zvýšiť ich rýchlosť. Táto nová metóda sa volá nepravá epipolárna podmienka. Umožňuje redukovať množinu možných korešpondenčných bodov pre algoritmy rovnakým spôsobom ako epipolárna podmienka (epipolar constraint) [5]. Tieto dve metódy sa dajú použiť spolu pre zvýšenie efektivity stereo algoritmov a to vo všetkých prípadoch, kde sa dá použiť epipolárna podmienka.

Keywords: epipolar, stereo, computer vision