De Montfort University
Leicester

FEATURE-BASED
IMAGE REGISTRATION
IN REAL TIME APPLICATIONS

September 2007

PhD Thesis

Alexander Bobkov

2007
# Contents

## CHAPTER 1. INTRODUCTION AND LITERATURE REVIEW

1.1. Introduction .............................................................................................................. 10
   1.1.1. Specifics of aerial image registration ................................................................. 11
   1.1.2. A brief review of image analysis tasks ................................................................. 12
   1.1.3. Aims and objectives .......................................................................................... 15
   1.1.4. Thesis outline ................................................................................................... 16

1.2. Image registration problem .................................................................................... 17
   1.2.1. Image registration: statement of the problem ..................................................... 18
   1.2.2. Classification of image registration methods ..................................................... 19
      1.2.2.1. Space transformation ..................................................................................... 19
         1.2.2.1.1 Global transformation ................................................................................. 19
         1.2.2.1.2 Local transformation .................................................................................. 20
         1.2.2.1.3 Optical flow .............................................................................................. 20
      1.2.2.2. Image representation ..................................................................................... 21
         1.2.2.2.1 Image representation methods – an overview .......................................... 22
         1.2.2.2.2 Boundary representation ............................................................................ 23
         1.2.2.2.3 Region-based representation ..................................................................... 25
         1.2.2.2.4 Summary on image representation ............................................................. 27
      1.2.2.3. Optimisation strategy .................................................................................... 28
         1.2.2.3.1 Optimality criterion .................................................................................. 28
         1.2.2.3.2 Optimisation algorithm ............................................................................. 29
         1.2.2.3.3 Exhaustive search ..................................................................................... 30
         1.2.2.3.4 Graph-oriented search .............................................................................. 30
         1.2.2.3.5 Hough Transform strategy ........................................................................ 31

1.3. Line and curve extraction ....................................................................................... 32
   1.3.1. Parameterisation scheme selection ...................................................................... 33
   1.3.2. Reducing the parameter space .......................................................................... 34
   1.3.3. Parameter space decomposition ........................................................................ 35
   1.3.4. Using additional information ............................................................................. 35
   1.3.5. Using different analysing orders ........................................................................ 35
   1.3.6. Line splitting problem ....................................................................................... 36
   1.3.7. Line segment detection ...................................................................................... 37

1.4. Contour detection ..................................................................................................... 38
   1.4.1. Edge detector .................................................................................................... 39
   1.4.2. Edge labelling .................................................................................................... 42
      1.4.2.1. Edge thinning .............................................................................................. 42
         1.4.2.1.1 Skeletonization ......................................................................................... 43
         1.4.2.1.2 Non-maximum suppression ...................................................................... 43
   1.4.2.2. Thresholding .................................................................................................. 43

1.5. Summary .................................................................................................................. 44

## CHAPTER 2. IMAGE CONTOUR EXTRACTION

2
2.1. Differential method ................................................................. 47
2.2. Statistical methods ................................................................ 48
  2.2.1. How to select the area size for a statistics collecting .......... 49
  2.2.2. How to determine a threshold and amount of modes ......... 49
  2.2.3. Statistical method characteristics................................. 50
2.3. Contour thinning .................................................................... 50
  2.3.1. Thresholding ................................................................... 52
  2.3.2. Global and local thresholding comparison ....................... 54
  2.3.3. Non-maximum suppression ............................................. 57
  2.3.4. Use the thresholding and edge thinning at the same time .. 59
2.4. Comparison of statistical and differential methods ............... 59
2.5. Conclusion ............................................................................. 65

CHAPTER 3. LINE AND CORNER DETECTION .............................. 67
  3.1. Contour representation ......................................................... 67
  3.2. Line detection ....................................................................... 68
  3.3. Arc and curve detection ....................................................... 69
  3.4. Corner detection .................................................................. 70
  3.5. Line segment detection ....................................................... 72
  3.6. Proposed method of line segment detection ......................... 74
    3.6.1. Extended Hough transformation ................................. 74
    3.6.2. Algorithm based on Extended Hough Transform .......... 76
    3.6.3. Performance of proposed algorithm ......................... 78
    3.6.4. Gluing of line segments ............................................. 78
  3.7. Experiments ......................................................................... 80
    3.7.1. Performance analysis ................................................... 81
    3.7.2. A metrics problem ....................................................... 83
    3.7.3. An optimal discretization step selection ....................... 85
    3.7.4. Minimal object size that can be approximated by line segments 88
  3.8. Conclusion .......................................................................... 88

CHAPTER 4. IMAGE REGISTRATION ............................................... 90
  4.1. Introduction .......................................................................... 90
  4.2. Statement of the problem .................................................... 90
  4.3. Image registration strategies ............................................... 91
    4.3.1. Exhaustive search ....................................................... 91
    4.3.2. Graph matching ........................................................... 91
    4.3.3. Hough Transform strategy ......................................... 92
  4.4. Hough Transform strategy variations ...................................... 93
    4.4.1. Point features formed from non-point ones .................. 93
    4.4.2. The search space decomposition ................................. 94
    4.4.3. Standard Hough Transformation approach ................. 94
  4.5. Exhaustive search ............................................................... 95
  4.6. Methods of endless line comparison ...................................... 95
    4.6.1. Line matching based on Hough transform .................... 95
Abstract

This work is devoted to the task of image registration in real time. The image registration task is task of finding position of the one image on another. Images can be obtained at different time, from different sources and in different conditions.

The task of image registration is fundamental task and it is important for many practical applications, such like visual navigation and surface monitoring by satellites and unmanned flying vehicles, road traffic analysis, different applications of computer vision in robotics and many others. In this work we pay special attention to the task of visual navigation. However, results obtained can be used in the different areas.

In the terms of visual navigation, the task can be described as follows. Suppose that we have a basic image (map) that is obtained before and known a priory, and the current image (frame), obtained from onboard video camera in real time. Our task is to find position of the current image on the basic image, or that is the same, to find the area on the basic image that looks like the current image. This task is essential, for example, for unmanned flying vehicle navigation by given path or for inspecting given region, when other navigation methods (like inertial and global positioning system) cannot produce enough accuracy.

The aim of this work is to develop a common approach to the task, investigate possible solutions, investigate its advantages and disadvantages, implement most promising algorithms, research its characteristics in different conditions, compare them with existing ones, and, finally, make conclusion about possibility to use them in practice.

The basic idea of this work is to use image representation as a set of line segments. Line segments are universal feature that are usually present on most of natural images in enough amounts, and they can be easily extracted and matched. This makes the approach very promising and popular for many researchers. However, the results already achieved are far from optimal. The approach has both advantages and disadvantages, obvious and hidden. Our objective is to determine the primary characteristics of this approach, discover the factors that affect them, research their behaviour in different conditions and for different data, compare them with the characteristics of other methods used in practice and understand the practical use of the approach.

The primary strategy of the work is using of the Hough transform (HT). HT is a common method that allows finding typical objects on the image. The specific requirement of the task is high performance. However, you can not get extra performance for free – the price is extra equipment or loss in accuracy and reliability. HT allows increasing in performance in exchange to extra memory for storing temporary results. This looks promising because modern systems have a lot of relatively low cost memory.

We will use the HT approach twice, once for line segment extraction and then for matching them. Existing schemes of HT for line segment detection have good reliability, but provides insufficient accuracy. The work proposes extended
scheme of HT that allow increasing the accuracy of line segment detection up to one pixel without loss of performance. The accuracy becomes an important factor in enabling high reliability in further line matching.

We then came back to the HT in a line segment matching task. Different matching schemes were analysed, and best of them tested and compared with existing methods.

The direct testing of proposed methods in a flying laboratory is very expensive and hard (imagine the software debugging process when every mistake can cause loss of equipment), so another approach is used. We reproduce the task conditions in the robotic laboratory. We use a manipulator, which carries the camera over the surface model, emulating the flying vehicle behaviour in the proper scale. This allows implementation of image analysis in the conditions that are near to natural.

The results of experiments show that proposed methods can be implemented in real time on existing equipment; they provide high accuracy and enough reliability for use in practice. The methods are insensitive to the brightness and light conditions that make them preferable in comparison with existing approaches.

The results of the work expand the knowledge about HT, open new areas where it can be successfully implemented, open new ways to accurately detect of geometric objects and show the possibility of using Hough transforms to locate random shapes described by line segments.

The results of the work can be used in different practical applications that require accurate automatic visual navigation. Other possible application areas are image registration from different data sources, object detection in traffic analysis tasks, object localisation in robot visual field and surely many others.
Acknowledgments

I would like to use this occasion to thank many people, who helped me working over this book.

I would like to thank my supervisor Jeffrey Knight for the guidance and time spent on me,
Konstantin Pupkov for the equipment and conditions for working,
Evgeniy Lobusov for the priceless advices,
MT and Inex for deep discussion of my ideas,
Charlotte Kitson and Veena Vorra for help and support,
and of course my wife Olenka who stimulated me to complete this work.
## Abbreviations

<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ARPS</td>
<td>Automated Robot Programming System</td>
</tr>
<tr>
<td>DG</td>
<td>Differential Gradient</td>
</tr>
<tr>
<td>DGHT</td>
<td>Dynamic Generalised Hough Transformation</td>
</tr>
<tr>
<td>EHT</td>
<td>Extended Hough Transformation</td>
</tr>
<tr>
<td>GHT</td>
<td>Generalised Hough Transformation</td>
</tr>
<tr>
<td>HT</td>
<td>Hough Transformation</td>
</tr>
<tr>
<td>IR</td>
<td>Image Registration</td>
</tr>
<tr>
<td>LSC</td>
<td>Line Segment Comparison</td>
</tr>
<tr>
<td>LSPC</td>
<td>Line Segment Pair Comparison</td>
</tr>
<tr>
<td>MAT</td>
<td>Median Axis Transformation</td>
</tr>
<tr>
<td>NMS</td>
<td>Non-Maximum Suppression</td>
</tr>
<tr>
<td>NP</td>
<td>Non-Polinomial</td>
</tr>
<tr>
<td>PHT</td>
<td>Probabilistic Hough Transformation</td>
</tr>
<tr>
<td>RHT</td>
<td>Random Hough Transformation</td>
</tr>
<tr>
<td>SHT</td>
<td>Standard Hough Transformation</td>
</tr>
<tr>
<td>TM</td>
<td>Template Matching</td>
</tr>
</tbody>
</table>
CHAPTER 1. INTRODUCTION AND LITERATURE REVIEW

1.1. Introduction

The use of video image for automatic control of dynamic object is a very important task, which has attracted a significant amount of attention over the past years. On the one side this relates with the practical needs, and on the other side – with the fast development of technology area, through the development of computers with very high performance.

The task of image registration can be described as follows. Suppose that we have a basic image that is obtained before and known a priory, and the current image, obtained from video camera in real time. Our task is to find position of the current image on the basic image, or that is the same, to find the area on the basic image that looks like the current image.

The task of image registration is fundamental task and it is important for many practical applications, such like visual navigation and surface monitoring by satellites and unmanned flying vehicles, road traffic analysis, different applications of computer vision in robotics and many others. In this work we will take especial attention to the task of visual navigation in first of all. However, obtained results can be used in the different areas.

Since the task must be solved in real time, the implemented algorithms must provide enough performance to process incoming image frame before then the next one will arrive. Usually it is about 25 frames per second; however, several applications require higher frequency (100 frames per second or may be even more). Another requirement related with system stability: computation delays must provide enough stability for the control loop. And finally, there are application-specific requirements. For example, if we looking for moving object, we must process the frame before the object leaves the visual field. This gives a typical computation time no greater than 50 milliseconds in the conditions, when the available hardware performance are significantly limited.

A video image contains a huge amount of information, which can be used for analysis and control – much more than can be extracted from the other sensory systems. Image systems are today relatively simple, and can be exact, reliable, and flexible. Processing by way of different physical phenomena is also possible – image in infrared and ultraviolet diapasons, temperature and altitude field, microscopic and telescopic image. Video systems are capable of obtaining information in a wide range of conditions and in different applications.

The main disadvantage of video system is related with its main advantage – the huge stream of information requires significant computation resources to be processed in real time. A video sequence is a set of frames obtained with a small delay (usually it is 0,04 sec). The frame consists of a set of pixels – digitised values
of brightness at each point. So a camera with a tiny frame of 200x200 pixels and standard speed of 25 frames per second will generate 200x200x25 = 1 Mb data each second. This is too much for the most of real time applications using a reasonable amount of computer power.

Imagine another situation. Suppose we have tiny map of 2048x2048 size, and trying to find the position of medium size frame 1024x1024 on it by direct comparison. Possible amount of positions is 1024x1024, and we need at least 1024x1024 of operations to check each of them. So we got 2^40, or 1.2x10^12 operations – one quadrillion. It is uneasy task even for most powerful computers. So the effective computation algorithm is quite important.

1.1.1. Specifics of aerial image registration

This research is devoted to the task of image registration. The primary area of application is the task of searching the position of aerial or space image on the digital map in real time. This problem often arises in navigation (observe-matching navigation) and in a different surface monitoring tasks. The particular area of application puts specific conditions and limitations to the statement of the problem.

- Significant map size. This causes both significant computations for acceptable image registration, and significant memory to hold the map and the temporary results of the computations. Since the image registration system is often an on-board device, it places a serious limitation in terms of computing power and the amount of available memory.

- Sensor parameters are often known or can be estimated in off-line mode. The geometric transformation to convert an observed mage into a map is a rigid transformation without local distortions. These factors can be used to reduce both size and dimensionality of search space in order to speed up calculations and to bound the amount of memory required.

- Some of the important parameters like rotation and scale can be estimated from other sources of information. For example, rotation can be found using orientation information by inertial or magnetic compass; scale can be estimated from altimeter data; projective transformation parameters can be fixed by mounting the camera on the gyro-vertical platform. This also limits the dimensionality of required transformation.

- The brightness transformation always plays important role due to different illumination conditions, which depends on the sun positions, time of year, weather and so on.

And finally, the most serious requirement is the working in a real time.

The primary requirements for any real-time systems are performance, accuracy and reliability.

1. Performance. The system must analyse an image in a fixed time in order to reserve time to take a decision and produce control signals. If the time limit is exceeded, the system cannot work properly (for example, the object we trace can
left the visual field, or object we must find appears on the skipped frames), or system stability may be dangerously reduced in the case when information is used inside the control loop.

2. Accuracy. Information extracted from the image must meet a pre-specified accuracy. A reduction in accuracy can also make a system fail since it can be a problematic to tune manually the system parameters in a real-time.

3. Reliability. A system is reliable when it can keep working to the specified accuracy when external influence of different nature is present. This influence can be noise, changing of visual conditions and perspective, parameter fluctuations with respect to time and many others. A system that operates in real time must react adequately to such influences.

It is easy to see that these three requirements are mutually exclusive: faster methods can be less accurate and less reliable; more reliable and robust methods may have a lower accuracy and performance. The task is to provide such characteristics that are acceptable for the concrete practical application in question.

Indeed the relation between performance, accuracy and reliability is more complex. For example, an increase in performance can provide analysis of more frames in a given time. This increases both accuracy and reliability of the obtained information, and can facilitate faster methods. In such a way, performance can be considered as a key factor that is affected in a complex way to the accuracy and reliability in a context of the challenge of a practical task and the available methods of image analysis.

1.1.2. A brief review of image analysis tasks

Dynamic control using video images can be found in a wide range of different areas in mechanics and industry. Thus there are a wide range of tasks where an A number of classification schemes describing image extraction exist, and commonly used scheme is shown on fig.1.1.

1.1.2.1. Recognition and identification of objects

In relation to partial task requirements, this task can be also divided into several groups.

a) Looking for an object on a static image

This is a classical image recognition task, which requires identifying the object on the image as one of existing samples, usually using a function of maximum matching. Real tasks are more complicated due to a presence of other objects and background with complex structure, overlapping by other objects, absence of priority information about object location, orientation, scale, illumination etc. The task can be rewritten as a task of object position determination, the object separation from background and so on.
Fig. 1.1. Classification of image registration tasks
Object searching can be performed for the following identification (e.g. automatic recognition of fingerprints or iris), for measurement of object characteristics (size, colour, texture, orientation etc.). Objects found can be also used to glue overlapped images (e.g. obtaining a whole map by a series of photos), data merging from different sources (e.g. merging visible and infra red images in order to enhance the image quality) and others.

Tasks of this group can be reduced to task of searching one or several sample objects on the image. The samples can be images, or sets of image structural elements, or even object parameter sets.

b) Object tracing

In this task group the object to be found and recognised is moving, and its motion differs from the whole image motion. There are tasks of object position monitoring (object approaching, collision avoidance, following an object). If the task is solved for each frame pair, it transforms to a previous task. But in several cases there are a set of limitation that can prevent to do this: for example, whole object or its significant part can move outside the visual field, or time limitation prevents each processing of frames at a fast enough refresh rate. Other approaches must be used in this case.

c) Trajectory reconstruction

In some cases it is impossible to identify the object perhaps due to low image quality or small object size. Nevertheless, the object can still be recognised by the parameters of motion and trajectory. In the simplest case there is only one moving object on the image and it can be easy detected. In more complex cases all moving objects must be monitored, and required object may be found by its motion dynamics.

1.1.2.2. Tasks of self-motion determination

In this group of task the motion of the camera (and the object that holds it) must be determined in relation to a visual surface – the scene. In regard to a motion type, two different cases can be described. If the object moves mainly in a parallel direction to a scene, and changes of distance to the scene can be neglected, this is known as a flat task. In this task, all parts of image have the same behaviour. This task often appears in navigation of flying and space object, which moves on fixed or slowly changing altitude, i.e. parallel to a earth surface. The task can arise in industrial applications, when the tool with the camera moves over a working surface – for example, in tasks of microscopic scanning and monitoring.

In the case when the motion is directed to a scene, then this is a three-dimensional task. This task arises, for example, in mobile robot control applications, in tasks of automatic targeting and docking. The difficulty of this task is that the motion parameters of object depend on the distance to them. An additional task is necessity to measure the distances to the scene elements. An additional difficulty is the possible non-linear characteristics of the camera, which when present is difficult to correct in real time.

The first task deals mainly with image translations, rare with rotation and scaling. Often the rotation angle and scale is constant or can be found from other
sources. Second task deals with scaling first of all, since it allows estimation of the distance to objects. Other transformations – translation and rotation – are less important; often the rotation angle is known or constant, and translation is limited by one direction, e.g. in the autonomous mobile navigation task.

However, in common case more complex transformations must be taken into account. The typical reasons are that the visual scene can be not flat or observed from the small angle, or optical system often adds non-linear distortions. The overview of the different transformation affection can be found in [Brown 1992].

1.1.2.3. Video scene analysis

In the common case, it can be required to solve both of the above mentioned tasks together. For example, in a mobile robot navigation task it is required not only to monitor and correct the desired trajectory, but also avoid collisions with other objects, that stand or move, change direction in order to pass the obstacle, give a way to another moving object and so on. In many cases it is also required to determine the relationships between detected objects, determine the character of it interaction, and predict the future states of the scene. This is a most complex task, and universal methods to solve them are unknown.

In such a way, most image analysis tasks consider looking for a relation between two or more images. This can be either looking for an object position on an image (localisation task), or looking for a feature that is nearest to the selected image fragment (identification task). These two tasks are tied: to recognise an object, it must be firstly found, and vice versa, to find the object it is required to know what exactly must be found, i.e. it is required to determine a parameter set that belongs to desired object. The correspondence between localisation and identification task allows applying solutions, found for one task, to another task. In the simplest cases, the task can be reduced either to localisation or recognition. For example, if the identification procedure is applied to every image area, the object position can be found. Or, looking for all possible samples on the image, it can be found which of them are present on the image. The task of sample localisation is a subject of the image registration theory, whereas identification task is subject of the image recognition theory.

This research deals with sample localisation task only. However, solutions that can be found can be applied to the image recognition task as well, since both tasks are tightly related and have its inner likelihood.

1.1.3. Aims and objectives

Our primary task in this work is developing and researching methods of image registration that can be used in real environment and in a real time. The primary domain of application is the visual navigation task; however, other applications are also can be taken into account.
The basic idea of this work is to use image representation as a set of line segments. Line segments are universal feature that are usually present on most of natural images in enough amounts, and they can be easily extracted and matched. This makes the approach very promising and popular for many researchers. However, the results already achieved are far from optimal. The approach has both advantages and disadvantages, obvious and hidden. Our objective is to determine the primary characteristics of this approach, discover the factors affected on them, research their behaviour in a different conditions and for different data, compare them with the characteristics of other methods used in practice and understand the practical use of the approach.

We have selected the Generalized Hough Transform as the key strategy of line segments matching. Our next objective is analysing the possible methods of line matching, investigate its advantages and disadvantages, measure its characteristics, and make conclusion about practical usage.

In order to represent the image as a set of line segments, we also used the Hough Transform. Existing schemes of HT for line segment detection have good reliability, but provides insufficient accuracy. So the next objective is producing the extended scheme of HT that would allow increasing the accuracy of line segment detection up to one pixel without loss of performance. The accuracy is quite important factor because it allows reaching high reliability in further line matching. We also need to research properties and characteristics of obtained line segment detector.

To implement the line segment detection, we need to extract the contour of the image – the line segments can be found only on the object edges. Contour detection task is not so interesting because it is deeply researched; however obtaining the contour detector of proper characteristics is still difficult task. Our objective would be review existing approaches to the contour detection, select the proper components and combine them. Then we need to measure characteristics of the contour detector in order to be sure that it provides suitable characteristics for line detection.

The final objective is testing of proposed methods on statistically representative data sets in order to ensure that the characteristics are suitable for solving practical tasks. The direct testing on the flying vehicle is very expensive and hard (imagine the software debugging process when every mistake can cause loss of equipment), so another approach is used. We reproduced the task conditions in the robotic laboratory. We used a manipulator, which carried the camera over the surface model, emulating the flying vehicle behaviour in the proper scale. This allows implementing image analysis in the conditions that are near to natural. Appropriate software would be also developed.

1.1.4. Thesis outline

The thesis is organized as follows.

The rest of the chapter is analysis of image registration task and review of the literature on the image registration and the related topics – shape
representation, line detection and edge extraction. The objectives is to review existing image analysis research, to examine existing solution methodology, and to determine a common approach to developing a practical image analysis system that can process a wide range of images in real time.

Second part is devoted to the contour extraction. Here we select the structure of the contour extraction algorithm and select proper components for its implementation. We compared direction gradient and statistical methods to choose the best edge detector, suitable for further line segment extraction and matching. Then the proper thresholding algorithm are chosen and tested. Finally, the edge thinning method will be added, and whole contour detection algorithm will be tested and tuned in order to produce good contour for further line detection.

Third chapter is devoted to the line segment detector development. We used well known Hough transform approach. Its disadvantages were analysed and more accurate and fast computation scheme were proposed. Then the properties of the proposed algorithm were tested in order to select proper parameters.

Fourth chapter is devoted to the line matching. The main approach used for line matching in this work is Generalized Hough transform. We then came back again to the Hough transform and describe possible approaches of line matching. Then we select the best of them and test them thoroughly in order to determine the performance, accuracy and reliability in the different conditions. The measurement uses statistics collected on large data sets.

Fifth chapter is the discussion of image registration methods proposed in the previous chapters. We analysed the performance, accuracy and reliability of the methods in different conditions and environment, showed the negative factors that can be affected to characteristics, and compared the method behaviour with the other popular methods commonly used. It is shown that proposed methods are suitable for the task of image registration for visual navigation purposes, have high characteristics, and can be applied in many other areas.

In the appendix reader can find the description of our equipment and software used to provide experiment materials for the work. This is robotic equipment for laboratory emulation of conditions of the visual navigation of unmanned flying vehicle with a camera. The software we developed are used to process the obtained data, to test the contour extraction, line detection and matching algorithms, to collect and process statistics on algorithm behaviour in different environment and different parameters.

1.2. Image registration problem

The Image registration problem is one of the fundamental and important problems in computer vision. At the current time, a problem of image matching is solved only for several particular applications and still remains an actual problem. On the one hand, researchers try to build reliable and totally autonomous systems
that can operate without human aid. Such systems have a wide practical interest, but they are developed only for a very narrow class of tasks. On the other hand, researchers try to build universal task-independent systems to check hypotheses of image interpretation area and to model biological systems of image acceptance.

1.2.1. Image registration: statement of the problem

There are two images $I_1$ and $I_2$. The registration task is to find a geometric transformation $g(I)$ and brightness transformation $f(I)$, which transform image $I_i$ in such a way that corresponding pixels of both image match to each other:

$$I_i(X) = f(I_2(g(X)))$$

(1.1)

The co-ordinate system can be different due to a different position of a video camera, camera rotation or motion of an observed object itself. So the primary task is to transform both images into a common co-ordinate system. The brightness transformation can be also taken into account due to a possible difference in light conditions, different season or time of the day. Beside this, images in the common case can be obtained by different kinds of sensors, and also require both space and brightness transformation.

Since the equation (1.1) is inaccurate in practice, the quality functional $C_1$ is introduced as a matching quality criterion for selected transformations of space and brightness:

$$C_1(g, f) = \int_X [I_i(X) - f(I_2(g(X)))]dX$$

(1.2)

The following definition will be given.

**Control points**, or **reference points** are points, which positions are known on both images.

Since the control points must be transformed exactly to their pairs on another image, this points play role of limitations to required space transformation:

$$X'_i = g(X_i), \quad i = 1..N,$$

(1.3)

where $X_i$ and $X'_i$ are control points of first and second image, $N$ is amount of control point pairs.

The error $C_2$ in the control point mapping can be used to estimate the accuracy of the selected space transformation:

$$C_2(g) = \sum_{i=1}^{N} \|X'_i - g(X_i)\|^2$$

(1.4)

In such a way, the task of image registration is to find a space transformation $g$ and brightness transformation $f$, which provides a minimum of target function (1.2) or (1.4).
1.2.2. Classification of image registration methods

There are a large number of image registration methods that have been developed for different limitations of source data. These limitations arise when a real practical task is investigated, and they can change from task to task. But methods can have significant differences even for the same limitations. So in order to compare image registration methods and to determine their similar features these methods must be classified by their several characteristics.

The most popular classification scheme that uniformly describes all known image registration methods is the scheme which includes three classification characteristics:

1. Search space – a possible space transformation and brightness transformation between processing images.
2. Type of elements used.
3. The strategy used for optimal solution searching.

These characteristics are not independent of each other. For example, the selection of a typical image element shrinks the possible search space, and both these characteristics determine, in most of cases, the search strategy.

1.2.2.1. Space transformation

According to the task conditions, it is required to find a geometric transformation \( g \) and brightness transformation \( f \), which provides a minimum of a target function. It is impossible to find this minimum among all possible functions \( g \) and \( f \), so the class of possible transformation must be defined.

In most of the work on image registration, the search of brightness transformation is not performed: either the brightness of both images is approximately the same, or the method developed is invariable to a brightness transformation. In this case the brightness transformation can be found (if it is required) after the image registration.

The space transformation can be described in three forms: the global transformation, the local transformation and the optic flow.

1.2.2.1.1 Global transformation

The global transformation defines a transformation that is uniquely applied to all image pixels or elements. It transforms all the area of one image to another image. This can be a transformation from the motion group, a homothetic transformation, affine transformation or projective transformation [Brown, 1992].

The projective transformation is non-linear transformation. It is defined by following equations:
The affine transformation can be produced from projective transformation, if $a_3$ and $a_4$ is zero:

$$x' = \frac{a_3x + a_6y + a_2}{a_3x + a_4y + 1}, \quad y' = \frac{a_5x + a_7y + a_3}{a_3x + a_4y + 1} \quad (1.5)$$

The affine transformation is a particular case of the projective transformation, and it is a polynomial transformation of the first degree.

The homothetic transformation is a particular case of the affine transformation, and it forms a transformation group. The homothetic group includes motion group and uniform scale transformation. The motion group includes rotation and parallel shift. The parallel shift transformation also forms a group.

When the space transformation cannot be described by uniform global transformation or if the transformation model is unknown, other methods of transformation description must be used.

### 1.2.2.1.2. Local transformation

The local transformation, or elastic transformation is described like a global transformation, but the transformation parameters are determined for a local areas. These parameters can either be determined on separate control points and interpolated to a whole area, or they can be a constant in the local areas.

### 1.2.2.1.3. Optical flow

The optical flow or translation field is a set of displacement vectors for each pixel. When scenes contain moving objects, analysis is necessarily more complex than for scenes where everything is stationary, since temporal variations in intensity have to be taken into account. However it is possible to segment moving objects by virtue of their motion: image differencing over successive pairs of frames should permit this to be achieved.

The optical flow can be considered as a discrete or continuous function that must be optimised. Limitations to this function are obtained from priory considerations. The optical flow is used in the cases when the global transformation is absent, and pixel translations are small. In this case a global transformation $g(X)$ can be written via an optic flow $u(X)$ in a following way:

$$g(X) = X + u(X)$$

The problem that requires an image registration applies significant limitations to a geometric transformation. For example, medical applications use three-dimensional affine transformations or continuous optic flows. The stereovision tasks use an optical flow with a single component that oriented among the epipolar lines.

On the other side, the limitations of space transform can be increased by introduction of simplifying assumptions. For example, in air and space...
photographic images registration tasks the affine transformation is often used, and projective components and local distortions are neglected. Stereovision tasks also suppose that global transformation is absent, and the equations of epipolar lines are known. Due to those simplifying assumptions, the image registration task can be solved approximately and applied to the requirements to image obtaining devices.

Another and more widely used form is a limitation to a transformation coefficient value, i.e. only a part of the search space is taken into account. An example of parameter limitation is a limitation of scale. When the scale is significantly different from the one in question, it is required to define its approximate value.

The development of more universal methods remains a research challenge.

### 1.2.2.2. Image representation

There are two common approaches to image element selection: area-based and detail-based methods.

Area-based methods are also known as correlation-like methods or template matching. They use the image pixel as a typical element. The main information is pixel brightness, and the task is reduced to a minimisation of a target function described by equation (1.2).

Detail-based methods use contour pixels, structural or geometric elements, texture spots or even whole objects. Each image element gives a control point with appropriate co-ordinates. The task is to find a correspondence between control points, i.e. to minimise a target function described by equation (1.4).

Both approaches have their own advantages and disadvantages. There are different points of view, as to which approach is the more promising – see [Brown, 1992] for more details.

Area-based methods are more common. They apply no restrictions to image content and allow one to obtain more accurate and reliable matching, since the whole information from the image can be used. But these methods cannot separate the invariant information from the information which is changing from image to image. So they are computationally extensive, and they are unsuitable for the searching of global transformation with large number of parameters.

The detail-based methods require a lesser amount of computations since the source data size is significantly reduced. Furthermore, since the structural elements do not use brightness values directly, these methods are less sensitive to change in lightning conditions, and make them more reliable in several applications. But the detail extraction itself is a complex task. For example, the accuracy of detail detection can be lower than the accuracy of control point selection. The amount of structural elements and the accuracy of appropriate control points are usually limited. That is why details cannot provide information about local translations, and global transformation can be determined inaccurately.

An important feature in the detail-based method is the building of an adequate image description. The process of description building must separate
relevant information, which represent spatial relations between scene physical objects and do not depend on survey conditions. So the detail-based methods play an important role on image interpretation and understanding. The effectiveness of detail-based methods can serve as criterion of model adequacy.

In several cases a mixed approach can be used in order to combine the advantages of detail-based and area-based methods. The first step is fast, reliable, but inaccurate matching by the detail-based method and the second step is to use the area-based method to define a more exact position. There is a method that uses a multi-layer image registration system that uses different presentation of the image with increasing levels of abstraction. They can use different element sets for each level. That is, for example, an areas of pixels with the same characteristics, groups of contour pixels and different structural elements. But the theory of such a hierarchical system is practically absent to this day.

1.2.2.2.1. Image representation methods – an overview

The first stage of image analysis is obtaining its numeric representation. Depending of the image representation, the image analysis methods used can have different principles, different abilities and different characteristics. In Leavers [Leavers, 1992] the requirements for good image representation are reviewed. A good representation must have the following characteristics:
1. Decomposition is the ability to represent an image as a free combination of simple elements. The set of these components must be able to describe any image that can appear in a given application.
2. Accessibility – a representation must be computationally realistic. The image description must be easily obtained and easily processed.
3. Invariance – insensitivity to change in observing conditions – to the image geometric distortion and changes in illumination.
4. Geometric and spatial relations. One shape must have only one representation, and vice versa – one description must correspond to only one shape.
5. Similarity. Representation must describe both similarities and differences of objects.
6. Saliency. Primary characteristics must bring essential information about the object while others must describe the detail description.
7. Stability to noise and incomplete data.

The simplest and most widely used image representation is a rectangular array, each cell of which contains a brightness value of an appropriate object point. The array is called a raster image, and each cell is called a pixel. Such a form of image is usually obtained from a video camera or another source, and then it is used for further processing and obtaining a more compact description. The raster image contains a maximum amount of information about a visual object. This is main advantage, but also a main problem of raster representation: it is practically impossible to process such a lot of information in real time. So a more compact but less full and less accurate presentation must be used instead.
Methods of flat shape representation can be classified in several ways. One method of classifications has been proposed by Loncaric [Sven Loncaric, 1997]. It takes into account:
- Using full information about an object or about its boundary only: boundary, (external) representations and representation of the whole shape (region-based, global, or internal representation).
- A form of representation: a set of numbers (scalars, vectors, matrices etc.) or a set of geometric features.
- A level of information preserving. Information-preserving presentation provides a way to reconstruct (completely or partially) an original shape of an object, while non-preserving representation characterises the shape but does not provide information to restore it.

1.2.2.2. Boundary representation

Boundary representation describes only the shape of the objects on the image regardless of colour, brightness or texture. This provides less information about object but makes the representation insensitive to changes in visibility and illumination conditions. The boundary representation is suitable to describe object contours, coastal lines, roads and general line segments. To obtain a boundary representation, contour pixel detection is required.

**Edge pixels.** The simplest boundary representation is a set of edge pixels. Edge pixels are usually less than 30% of the whole image pixels, and carry the most significant information in comparison to non-edge pixels (inside the object or belonging to background). So the use of edge pixels provides significant growth in processing performance while it represents an original image without significant loss of information. Edge pixels can be also binarized in order to provide a more compact description and change the algebraic operations to a faster logical operation, or information can be supplemented by edge orientation information for more exact edge pixel comparison etc. Nevertheless, such a representation has significant disadvantages: it still is a large description, that is sensitive to noise and to geometric distortions. Now the edge-based shape matching is a widely used approach with well-developed theory – see, for example, [Nack, 1997] and [Mediony and Nevatia 1984].

**Chain codes.** Another simple form of boundary representation are chain codes, proposed by Freeman [Freeman,1974]. Chain codes allow approximating the contour by a sequence of horizontal, vertical and diagonal steps. Beginning from a starting point on the contour, a nearest edge pixel is selected and included to the encoding. When these points are joined together, they form a polygonal approximation of the shape boundary – a chain. The chain is completely specified by a starting point and the sequence of directions required to follow the chain.

A more complex form of chain codes was proposed by Parui and Majumer [Parui et al, 1983] in order to perform a symmetry analysis. The shape boundary is
represented in hierarchy: higher level is rough description of lower number of elements while a lower layer provides a more sharp and exact description.

Chain codes find a wide application in computer graphics systems since they provide good shape approximation. But chain codes are not used in a real world application since they have a lot of disadvantages. Chain codes provide a wide description that is hard to use and process. The description is non-uniform since it depends on a start point selection. The whole code is sensitive to noise and requires an enclosed contour.

**Boundary approximation techniques.** Another representation that is similar to chain codes is a piecewise linear approximation of a boundary [Pavlidis, 1975], [Ramer, 1972] [Rosenfield et al 1973], [Davis, 1977]. Approximation can be done either by looking for the boundary segments that are well fitted by lines, or by looking for break points as boundary segments of high curvature. A further development of piecewise linear approximation is given, for example, in [Bengtsson and Eklundh 1991]. Another paper, [Lagunovskiy, 1997] proposes an approximation-based approach that allows uniting the linear pieces into global linear features.

And further popular method is approximation by splines. An overview of this method is given in [Ikebe and Miyamoto 1982]. Splines provide a more smooth approximation.

The disadvantages of approximation techniques are appropriately the same as chain codes: the same shape can have a significantly different descriptions; the description size is still large and contains a mix of global and local features.

**Salient features.** An approach that has no such disadvantages is a boundary shape presentation as a set of global features: line segments, arcs, polynomial curves etc.. Such a presentation is less exact, but it provides robust and uniform shape representation that is easy to obtain process and analyse. The representation has no exact information for the backward shape restoration, but it provides compact and reliable shape description.

Representation as a set of global linear features may be obtained either by collecting small line pieces into one or by global detection. The line detection will be examined further. Global lines seem to provide good representation due to its universality (many objects have linear features or features that can be roughly approximated by lines) and ease of obtaining and using the description.

Representation by circles and arcs is more complicated since the arc is a more complex object than lines. This representation is usually used if it is known that an image will always have such features. This presentation is more exact, but typically the algorithm of arc detection is slower and less robust than that of line detection. This representation can be a good addition to a line set representation.

One more feature that can be used in boundary representation is a high-power curve such as a parabola or Bezier curve. This can provide an exact contour approximation by a small number of curves, but it cannot provide a uniform presentation: one object can be described in a many ways, so it will be hard to compare or distinguish objects.
**Point features.** Another object group that can be used for representation is point-like features. There are, for example, centres of arcs and circles, points of maximum contour curvature, line crossing points and some application-specific objects, for example – special marker that is used in cartography and medical imaging etc. These features do not preserve all the information about an original shape, but allow simple and reliable image registration.

One of the most important point features are corners. Corners can be considered either as points of maximum boundary curvature or curve interception points. Corners provide very simple and reliable image registration algorithms. A survey of corner detection techniques is given in [Rohr, 1994].

The main disadvantage of an approach based on point feature matching is that the point features are not universal and they may be simply absent on an image.

**Elementary shapes.** It is possible to present the boundary by more complex primitives – as a set of corners, stripes and rectangles etc. These features can be easily detected and processed. The main disadvantage of such an approach is its low universality: a selected feature can simply be absent on the image, or it can present in a qualities that is not robust enough for further processing. Nevertheless, these features can be described as a combination of global lines and can be used successfully as additional to a linear representation.

1.2.2.2.3. **Region-based representation**

The region-based representation uses information about both object shape and object colour (brightness, texture etc.). This representation is less compact but has significant advantages on high noise image processing, when reliable edge detection becomes almost impossible. The region-based representation is better suited to represent objects like buildings [Huertas et al 1988], [Hsieh et al 1992], forests, town areas [Roux, 1996], lakes [Ghostasby et al, 1985], [Holm 1991] and so on.

One of the main problems of the global shape representation is object separation each from other. This is a complex task when colour, shape and position of objects are unknown. Objects can be overlapped or can drop shadows on each other. A comprehensive review of image segmentation methods is given in [Pal, 1993].

One popular technique to separate objects from each other is clustering [Duda et al, 1989]. It allows solution of the task by collecting information about the distribution of pixel position and brightness. It determines which pixel belongs to an object by minimizing some measure, for example – minimal square error. Clustering by itself contains a number of difficult problems. For example, it requires knowledge of the number of objects in the image, the relation between larger object and its parts etc. At the present, these problems are under severe investigation.
Moments. Use of moments for shape description was initiated by Hu [Hu, 1962]. The zero-order moment is the shape area; first-order moments can be used to determine the shape mass centre; second-order moments (moments of inertia) can be used to determine the principal axes of the shape. In his paper, Hu proposed to use invariant moments that do not depend on the position, orientation and scale of the shape. The advantage of moment methods is that they are mathematically compact. The disadvantage is that it is difficult to correlate high-order moments with a shape features. Furthermore, local information and shape features cannot be detected. Targets having an irregular structure require the computation of high order moments for recognition, which require a lot of computations and cannot provide a high accuracy. Another important disadvantage is that if the shape is partially occluded, then the moments of the resulting image are considerably different from those of the unoccluded shape.

Fourier transform. The 2D spatial Fourier transform is another well-known method of the shape representation, which has a highly developed theory and much well tested software. Fourier shape models can be made rotation, translation and scale invariant, and it can be good alternative to the moment approach. The disadvantage is computational extensiveness and impossibility to describe local features of the shape. As in a moment approach, Fourier transform of partially occluded shape radically differs from the unoccluded shape.

Ribbons. The most popular and the most studied global space domain method is the Medial Axis Transform (MAT), also known as prairie fire transform and skeleton transform. MAT method was originally proposed by Blum [Blum 1967]. The idea of this approach is to represent the shape using a graph and hope that the important shape features are preserved in the graph. The MAT description includes the shape symmetric axis (a skeleton) and generating object. The centre of the generating object moves along the axis and changes in size, producing an original shape. The shapes described in such a way are known as ribbons. A good comparison of ribbon-like techniques is given in [Rosenfield, 1986] and [Ponce, 1990].

The MAT approach is computationally realistic, and it produces perceptually meaningful primitives. The main disadvantage is that the generation of the symmetry axes is not a straight forward process and it is not easy to specify how to define a set of generator shapes.

Decomposition. Decomposition is a shape representation as a set of elementary shapes, for example – triangles, circles or simple ribbon-like objects. Decomposition allows representing the shape as a hierarchical set of elementary shapes. The disadvantages are the large amount of objects required to obtain an exact description and the low performance. Decomposition also cannot provide information about object texture.

Areas of interest. Areas of interest are the specific windows having local maximum variances, line interceptions, points of local maximum curvature on contours [Goshtasby, 1988]. Areas of interest include most important information about the image, and can be successfully used for image description and shape matching. The main disadvantage of this approach is the problem with a sharp and
uniformly defining the area of interest: an appropriately the same images can have different areas of interest.

1.2.2.2.4. Summary on image representation

One of the most interesting and promising directions of the shape representation is the representation in a form of line segments. The most important characteristics of line set representation are:

- Line segments are one of the most common and universal objects that allow representation of a wide range of shape boundaries.

- A shape representation in a form of a line segment set can provide simple and fast feature extraction and matching algorithms. This makes it possible to use the approach in real time systems.

- Representation is invariant both for brightness and rigid geometric transformations. The representation is independent on image brightness since only contour shape information is used. The shape information is contained in mutual position of lines, which makes the description invariant to the rotation, translation and scaling.

- Representation provides uniform geometric and spatial relations: One shape has only one representation, and vice versa – one description corresponds to only one shape.

- Representation allows description of both image similarity and significant differences.

- Lines can be used both to bring essential information about shape and to describe shape details.

- And finally, this kind of representation can process noisy and incomplete data. On the one hand, existing line detection techniques allow the finding of lines even in low quality images with significant noise. On the other hand, line matching techniques provide reliable image registration even when a significant part of the lines are distorted or lost.

However, line representation has its own problems that must be solved to provide fast, accurate and reliable image registration.

Extraction of a set of linear features usually requires the additional step of boundary pixel detection. This is a critical step. On the one hand, its precision and reliability directly affect line extraction and matching. So if the precision or reliability is lost on this stage, it is impossible to restore them at a further stage. On the other hand, edge detection can deal with an intensive raw image of large size, so only a limited selection of tools is available to provide real-time processing.

Another problem is providing an acceptable trade-off between accuracy and performance at the line detection stage. Exact methods have a low performance whereas fast methods cannot provide enough accuracy. So it is required to select a balance between accuracy and performance of the line detector.

Final problem is the performance of the line matching algorithm. The line matching approach provides good performance in comparison with other methods,
but the amount of computation grows fast with an increasing amount of lines in the description. The problem is that lines are non-point features, as is required for matching. To obtain a point feature, lines must be combined into pairs to provide a reference point (for example – line joining and crossing point), but it is possible for the amount of line pairs to be too large. So it is required to search for a more effective line matching method.

1.2.2.3. Optimisation strategy

An optimising strategy can consists of two elements – the quality criterion of image matching (optimality criterion) and an optimising algorithm which determines a search order in parameter space.

1.2.2.3.1. Optimality criterion

An optimality criterion depends on selected key elements. The aim for effective optimality criterion selection is to reduce the influence of brightness and geometric distortions.

Since the main parameters of the elements are co-ordinates, the least-squares method is often used (1.4). This equation can have additional items, if structural elements have also non-co-ordinate characteristics, e.g. the size or orientation. Furthermore, since some elements on the image can have no pair, an optimising criterion must include a special item to count a number of matching items.

If the key elements are contour pixel, the optimality criterion can be taken as a distance map. Distance map $D(x)$ is a reflection of contour pixels to an image space:

$$D(x') = \min_{i \in I} \| x' - x'_i \|,$$  \hspace{1cm} (1.7)

and appropriate optimality criterion is:

$$C_3(x) = \sum_{i=1}^{N} D^2(g(x))$$ \hspace{1cm} (1.8)

The criterion (1.8) can be generalised for the case when the edge orientation in contour pixels is known. Edge orientation can be obtained from the brightness gradient direction in that point. This allows finding a real space transformation more accurate and reliable.

If the proper feature-specific is selected or features are point-like, the least square or correlation criteria can also be used.

A least-squares criterion is widely used in the area-based methods. In this case the sum of brightness difference squares is minimised (1.2).

Instead of a least-square criterion, a cross-correlation function is often used [Pratt 1978]. The maximum position of cross-correlation function defines the optimal translation between images, and provides a pair of control points to perform matching. The main advantage of such an approach is the ability to
calculate a cross-correlation function using fast Fourier transformation. The amount of operations required will be reduced from $N^2$ in the case of least squares to $N\log_2 N$.

### 1.2.2.3.2. Optimisation algorithm

Since the optimisation criterion has a many local extremes, it is required to produce an optimisation algorithm able to find a global extreme. In the common case, the global extreme can be found by reviewing all its values. This is computationally extensive procedure; so other methods must be used whenever possible.

If the optimising criterion has a correlation nature, then gradient methods can be used. A set of partial derivatives are calculated at the current point, the direction of quality increasing is determined, and the next point on this direction became a new current point. Such an approach allows examining only a small number of pixels, but it requires a criterion for which the gradient can be calculated. Furthermore, it requires a start point to be situated near to the global extreme position. If this condition fails, the local extreme can be found instead.

The optimisation algorithm must provide a compromise between performance and probability of reaching the global extreme. One way to do this is to use methods of changing resolution, or pyramidal methods. The idea of these methods is to use a whole set of images with resolution growing from rough to sharp. Each iteration uses results of the previous iteration. Since the number of pixels on the rough image is significantly lower, searching requires less time. On the more sharp images a search space is examined only near the point found on the previous step. Furthermore, since noise usually has a high-frequency spectrum, it is suppressed on the low-resolution images, so the reliability of image registration increases.

Using additional characteristics of structural elements can also provide a search limitation. For example, in the case of line segments these characteristics are line orientation and size. This allows putting additional limitations to a possible matching of structural elements (e.g. line orientation and size provides limitation on possible values of rotation and scale).

When the image structural elements are used to estimate the matching quality \( (1.4) \), the possible correspondence between elements of both images can provide additional information to speed up the image matching process.

In the common case, it is required to know only one pair of appropriate point-like features on both images to estimate the position of one image on another, whereas two feature pairs provide additional information about rotation and scale. But this approach does not work in practice, because each feature can have more than one candidate feature on the other image (features, usually, cannot be uniformly identified), some of features can be invisible on the image and others have no pair (if a feature is occluded by another object or cannot be detected due to
low brightness or poor visibility conditions). So the matching process must take into account as many features as possible.

There are three common searching strategies known for the feature matching [Brown 1992]:

- Exhaustive search
- Graph matching approaches
- Clustering approach, or search using Hough transform (HT)

### 1.2.2.3.3. Exhaustive search

In this case, for each possible set of transformation parameters, the image is transformed and then the criterion of matching quality is calculated. The maximum quality criterion value is looked for, and its parameters are the desired object position.

This approach is simple and very reliable, but it requires significant amounts of computation, which depend both on the image size (via quality criterion calculation) and on the search space volume. The approach can be useful when the amount of possible positions is small, or images are small, or the quality criterion can be easily obtained – for example, if the image is binary.

The approach is widely used in observe extreme-correlation systems, when the space transformation is limited by parallel shift group and special devices are used to compute a correlation quality criterion. If other transformations are present, or image size is significant, this approach proves difficult to work in real time.

### 1.2.2.3.4. Graph-oriented search

An alternative to the exhaustive search is a graph-based approach. It can be described as follows. Let us suppose both the observed image and sample are presented as a set of features. Each pair of features that correspond to each other produces a space transformation, which transforms one feature to another. So it is required to find a transformation that leads to the deriving of the maximum matching of features. A systematic way to represent this is to construct a match graph, in which the nodes represent feature assignments, and the arcs joining nodes represent pairwise compatibility between assignments. To find the best match it is necessary to find regions of the match graph where the cross-linkages are maximal.

A complete subgraph, where all pairs of nodes are connected by arcs, is called clique. Hence, the maximum cliques are taken as it leads to the most reliable matches between the sample and observed image. One of the first approaches of graph-based feature matching was proposed in [Marr et al, 1979].

A main problem of the matching graph approach is that there is an exponential growth of required computations as the number of features increases. The quantity of all possible feature combinations that define a search tree depends exponentially on the amount of structural elements. The task of looking for a
subgraph in a graph belongs to NP-complete tasks class (like a travelling salesman task). For such tasks, there is no known means of ensuring that they are executed in polynomial time. Thus, given a graph of $n$ nodes, it is not known how to find the maximum cliques in a time that is bounded by a polynomial of $n$, current indications being that it is at least exponential in $n$. For example, to process a graph of $n$ nodes, it is required to check about $2^n$ nodes in it. So the exhaustive search is acceptable only in a limited group of tasks.

More popular searching methods are the stochastic methods or finite searching methods like dynamical programming, annealing methods, genetic algorithms etc.

In [Barnard et al 1980] a solution to the correspondence problem based on the relaxation algorithm is proposed. Further, the shape registration problem has been formulated as optimisation models and solved by appropriate mathematical techniques. An example can be Lagrangian relaxation [Mundy et al. 1992], [Noble et al 1992], [Ventura et al, 1995], gradient-based descending [Chen and Ventura, 1995], [Ponce et al. 1992].

In [Ullman 1979] the correspondence problem is converted into the problem of finding the maximum weighted cover of bipartite graph, and linear programming methods are applied to this graph problem.

In [Medioni et al 1984] the matching algorithm is applied to the set of linear features. Pong [Pong et al 1989] proposed a hierarchical approach based on Barnard’s scheme, which operates in a two stages. Image matching based on corner points is performed first, providing information for further solving of the ambiguities for the correspondence of the edge points.

Another popular approach in image registration is dynamic programming. It allows solving a problem by dividing it to sub-problems. Larger problems are solved using the best solutions of sub-problem and avoiding redundant calculations. An example of this approach is [Maitre and Wu 1987] – for registration of geographic contour with a map, [Milios, 1989] – for shape matching. Dynamic programming requires that data on both images must have the same intrinsic order, so it can be used in a limited number of applications.

All these methods perform cutting of unresponsive search branches to increase performance. However, this can lead to the selection of non-optimal transformation.

At the current time the matching graph approach appears to be the main direction of image matching method development.

**1.2.2.3.5. Hough Transform strategy**

This strategy uses the transformation similar to the HT for figure position search referred above. Feature space is mapped into parameter space. Each feature pair produces a subset of votes in the parameter space, so vote clusters in the parameter space can be used to estimate the transformation parameters.

Each pair of corresponding elements on both frames produces a curve in the parameter space. The interception point of the curves indicates the possible
position of one image on another. The interception point that belongs to most curves will have the same co-ordinates as the co-ordinates of the sample position on the image.

If the parameter space has more than two co-ordinates, each pair of corresponding elements produce a surface (or hyper-surface), and the amount of calculations increases. However, the amount of computation remains a linear function of the number of elements.

Such an approach originally appeared in [Ballard, 1982], where HT strategy was applied for edge matching. In further works like [Stockman et al, 1982], [Goshtasby and Stockman, 1985] and further, in [Davies, 1992] the method was extended for point-like features.

Shekhar in [Shekhar et al, 1999] proposes a feature consensus mechanism. The idea of this approach is to collect votes in parameter space separately for each feature. The feature consensus mechanism extends the capabilities of HT methods such that complex non-rigid local transformations can be dealt with.

A comparison of graph-based and HT-based strategies is given in [Davies, 1991]. Davies shows that HT strategy is robust and significantly faster, since it has linear computation complexity in comparison to exponential complexity of graph matching strategy. However, HT strategy has some limitations. The votes in parameter space forms spread clusters that are hard to process. So the overall accuracy is low due to a low accuracy in the cluster location. Since all calculations are performed in the image-like discrete parameter space, additional specific problems arise related to quantization and cell value splitting. Another problem is the production of significant amount of false responses.

Another, at present unsolved problem is the growth of required computation when the number of parameters to be determined is high. The problem dimensionality is equal to the number of transformation parameters. Davies [Davies, 1990] shows that the problem can be solved by parameter space decomposition, but this caused significant loss of reliability and can only be used in a limited number of applications.

However, the fast development of HT theory in the recent past makes it possible to re-examine the characteristics of these methods by applying new theoretical results to them.

1.3. Line and curve extraction

There are two common approaches to line and curve detection [Leavers 1992]. On of them is generating curves from elementary line segments and merging them into one. This approach provides high quality contour representation and is widely used in computer graphics and CAD-systems. Another area tracing method application is cartography and geographic information systems. Tracing is
a common technique for extraction of some kinds of features from aerial and space images in order to obtain an electronic vectored map. These features can be roads, buildings, agricultural fields, rivers, lakes and even lesser objects like separate cars.

The common disadvantage of tracing methods is its low performance and high dependence on the image quality. So tracing techniques are rarely used in real time applications, and other methods – faster and reliable – should be used instead.

Another approach for feature extraction is the use of a global integral transformation that transforms the image space into a space of curve parameters. After such a transformation a curve transforms into a separate pixel the brightness of which is equal to amount of points in that curve. This kind of transformation is known as the Hough Transformation. This approach provides faster and more reliable feature detection but it has a less accuracy of contour representation.

The main directions in which modern development of HT theory occurs reflect practical problems that arise in feature detection area. This is a problem of proper representation selection, reducing of parameter space size and dimensionality, selection of authentic responses in parameter space and some other.

1.3.1. Parameterisation scheme selection

The choice of parameterisation scheme used can directly affects algorithm performance, since computations for parameter determination take a significant part of the computation time. Furthermore, it can also affects to the accuracy of parameter determination – the size and shape of parameter space, the accuracy distribution in the different parts of parameter space, and, finally, to the ease of further use for selected presentation.

Historically the first parameterisation was a slope-intercept parametric representation [Hough,1962]: \( y = kx + b \). The \( k \) and \( b \) parameters generates the dual parameter space so that each line \( l \) in original \((x,y)\) space produces one point (the “vote”) in the dual \((k,b)\) space. All points belonging to the direct line \( l \) produces a set of direct lines in the dual space. These lines have the common interception point that is the vote of original line \( l \).

The primary disadvantage is that the \( k \) and \( b \) parameters approach infinity for vertical lines. Kultanen in [Kultanen 1990] proposed possible solution is separate detecting of vertical and horizontal lines. Another disadvantage is significant non-linearity of parameter space: the accuracy of description depends on line position and orientation. For this moment a scheme found a very limited application.

Normal line representation was proposed by Duda and Hart [Duda et al 1972]: \( \rho = x \cos \phi + y \sin \phi \), where \( \rho \) is a length of normal to a line, and \( \phi \) is a slope angle of the normal. This parameterisation allows representation of a line insuspective of direction in a limited parameter space. The main disadvantage of this representation is that it is a computationally extensive (perhaps the highest among other representation used in practice). Furthermore, each edge pixel
provides a sine in the dual parameter space instead of a direct line as in slope-intercept case. So building of exact sine line requires additional computations. Nevertheless, there are number of approaches that can reduce the amount of computation required. So this approach remains a most popular scheme and usually referred as Standard Hough Transform (SHT).

Foot-of-the normal representation proposed by Davies [Davies, 1990]. A foot of normal is used to describe the line instead of normal length and slope angle. The scheme is faster, but has a singularity for the lines that pass through the origin: these lines have a same foot of normal (0, 0) and cannot be detected. In Davies [Davies, 2003] perspectives of practical usage of this scheme are analysed.

A Modified Hough Transform [Wallace R.S. 1985] (also known as a Muff Transform) is another well-known parameterisation. It is a bounded straight line parameterisation that uses the two intersection points of a line with the perimeter of the image. Muff Transform is used in [Shpilman and Brailovsky, 1999] for line segment detection. The major advantage is that the resolution of the lines represented is uniform and matches exactly the set of lines, which can be drawn across the image.

Each of parameterisation has its own advantages and drawbacks. They have different complexity, different accuracy and different singularities. And one of the important tasks is to unite advantages of different methods without significant amount of computations.

1.3.2. Reducing the parameter space

A larger size of parameter space requires not only more memory but also additional computation, related to initialisation, indexation for random access and searching for significant votes and filtering operations. The situation is especially critical when a shape with many parameters is detected and the parameter space becomes dimensionally large. So reducing parameter space size without significant loss of accuracy is an actual problem. One possible approach is to use different quantisation for different areas of parameter space: areas of higher density to have lesser quantisation and a higher amount of counter cells in order to provide significant accuracy, while empty areas have rough quantisation with a small number of cells.

One of the first such approaches is Fast HT [Li et al, 1985]. Li’s approach starts with rough quantisation, and then to decrease the quantisation step in quadrants which obtain an amount of votes higher than a selected threshold level. The disadvantage is significant line splitting.

Another approach is Adaptive HT proposed in Illingworth [Illingworth, 1987]. It provides a special mechanism to focus attention on significant responses to prevent line splitting.

A comparison of dynamical quantisation techniques is given by Princen [Princen et al 1989]. Princen concludes that a main disadvantage of such techniques is the indexation complexity. As a result the increase in performance becomes insignificant, and the algorithm although more complex is less reliable.
Other considerations as how to reduce parameter space size is given by Davies [Davies, 2003]: this paper determines the parts of the parameter space that are not used for transformations and therefore can be eliminated.

1.3.3. Parameter space decomposition

The idea of parameter space decomposition approach is to decompose the parameter space of \( T^n \) elements into \( n \) separate one-dimensional spaces of \( T \) elements [Illingworth et al 1987], each of them being a projection of the whole parameter space to the appropriate axis. This allows significant reduction in a computation and memory requirements, but produces unreliable results when amount of features to be detected is large. This happens because several features can produce the same projection and became inextricable. The presence of noise also makes this scheme very unreliable.

1.3.4. Using additional information

The simplest source of additional information is edge orientation. If it is known, each edge pixel will produce a single vote in the parameter space instead of whole curve. This is important because it leads to a significant reduction in computational load. Davies [Davies 1990] proposes approach to use edge direction information and it is referred as the Generalised Hough Transform (GHT). The problem is that the exact edge orientation is often impossible to obtain due to presence of noise [Leavers, 1993]. Another approach uses gradient direction to significantly reduce sine length. A further approach discussed in the present research is to use information about local point distribution.

1.3.5. Using different analysing orders

In order to get additional information about each boundary pixel a progressive analysing order can be modified. The idea is that every other boundary point belonging to the same detected shape allows a reduction in the amount of parameters to be calculated by one. In such a way, when \( n \) pixels of a shape are known, it is possible to mark a single point in parameter space instead of drawing \( n-1 \) dimensional hyper-surface.

The problem of this approach is that the amount of \( n \) point combinations grows exponentially with \( n \), and it is impossible to process all of these combinations. The one of possible solution is to choose randomly an amount of pixel combination that can be processed in a given time, and find their vote. Such an approach is known as Probabilistic Hough Transform (PHT) – see [Kiryati et al 1991], [Bergen et al 1991], [Xu et al 1990], [Califano et al 1989]. A comparison of different PHT techniques is given in [Kiryati et al 2000].

A further development of this approach for complex shape detection is Dynamic Generalised Hough Transform (DGHT). The DGHT variant proposed by Leavers [Leavers 1990] includes random selection of boundary pixels.
combination, decomposition of parameter space and removing detected features from the source boundary data.

The PHT approaches provide a good reliability when significant noise is present. The main disadvantage is the complexity of the algorithm parameter determination. At the current time, this approach is the one of the most promising directions of HT theory development. Another disadvantage of PHT is its probabilistic characters that do not guarantee the same results for the same original data.

The proposed research analyses another way to use additional information by combining the edge orientation information with the distribution of boundary points in a local area. This allows limiting amount of combinations and increasing performance but provides lower resistance to a noise.

**1.3.6. Line splitting problem**

One of serious problems in HT theory is the problem of feature splitting. Due to the discrete nature of the HT itself, feature parameters must be quantized. In this case a vote in parameter space that corresponds to one line has a chance to be divided into several votes of significantly lesser magnitude. Such votes can be lost among of the noise. Conditions exist when a segment of any large size will produce a set of votes of fixed and low magnitude that do not depended on segment size. In this case a whole long segment will be lost.

The problem of vote splitting is most thoroughly investigated for the case of direct lines and line segments.

Leavers [Leavers 1992] proposed a further approach to prevent splitting by using of post-filtration. Split votes are placed in parameter space not chaotically but in a specific order. If normal parameterisation is used, the votes produce a butterfly-like shape (central cluster with the sinusoidal “wings”). This makes it possible to detect the “butterflies” in order to separate line splits from the noise and to merge them into an original line. The design of butterfly filter is described in [Leavers 1992].

The butterfly filter is the low-level filter that deals with separate pixels, so it size is limited by performance requirements. A common filter size is 3x3 or more rarely 5x5, so it can process only tight split clusters. If splits are situated some distance within the parameter space, this filter cannot merge them.

A more complex butterfly filter that allows separate neighbour lines with significantly overlapped butterflies is proposed by Atiquzzaman [Atiquzzaman and Akhtar 1994] and further, by Kamat [Kamat and Ganesan, 1998].

If line segments are used instead of unbounded lines, then more intelligent algorithms are available – for example, the one proposed by Lagunovskiy [Lagunovskiy et. al. 1997]. In this case splits can be compared in the original spatial space instead of parameter space, so the splits can be found and merged regardless of their parameters. However, line segment detection requires more a complex detection algorithm in comparison to unbounded lines. Line segment
A line segment can be considered as a curve and can be detected by HT. A line segment requires four parameters to be uniquely described. The most comfortable form is a form of co-ordinates of both ends, but it cannot be used with HT since separate edge pixels contain no information about line segment ends. So less comfortable and less accurate forms are used. One of the most popular forms is a set of appropriate line parameters (for example, normal parameters), segment centre position of this line and segment length.

The primary problem of such an approach is that the line segment is considered to have at least 4 parameters to be properly described, and they require a 4-dimensional parameter space. This causes a huge amount of required memory to implement detection, and significant amount of computation as well.

Another problem is the accuracy of line segment detection. An accuracy of line segment detection with this parameterisation depends directly on discretization step and accuracy of the gradient direction detection. Maximum accuracy that can be obtained using differential filters of practical suitable size is about 1° but due to the influence of such factors as image retrace with discretization, natural curvature contour, texture influence etc. is over an order higher.

Since the line segments are nice candidates to a fast, universal and complete description of an image, there are lots of papers that investigate line segment detection problems and propose different techniques to avoid them. In [Davies, 1990] the GHT is used to extract line segments that become a classical approach to the segment detection. Shpilman and Brailovsky [Shpilman and Brailovsky 1999] used a MUFF parameterisation scheme to determine line segment ends as interception points of line and window frame. Kamat and Ganesan [Kamat and Ganesan, 1998] show that the line segment ends coordinates can be extracted from butterfly-shape vote distribution in the parameter space. In [Foresti, 2000] a focusing algorithm is used in order to reduce the size and dimensionality of search space. This approach provides additional information about line segment distribution that can be used for more complex and exact detection. In the [Ching, 2001] the iterative technique proposed that allows avoiding a line splitting. This approach cannot provide high performance but it produces high accuracy.

All these approaches allow to moderate primary line segment detection problems, however, the price is either significant loss of performance or loss of reliability in the complex or noised images.
1.4. Contour detection

Contour detection is a necessary stage of almost all feature-based techniques. The most important and salient features describe object boundaries, or edge pixels; hence, they contain the most important information about object shape.

The theory of edge detection is well researched, and many different tools and techniques are available. However, the demands of real-time application create exacting requirements on edge detection properties and limit the number of methods that can be effectively used. These requirements are:

- **Performance**. The edge detection stage processes the raw intensive images and has to deal with a significant volume of information. Hence, global techniques cannot be used to process the whole image, and local area techniques are used instead. Such approaches can increase performance; however, they may be constrained by their operation on an incomplete set of image information and raise sensitivity.

- **Reliability**. This is the ability to extract edge pixels correctly on images of low quality – noisy, blurred, with geometric distortion etc. Since the reliability and performance are always mutually exclusive requirements, each practical application can provide their own trade-off between these characteristics.

- **Precision**. Many popular edge detectors produce misplaced edge pixels as the price for noise filtration. This causes distortions in object feature locations and creates additional problems on the feature matching stage. Furthermore, to obtain more exact position of edge pixels, more complex – and thus slower – algorithms must be used. Precision is one more components in the performance-reliability-precision triad of mutually exclusive requirements.

- **Contour width**. In the most of cases a contour of one pixel width is required for reliable feature detection. If the width is greater than one pixel, it is required to apply additional algorithms of edge thinning.

- **Enclosed contour**. In some cases the contour extracted must be enclosed and contain no gaps. However, a number of techniques are known that do not required enclosed contours for reliable feature detection, and HT is one of them.

- **Edge orientation information**. Edges can provide information not only about the contour pixel position, but also about the edge orientation in these pixels. The edge orientation is usually found from the direction of brightness gradient in the edge pixels. The edge orientation is an important property for a further feature detection stage. The more exact information about edge orientation that is available the faster feature detection can be performed. However, the precision of edge detection is limited both by local area size used for edge orientation computation, and by the level of noise suppression. Increasing the computation area significantly slows down the computations whereas noise suppression displaces the edge pixel position. So the edge orientation precision is often a parameter that requires accurate tuning.
Insignificant details suppression. Usually the edge detector produces not only salient contours but also other edges that can be caused by noise, texture, tiny details, unsmooth colour changing etc. These items are not desired within feature detection since they can distort results and slow down the process, and in certain circumstances can be eliminated. The main parameter that allows distinguishing these items from the real contours is lower brightness gradient magnitude. So the most popular technique of detail suppression is thresholding, or binarization – an elimination of all pixels with lower gradient magnitude than given threshold. However, proper selection of threshold value and the algorithm itself is a complex and still challenging task.

A contour detection algorithm must contain three stages: edge detection, thresholding and edge thinning.

1.4.1. Edge detector

The most widely used edge detectors can be classified in following groups in relation to the principle used:

–Tracing methods. These methods use bypassing to the perimeter of uniform areas. The trace goes through the contour pixels required. The simplest tracing algorithm is a “bug algorithm” (Fig.1.2). “Bug” begins moving from white area in direction to the black. As soon as it have reached the black pixel it turn to the left and goes to the next pixel. If this pixel is white, bug turns to the right, if white – to the left, and so on until it returns to the original point. Co-ordinates of passage from black to white pixels describe an object contour.

Tracing methods can be implemented to a binary image only, which contains only black or white pixels; hence a thresholding procedure has to be applied to the source image. Thresholding itself is a complex problem that has no universal solution.

The advantages of tracing methods include high precision, one pixel contour width and enclosed contour. Disadvantages are the low performance and low stability. Due to these disadvantages tracing methods are rarely used in real time applications. Their main application area is computer graphics and CAD systems.
–Differential methods. These methods use numeral spatial differentiation. A first or second derivative is usually used (however, in [Mathieu et al 2003] methods are discussed that use derivative of fractional dimensionality). First derivative has a local maximum in the areas of maximum spatial speed of brightness changing, so the position of local maximums can be accepted as object boundaries. Second derivative crosses a zero in the areas of curvature changing, so the zero crossing points can be also accepted as edges. These properties allow building a detector, based on derivative estimation by calculating finite differences in the area around selected point. Second-order methods allow simplifying edge searching, since zero-crossing areas are easy to obtain. First-order methods allow obtaining information about both brightness gradient and edge orientation, and less susceptible to noise, but an additional operation of local maximum searching is required (it is usually referred as edge thinning).

Numerical differentiation operation can be performed as an image convolution with a special matrix (mask), like other filtration operation, so differentiation operation is commonly referred as filtration.

Differential filters are widely used due to their simplicity, relatively high performance and reliability. However, they have significant limitations. A differentiation operation highlights spatial high-frequency elements, so it amplifies noise as well. The common solution is smoothing pre-filtration. However, it can displace edge points from their original location and rounds sharp corners, so it is often unwanted.

A detailed analysis of differential filters design can be found, for example, in [Davies, 1990].

There are two widely known approaches to brightness gradient calculation – template matching (TM) and differential gradient (DG) methods.

TM methods are based on the look for a maximum replay among replays of the filter set (masks). Usually the set contains from 8 to 12 masks. When the mask is applied to the image in a current area, the reply can be found as a sum:

$$R_k = \sum_{i=1}^{n} \sum_{j=1}^{n} I[x + i - n/2, y + j - n/2] \cdot M_k[i, j]$$
where $R_k$ is reply to a mask, $I$ is a image and $M_k$ is a $k$ mask from the set.

Each mask corresponds to one of gradient directions. The direction $k^*$ which produces a maximum reply $R_k^*$ can be taken as the gradient direction estimate. The reply value $R_k^*$ can be taken as a gradient magnitude estimate.

There are number of methods which allow increasing of the gradient estimate accuracy either by increasing the number of masks used (and by appropriate decrease angle quantization step) or by taking into account several near-the-optimal replays.

TM filters were historically one of the first methods of edge detection. It must be noted that each response $R_k$ is the projection of the brightness gradient to the appropriate direction. However, two projections are required to calculate an original gradient vector. This property is used in DG filters.

DG methods, unlike the template matching, require only two masks, which correspond to the two orthogonal directions. In the simplest case one mask $M_x$ correspond to the axis x direction and another, $M_y$ – to the y. Response $R_x$ to the mask $M_x$ is the estimate of brightness derivative in the x direction, response $R_y$ to the mask $M_y$ – in the y direction.

The magnitude of the local gradient can be found from the responses $R_x$ and $R_y$:

$$|G| = \sqrt{R(M_x)^2 + R(M_y)^2}$$

The gradient orientation can be found as follows:

$$\arg(G) = \arctg(R(M_y)/R(M_x))$$

The DG approach produces higher accuracy. The DG approach uses only two responses instead of 8-12 as in TM, but it requires additional calculations for tangent and square root calculation. In practical applications an approximate formula for root calculation and table of arctangent values are usually used to speed up computations.

The DG method based on Sobel 3x3 filter is still the most popular choice for most of practical applications.

–**Statistical methods.** Statistical methods allow avoiding the several disadvantages of differential methods. They work as follows. The statistics of the pixel brightness distribution is calculated in a local image area. This statistics is used to determine the amount of modes in the distribution. If the selected area has no contours, the distribution will have a single mode. Otherwise, if the area contains contours, the distribution will have two or more modes: one mode corresponds to brightness of the main object, and others modes correspond to edge pixels and lesser objects. In this case the main mode must be separated from others, and pixels of other modes must be marked as required contour pixels.

Statistical methods are more reliable to noise in comparison to differential methods. They also do not misplace the contour pixels from its original location. This is important feature for several applications. However, there are kinds of
noise that make statistical methods work significantly less reliable, and require more thoroughly tuning. The properties of statistical methods are investigated much poorer then one for differential methods.

A more detailed survey of edge detection techniques can be found in [Ziou, 1998].

1.4.2. Edge labelling

Differential filters are most popular as tools for edge detection. However, they require additional operations for localising edges and increasing signal-to-noise ratio by suppression of false edges. This operation is referred as edge labelling, or edge procession. The type of edge labelling depends on the edge detector used.

If gradient methods are used, edges can be found as areas of local maximum of gradient magnitude. Earlier methods used a thresholding operation with a following skeletonization. In [Canny, 1983] another approach was proposed – non-maximum suppression. The idea of this method is to extract local maximum in the gradient direction by suppressing pixels with gradient magnitude lower than those neighbour pixels have.

Second-order filters require edge labelling as well: it is required to find the point of zero crossing. The simplest approach is to compare each pixel with its left and lower neighbour. If they have different signs, the pixel is marked as an edge point. However, this simple algorithm will produce phantom edges. For example, pixels of minimum gradient will also produce a second order derivative zero crossing, however, they cannot contain edges.

Another common operation of edge labelling is edge cleaning. It includes elimination of noise and false edges. The problem is that not only object boundaries produce gradient local maximum and second-order zero crossing. Such factors as texture and non-smooth stair-like colour changing also produce them.

The most common approach is to remove responses that have a weak gradient magnitude. As a rule, object boundaries produce higher contrast with background, than texture elements inside the object. So different thresholding approaches can help to solve this task.

In such a way, two operations are commonly used for edge labelling – edge thinning and thresholding.

1.4.2.1. Edge thinning

In most cases a contour of one pixel width is required for reliable feature detection. Since most popular edge detectors cannot provide one-pixel width boundary additional processing is required.

Two common approaches are used for edge thinning: skeletonization and non-maximum suppression.
1.4.2.1. Skeletonization

Skeletonization produces a middle line (usually referred as symmetry axis) of the thick edge. The prairie fire algorithm is a well-known skeletonization method. The boundary pixels of the shape to be thinned are marked as a fired. Each next pass marks shape pixels around fired ones – a distribution of wildfire. Pixels where fire waves meet each other produce the required skeletal line.

The main disadvantage of prairie fire algorithm is that it is iterative, and it can require large, and an unknown number of iterations. In [Choy et al, 1995] another skeletonization method is proposed that requires only two passes.

1.4.2.1.2. Non-maximum suppression

Another approach to edge labelling is the non-maximum suppression (NMS) approach. It was introduced into wide practice of image processing by [Canny, 1983].

The idea of NMS is finding local maximum of the brightness gradient magnitude in the gradient direction, and suppressing all other pixels of non-maximum gradient magnitude. The remaining pixels then produce a required contour of single width.

The algorithm of NMS is quite simple: for each pixel of non-zero gradient magnitude, the neighbour pixels in the gradient direction are analysed. If one of them has a greater magnitude of the gradient, the current pixel is marked. After all are marked, non-maximum pixels are removed, the remaining pixels produces a contours of single width.

The original Canny’s filter also contains a restoration of pixels in the contour direction (orthogonal to brightness gradient) in order to suppress possible gaps in the contour line.

NMS is one-pass algorithm with low amount of computation per pixel, so it is significantly faster than other edge labelling algorithms like skeletonization. However, it is critical to the brightness gradient accuracy determination, and requires additional filtration and thresholding to suppress noise, texture elements and insignificant detail. In comparison to non-maximum suppression, the skeletonization does not use information about edge orientation and thus it is less susceptible to the noise.

1.4.2.2. Thresholding

The task of thresholding is to eliminate pixels with a small value of brightness gradient magnitude and to separate contour pixels from texture elements and noise. These items can also produce local maximums of brightness gradient, and it is required to separate them from object boundary pixels. Since these items commonly have less contrast than the object and background, they have a brightness gradient of lesser magnitude. So the common approach of edge thresholding is to eliminate pixels with small value of brightness gradient magnitude.

Edge thresholding is less simple task than raw image thresholding, since the gradient magnitude distribution in a local area is either unimodal (when no edges
are in the area and thresholding is not required) or bimodal (when edge pixels produce a second mode), so the task is to determine which pixel belongs to each of two modes.

The simplest method is to select a constant threshold value and eliminate pixels with a gradient magnitude less than threshold value. The main problem in the threshold level calculation is that the threshold must be different for different images and even for the different areas of the same image.

To avoid this difficulty it is possible to use the adaptive threshold calculation. Adaptive thresholding uses additional information about, for example, gradient magnitude distribution, in order to select proper threshold level. Adaptive thresholding can use either global or local image characteristics. For the laster, it is possible to apply different thresholds for the differently illuminated regions of the same image.

The thresholding problem is one of the common problems within image processing applications, and it has no universal satisfactory solution to date. A survey of known thresholding methods can be found in [Davies 1990].

1.5. Summary

Image registration based on HT strategy is an interesting and promising technique. It provides a good ratio of performance, accuracy and reliability that is hard to achieve by other methods. The development of HT techniques has accelerated over the recent years, new techniques have been found that allows moderating of the limitations of this approach. A technique such as feature consensus allows not only rigid transformation but also local distortion processing. Parameter space decomposition allows reducing the dimensionality of search space. Probabilistic approaches allow managing an amount of feature combinations to be used for image comparison in order to reduce the amount of computation. Hierarchical schemes, adaptive and hashing techniques allow a “squeezing” the search space in order to reduce memory requirements. All these factors make the HT approaches to be promising in tasks where it is required to find matching of large images with insignificant distortions for a very short time.

One promising feature to provide image registration is line segments. In comparison to point features, line matching requires more sophisticated registration algorithm, that has received little research interest to the date. But the image representation has significant advantages like universality and detection simplicity, which makes them more promising in the processing of a wide range of images of different content, structure and visual conditions.

If it is required and if it is possible, the registration based on linear features can be easily enhanced using other features like clusters and arcs to obtain more a complete and more reliable image description. This can either accelerate the computations if these additional features are easy to find on compared images, or increase reliability in the case of low observing conditions.
Chapter 2. Image contour extraction

Contours play a very important role in the task of image matching. Contour pixels may be considered as control points that can be used directly for image registration. Furthermore, larger and more general features like angles, lines and stripes are a combination of contour pixels. To extract them an additional stage is required – the contour extraction.

Contour extraction is deeply researched but very important stage. If the important details are lost on this stage, it is impossible to correct the situation further. So this chapter is completely devoted to the proper selection of the blocks for the correct contour detection algorithm.

The stages of the contour extraction algorithm are shown on Fig.2.1. There are edge detection, binarisation and edge thinning.

Two algorithms of edge detection will be tested – differential gradient method and statistical method – in order to find best suited for our task. For the binarisation we will compare static and dynamic binarisation. The Canny’s non-maximum suppression method is selected for the final edge thinning because it has no good alternative on performance.

Classical approach usually includes additional stages – image pre filtration (smoothing) and contour enclosing. It was decided to remove these stages by the following reasons. Parameters of pre filtrations mainly depend on characteristics of used camera and must be selected separately in each case. There is no any “universal” pre filtration. The contour enclosing is not used because the line detection algorithm is insensitive to the gaps in the contour, and this computationally extensive stage is not necessary.

The primary characteristics of contour extraction algorithm quality are listed below (beginning from the more important characteristics):

1. Algorithm performance.
2. Reliability, i.e. ability to extract the edge pixels correctly on the images of low quality – noised, blurred, with geometric distortion etc.
3. Precision (many popular algorithms produce misplaced pixels)
4. Contour width. In the most of cases a contour of one pixel width is required. If the width is greater than one pixel, it is required to apply additional algorithms of edge thinning.
5. Enclosed contour. In some cases the contour extracted must be enclosed and contain no gaps.

The integral characteristic of the contour detector quality is a quality of subsequent extraction of larger contour elements – lines and angles, i.e. amount of correctly and incorrectly extracted elements. This characteristic is more important than quality characteristics described above, and indirectly depends on them. So it is important to compare algorithms of contour pixel detection by this integral characteristic.
Fig. 2.1. Contour extraction stages:

a) original image,
b) results of edge detection,
c) results of binarisation,
d) results of edge thinning – the contour
2.1. Differential method

Differential gradient (DG) method requires two masks, which correspond to the two orthogonal directions. In the simplest case one mask \( M_x \) correspond to the axis \( x \) direction and another, \( M_y \) – to the \( y \). Replay \( R_x \) to the mask \( M_x \) is the estimate of brightness derivative in the \( x \) direction, replay \( R_y \) to the mask \( M_y \) – in the \( y \) direction.

The magnitude of the local gradient can be found from the replays \( R_x \) and \( R_y \):

\[
| G | = \sqrt{R_x^2 + R_y^2}
\]

The gradient orientation can be found as follows:

\[
\text{arg}(G) = \text{arctg}(R_y / R_x)
\]

The DG approach requires computations of the root and arctangent, but it provides high accuracy. The following approximations can significantly reduce the amount of computations:

\[
| G | \approx | R_x | + | R_y |
\]

or

\[
| G | \approx \max(| R_x |, | R_y |)
\]

Both formulas provide the same relative error:

\[
\varepsilon = 1 - \cos 45^\circ = 0,29
\]

To increase the accuracy another approximation can be used:

\[
| G | \approx \frac{1}{2} \left( | R_x | + | R_y | + \max(| R_x |, | R_y |) \right)
\]

The relative error is:

\[
\varepsilon = 1 - \cos 22,5^\circ = 0,076
\]

Such an accuracy is enough for a most of practical applications.

The value of arctangent function can be calculated from tabulated values. The problem is that the argument of arctangent must change from \(-\infty\) to \(+\infty\), and cannot be tabulated directly. One of the possible solutions to find the \( \text{arctg}(y / x) \) value is the following.

There are two tables: one for the argument value greater than 1, and another – less than 1.

When \( |x|\geq|y| \), the argument value \( R=y/x \) belongs to a \([-1..1]\) Interval, and arctangent value can be found from the table \( T1: \text{arctg}(R)=T1[R] \).

When \( |x|<|y| \), the additional argument \( R^*=1/R=x/y \) is introduced. It is also belongs to a \((-1..1)\) interval. Arctangent value can be found from the table \( T2: \text{arctg}(R)=T2[R^*] \).

In such a way, values in the tables \( T1 \) and \( T2 \) can be tabulated in the interval of \((-1..+1)\) with any desired step, and the arctangent value can be calculated with any required precision for the any integer values of \( x \) and \( y \). The algorithm requires one integer division and one integer comparison to compute one arctangent value.

The problems of the differential detector development are examined in [Davies, 1990]. The most popular detector wide used in the practice is the Sobel filter with a mask size of 3x3. It provides high enough accuracy (about 3° for the
edge orientation) and requires relatively small amounts of computation. Another useful feature of this detector is the symmetry in relation to a central pixel.

Filters with a larger mask (e.g. 5x5) require in a 2.8 times more computations and provide an accuracy of 1°, but in much of cases such an accuracy is useless since non-instrumental component of the error is significantly greater.

Filters with a lesser mask (e.g. Roberts filter size of 2x2) are too sensitive to noise and image granularity and cannot give the reliable results.

Sobel filter masks are shown on Fig.2.2.

\[
\begin{bmatrix}
-1 & 0 & 1 \\
-3 & 0 & 3 \\
-1 & 0 & 1 \\
\end{bmatrix}
\]

\[
\begin{bmatrix}
-1 & -3 & -1 \\
0 & 0 & 0 \\
1 & 3 & 1 \\
\end{bmatrix}
\]

Fig.2.2. The Sobel filter masks

The DG methods have the following specifics:
- Differential methods by itself cannot find a local gradient maximum, and provides a contour width greater than 1. So after the gradient is found, additional operations to reduce a contour width are required.
- Differential methods use the derivative computation, so they highlight high-frequency elements: noise, corners, tiny details and texture.
- Differential methods always misplace the gradient maximum by one pixel from the real position of edge pixels.

### 2.2. Statistical methods

Statistical methods allow avoiding the several disadvantages of differential methods. They work as follows.

The statistics of the pixel brightness distribution is calculated in a local image area. This statistic is used to determine the amount of modes in the distribution. If the selected area has no contours, the distribution will have a single mode. Otherwise, if the area contains contours, the distribution will have two or more modes: one mode corresponds to brightness of the main object, and other modes correspond to edge pixels and lesser objects. In this case the main mode must be separated from others, and pixels of other modes must be marked as required contour pixels.

To develop a statistical detector, a following task must be solved:
- Determine the image local area size that will be used to collect a statistics,
- Develop the algorithm to determine amount of modes in the distribution,
- Develop the algorithm to determine threshold to separate main mode from others.

2.2.1. How to select the area size for a statistics collecting

The brightness difference between the object and background is often not a constant value neither for different images obtained at different times, nor even for different areas within the same image. So it must be determined for the each local area. The lesser an area size, the more accurate the brightness threshold will be determined. On the other hand, tiny areas cannot provide enough data to determine the number of modes and the threshold value. The best accuracy in the threshold calculation can be obtained for the local area size of 4x4.

An alternative approach is based on the fact that areas for statistics collection and for the contour detection must not be the same. For example, the statistics can be calculated for the area 4x4 and used to detect contours in the lesser area size of 2x2. The accuracy of edge detection significantly increases, the contour width reduces from 4 to 2 pixels, but the amount of computation increases by 4 times.

2.2.2. How to determine a threshold and amount of modes

There are many methods to find the amount of modes in the brightness distribution, which differ both by quality provided and by performance. One of the fast methods is the following.

After collecting data in a local area, the histogram consisting of three rows is built. The central row corresponds to an average value. If the distribution is unimodal, the central row will be greater than any other: \(G_1 + G_3 < 2G_2\). If the distribution is multimodal, this condition is wrong, and threshold can be found between mass centers of the main and additional modes.

The contour extraction algorithm can be described as follows.

1. For the each local area:
   1.1. Find average value \(I_{av}\) of pixel brightness in a current area.
   1.2. Find average value \(I_1\) for the pixels with brightness less than \(I_{av}\).
   1.3. Find average value \(I_2\) for the pixels with brightness greater than \(I_{av}\).
   1.4. Select threshold value \(I_t\) between \(I_1\) and \(I_2\) values.
   1.5. Find amount of pixels \(G_1, G_2\) and \(G_3\) in diapasons \([0 \ldots I_1 + (I_2 - I_1)/4),
   \([I_1 + (I_2 - I_1)/4 \ldots I_2 - (I_2 - I_1)/4)]\) and \((I_2 - (I_2 - I_1)/4 \ldots 255)\) appropriately.
   1.6. If \(G_1 + G_3 < 2G_2\), then distribution is unimodal, and all pixels in the area must be marked as containing no edges.
   1.7. In the other case, if \(G_1 + G_3 \geq 2G_2\), the distribution is multimodal:
1.7.1. If $G_1 > G_3$, then $G_1$ is the main mode. All pixels in the area with brightness greater than $I_t$ must be marked as edge contour pixels.

1.8.1. If $G_1 < G_3$, then $G_3$ is the main mode. All pixels in the area with brightness less than $I_t$ must be marked as edge contour pixels.

There are a number of methods that provide higher accuracy in threshold and number of computationed modes, for example – by approximation of the distribution by orthogonal polynomials. But these methods are more computationally intensive. The algorithm placed above provides better trade-off between performance, accuracy and reliability.

### 2.2.3. Statistical method characteristics

Statistical methods, like differential methods, provide the contour width greater than one pixel, and require additional operations for contour thinning. These operations require knowing the brightness gradient value in the contour pixels found. So the same differential methods will be used, but the number of pixels to be processed is reduced in 5-20 times.

Statistical methods are more reliable to noise in comparison to differential methods. But there are kinds of noise that make statistical methods work less reliably.

Statistical methods do not misplace the contour pixels from its original location. This is an important feature for several applications.

To select the best edge detector, the characteristics of statistical and gradient methods must be compared to examine the quality and correctness of the further detection of more complex contour elements.

### 2.3. Contour thinning

Using the edge detector the pixels with non-zero brightness derivative are found (Fig. 2.3). Often there are too many such pixels and only part of them define the elements of contour. Other pixels found are texture elements, small details, noise and pixels near the real contours (since the object edges are often blurred on the real images), and they must be eliminated.

The pixels to be deleted can be classified into three groups:

1. Pixels with a relatively small value of the gradient magnitude. This can be a texture elements and non-significant details.
2. Pixels with any gradient magnitude near the pixels with a greater gradient magnitude. Such pixels appear due to edge blurring.
3. Separate pixels with a large gradient magnitude. They appear due to high-frequency noise.
Fig. 2.3. Edge extraction and thresholding

a) Source image (model fragment of agricultural landscape)
b) Edges produced by the Sobel filter
c) Edges produced by thresholding
The pixels of the first group can be found by comparison with a local threshold: pixels with gradient magnitude lower than threshold must be deleted. Such operation is called binarization, or thresholding.

The processing of second group pixels is more complex for the following reason. The local maximums of the gradient magnitude are placed near another and form a contour line. The pixels on this line must remain while the pixels out of line must be deleted. One way to do it is to suppress non-maximum values in the gradient direction and highlight the pixels in perpendicular direction. Such operation is known as non-maximum suppression.

The processing of third group pixels is computationally extensive operation. One of the ways to implement it is to use emboss and dilate filters. However, such a filter can damage other edge pixels. Another solution is to process these pixels on the contour line detection stage. Since these pixels do not belong to a contour line, their presence does not affect the line detection process.

### 2.3.1. Thresholding

The task of thresholding is to eliminate pixels with a small value of brightness gradient magnitude and to separate contour pixels from texture elements and noise.

The simplest method is to select a constant threshold value \( L \) and to eliminate pixels with a gradient magnitude \( |G|<L \). The main problem in the threshold level calculation is that the threshold must be different for different images and even for the different areas of the same image.

To avoid this difficulty it is possible to use the adaptive threshold calculation. In this case the threshold value is calculated for local image areas size of from 8x8 to 16x16. Threshold of each area depends on average pixel brightness \( I_0 \) in this area. Then the threshold value is:

\[
L = k \times I_0,
\]

where \( k=1,2,\ldots,1,8 \) is a constant binarization coefficient.

The change of \( k \) value will not cause significant changes of extracted contour quality (as opposed to the threshold value \( L \) itself), and the value of \( k=1,5 \) is acceptable for a most of images.

Another modification of this method is calculating the threshold value \( L \) separately for different gradient orientations. This prevents removing important details near the brighter objects. This method is more computationally extensive (a local direction histogram must be found), but allows an increase in the threshold value without loss of information and hence it reduces the amount of false edges.

The results of both methods are approximately the same. Since the first method is simpler and thus more reliable, it looks more preferable.

Experiments on Fig.2.4 and 2.5 have showed that smooth changing of local threshold level causes smooth changing in number of contour points found and number of line segments detected. This allows using the local threshold level as a tuning parameter for selecting proper line amount. This property will be used in experiments in the further chapters.
Fig. 2.4. Results of the experimental determination of dependence of edge pixel amount from the threshold coefficient.

Fig. 2.5. Results of the experimental determination of dependence of extracted line amount on the threshold coefficient.
2.3.2. Global and local thresholding comparison

A global threshold level is not easy to determine, and manual tuning is often required. A local threshold level is not so critical: the same values of threshold level can be used for wide range of images.

To compare the properties of local and global thresholding, a series of experiments was performed.

First we have tested the thresholding behaviour in the case of brightness changes. Two images – bright and dark – were used for test (Fig.2.6). When global threshold of level 10 was applied to pictures of different brightness, the first picture has been processed properly whereas in the second picture a number of edges are corrupted. However, local thresholding of the same level (120%) allows the correct processing of both pictures.

Then the sensitivity to the level value was tested. We take the bright image from Fig.2.6,a and apply global and local thresholding of different levels (Fig.2.7). Results of global thresholding can change dramatically when a level changes by only one point. However, local thresholding results have no such dramatic changes even when the level changes significantly. The local thresholding appears is much less sensitive to value level variations.

In common case the adaptive threshold technique is far from optimal, but it provides an acceptable quality along with high performance. Some other methods discussed in [Davies 1990] can produce better results but require much more extensive computations, which is unacceptable for real time tasks.
Fig. 2.6. Global and local thresholding for the image of different brightness

a) and b) – original images
c) and d) – global threshold of 10 is applied to pictures of different brightness. Picture a) processed properly but picture b) edges are corrupted.
e) and f) – local threshold of 120% is applied. Both pictures are processed correctly.
Global threshold level is uneasy to determine, and manual tuning is often required.
Fig. 2.7. Global and local thresholding during level changing

a) Global thresholding of $T=3$ is applied
b) Global thresholding of $T=4$ is applied; many pixels have been removed.
c) Local thresholding of $L=120\%$ is applied
d) Local thresholding of $L=150\%$ is applied; most of pixels remain.

Small changes in global threshold level cause a slightly different result – in the case a) a lot of noise points remain whereas in the case b) they are mostly eliminated. But even significant changes of local threshold value in the cases c) and d) produce similar results.
2.3.3. Non-maximum suppression

The thresholding method described above is effective for suppression of noise and texture elements, but it cannot significantly reduce an edge width. For the best case the contour width must be a single pixel to exactly determine the parameters of a line passing through the contour. The wider a contour the lower the accuracy of line parameter computation and the more false lines will be found. So the reduction of contour width is a very important task.

On the other hand, a wide curved contour can be easily approximated by line segments, so a thinned contour must be near-to-linear.

The non-maximum suppression algorithm can be described as follows. For each pixel that potentially contains an edge (i.e. where the gradient magnitude is non-zero) the neighbouring pixels in the gradient direction are examined. If their gradient magnitude is greater than the current pixel, the current pixel is marked as a pixel to be removed (Fig.2.8). After all pixels are processed, marked pixels are removed. This procedure will affect only edge width, and simply searches a local maximum in a cross-section that is perpendicular to edge direction.

Such a procedure is extremely fast because it requires about 3 or 4 logic operations per edge pixel. Results it application is shown on Fig.2.9.

In several cases it is required to implement additional operations to restore and enclose the contour. An example of a method that has such an operation is the well-known Canny filter [Canny, 1986]. These operations are not significant for contour line detection, because modern line detection algorithms (e.g. Hough transformation) do not require the contour to be enclosed.

Fig.2.8. Scheme of edges thinning by non-maximum suppression (darker squares mean pixels with greater gradient magnitude; crosses mark pixels to be removed).
Fig. 2.9. Example of edge thinning

a) Source image
b) Gradient magnitude
c) Results of gradient non-maximum suppression
2.3.4. Use the thresholding and edge thinning at the same time

The use of edge thinning procedure can cause the amplification of noise and the extraction of small detail contours, which cannot be approximated by long lines or by wide arcs and hence, is useless. These items are hard to eliminate by thresholding (Fig.2.10, a). On the other hand, if the thresholding is implemented before edge thinning, it can cause the contour damaging in such a way that edge thinning will work incorrectly (Fig.2.10, b).

The root of problem is each of these procedures destroys information needed by the other. To avoid this problem, both procedures must be implemented independently. The pixels, which must remain after each of procedures, are marked separately. The destination image will have only those pixels that collect both marks.

Such an approach allows solving the problem completely and requires a insignificant amount of additional operations.

2.4. Comparison of statistical and differential methods

Statistical methods are free from many drawbacks of differential methods. They provide thinner and properly place contours and allow finding lines and angles more reliably and accurately. The statistical filter works better on images of average and high quality (Fig.2.11).

On the other side, when the image spectrum contains high-frequency components – noise, small detail and texture, the processing quality of statistical filters is reduced significantly (Fig.2.13).

Differential methods use the thresholding procedure to suppress high-frequency elements. The thresholding in statistical method has a lesser effect: since the edges are much thinner, it is harder to obtain enough statistical data to compute threshold correctly (Fig.2.14). Lower values of threshold do not allow eliminating high-frequency details, and higher values cause important detail elimination.

Another possible solution is to use high-frequency pre-filtration, e.g. Gauss or median filter. Even in this case the problem of the accurate selection of filter parameters still remains (Fig.2.15). Experiment shows that small changes in filter parameters (10% for STD in Gauss filter) causes significant changes in the extracted contour quality. That is why it is impossible to take the filter parameters as a constant.

Traditional methods of optimal filter development cannot be used in this work because the filter parameters must be found and optimised in real time.

In such a way, the statistical detectors look more promising in comparison to classical differential filters, but the problem of selecting the high-frequency suppression parameters is still has no good solution.
Fig. 2.10. Different variants of edge thinning

a) Results of thresholding after the thinning. There is no enough statistics for dynamic level computation, and some contours are corrupted

b) Results of thinning applied after thresholding. Amount of noise and insignificant details are reduced, however, some contours are lost.

c) Final variant—thinning and thresholding implemented at the same time
Fig. 2.11. Statistical methods comparison. Reduction of actual frame size allows producing better edges, however it requires more computations.
Amount of elements found

Differential method:
Edge pixels: 8140 (10%)
Lines: 295
Corners: 374

Statistical method:
Edge pixels: 9392 (11%)
Lines: 362
Corners: 521

Fig. 2.12. Comparison of statistical and gradient edge detector

Statistical methods allow extracting more contour pixels and producing less false elements for a high quality images.
Fig. 2.13. Comparison of statistical and gradient edge detector

If the image contains a high-frequency noise, statistical method does not suppress tiny details. As a result, a larger amount of false lines is found.
Fig. 2.14. Statistical method with pre-processing

A pre-processing with Gauss filter with a small $\sigma$ or with median filter can correct the previous situation, but the common results are worst than the gradient method can achieve.
Thresholding with \( k=80 \) is used

Lines found

Thresholding is not used

Fig.2.15. Statistical filter with thresholding of level \( L=80 \)

A larger threshold level cause the elimination of edge points, because the edges are thin and contain small number of pixels. A lesser threshold level cannot produce the desired effect.

2.5. Conclusion

This chapter examines the most popular methods of edge extraction and labelling. There are gradient methods, second derivative methods and statistical methods.

Statistical methods have received particular attention in the recent past since they looked very promising due to such properties as higher accuracy and noise
stability. Nevertheless, the behaviour of statistical methods has not been investigated systematically. There is a whole set of unsolved problems, e.g. related with pre-filtration parameter selection. Furthermore, there are number of special cases where the edge detection quality will be significantly lower than classical methods can provide.

Brightness second derivative methods are also widely used now. One of most popular schemes is the Marr-Hildred filter. Taking into account of second derivative allows analysing the edge position more thoroughly. Furthermore, it becomes possible to determine edge formation – single step, line, corner, edge crossing, T-formation etc. However, second derivative by itself is more sensitive to noise in comparison to the first derivative, and cannot provide information about edge direction and exact position. So the second derivative methods usually use both first and second derivatives of brightness in order to reduce noise effects and thereby increase edge detection accuracy. Unfortunately, the amount of computations required is raised as well, and makes this approach computationally expensive.

Gradient methods are known to be the first methods of edge detection. They are deeply investigated and widely used in practical applications. They are the fastest and most universal, and allow processing of a wide class images. One of most popular schemes that uses brightness gradient is a Canny’s filter. It is often used as an etalon for the different edge detection method comparison.

The main disadvantages of the gradient methods are limitation on accuracy and noise sensitivity, which cannot be catered for without a significant amount of computation. A gradient filter with a small mask size cannot determine accurately, regardless of noise, neither position nor orientation of the edge. The use of a larger mask causes an insignificant increasing in accuracy, but to realise this small increase many more computations a required in proportion to the number of mask elements.

To exactly localise an edge pixel, gradient methods use either thresholding or non-maximum suppression. Thresholding requires the threshold value to be properly calculated – it is complex task by itself. Non-maximum suppression results depend on proper determination of edge orientation, but the orientation accuracy is low and depends on noise for the masks of practically acceptable size. However, the method of edge labelling, proposed in this work, uses both adaptive thresholding and non-maximum suppression. This allows reducing the main disadvantages of gradient filters while keeping the amount of computation at a low level. So the method proposed in this work has potentially higher stability with respect to noise and higher performance, and can be successfully used for image processing in real time.

In such a way, this research proposes a new gradient-based edge detector. The proposed method combines well-known techniques in an optimal way, which allows provision of a good ratio of performance, accuracy and stability to noise. The method in certain circumstances has the potential to better characteristics than other methods provide, so it can be used as a first stage in the image registration process.
Chapter 3. Line and corner detection

In this chapter the methods of contour element detection will be reviewed and analysed. These elements are lines, line segments, corners, curves, arcs etc. The use of contour elements instead of image pixels leads to a significant simplification of the further task of image registration and recognition, but requires special detection methods. These methods must provide enough performance, accuracy and reliability.

3.1. Contour representation

A contour can be represented in several forms. Possible forms differ by description method performance, ease of further processing, description completeness and the overall number of objects required for representation.

The simplest way is to present a contour as a set of edge pixels. This way provides easy and fast method to obtain a contour description, but it operates with edge pixels and requires a lot of computation to compare two contours in this form.

Another way is to present a contour as a set of curves, e.g. polynoms or arcs. Such a description provides a complete and exact description, uses a small amount of objects, and requires only an average amount of computation. However, such a representation has the problem of non-unique representation: the same contours can have different representations. This causes a significant complication when attempting to compare contours.

One more approach is to approximate the contour by a set of line segments. On the one hand, lines provide a unique contour representation, can be easy detected, processed and compared. On the other hand, such contour representation is less accurate and less complete. For example, it would be hard to represent images that contain circles of small radius.

Another approximation is to present a contour as a set of corners and lines connecting them. The use of corners causes a significant increase in comparison performance due to corner properties and their small number. But this representation is even less accurate than line approximation. Another problem is that a contour on the raw image can have small number of corners (or not even one), and the comparison algorithm will fail.

In such a way, curves and arcs are hard to detect, hard to compare, but provide a most complete description. Line segments are easy to detect, average to compare, and provide average accurate description. Corners are easy to detect (when lines already found), easy to compare, but they provide poor description.

So the contour representation as a set of line segments looks more preferable than other methods.
3.2. Line detection

Many modern methods of line detection are based on the HT. This transformation allows converting the image pixel space to space of line parameters by such a way that local maximums in parameter space correspond to lines in the original image. It is convenient to use a length $\rho$ and angle $\varphi$ of normal vector to a line as this line parameter. Appropriate line equation can be present in a form:

$$\rho = x \cos \varphi + y \sin \varphi$$

where $(x, y)$ are co-ordinates of points belonging to this line.

A scheme of normal parameters $(\rho, \varphi)$ calculation is shown on Fig.3.1.

Fig.3.1. Scheme of calculation of normal parameters $(\rho, \varphi)$ to a line

Fig.3.2. Scheme of line detection
Normal to a line which contains current point \((x, y)\) with brightness gradient \((g_x, g_y)\) crosses this line in point \((x_0, y_0)\):

\[
x_0 = v \times g_x, \quad (3.1)
\]
\[
y_0 = v \times g_y, \quad (3.2)
\]

where \(v = (x \times g_x + y \times g_y) / (g_x^2 + g_y^2)\).

Then normal length \(\rho\) and angle \(\phi\) is:

\[
\rho = (x \times g_x + y \times g_y) / (g_x^2 + g_y^2)^{\frac{1}{2}} \quad (3.3)
\]
\[
\phi = \arctg \left( \frac{y_0}{x_0} \right) \quad (3.4)
\]

The algorithm of line detection may be described as follows: for each edge pixel \((x, y)\) with brightness gradient \((g_x, g_y)\) according to (3.3) and (3.4) a normal parameters \((\rho, \phi)\) are calculated, and appropriate counter \(C[\rho, \phi]\) is increased by one (Fig.3.2). After all pixels are processed, maximum counter values \(C[\rho^*, \phi^*]\) will represent line segments on the original image. Once such maximums are found, their co-ordinates \((\rho^*, \phi^*)\) are considered as required line parameters.

### 3.3. Arc and curve detection

Objects like arcs, curves and line segments can be also detected using GHT [Leavers, 1992 b]. In the common case, the GHT allows the detection of the parameters of any curve if its analytical description is known. The calculation scheme is the same: for each contour pixel, with known brightness gradient value, parameters of appropriate curve are calculated and accumulator call in parameter space is increased. After that, greater values of accumulators will correspond to curves to be detected.

The main difference between lines and other curves is that the line requires only two parameters to be uniquely described while all other curves require three or more. Every edge pixel with known gradient provides information only about two parameters. For example, each edge pixel provides information about a set of circles that can contain it. Circles can be of different radius, but their centres lie on one line (Fig.3.3).
Fig.3.3. Scheme of circle detection. Each edge pixel provides information about a set of circles that can contain it. Circles can be of different radius, but their centres lie on one line.

In the case of lines, every edge pixel allows calculation of a point in parameter space that uniquely characterises the line passing through the pixel. In the case of curves, every edge pixel does not allow the calculation of appropriate curve parameters exactly. Instead of this, it provides information about a curve in parameter space, and one of the curve points is a parameter of the original curve passing through the current edge pixel. When several edge pixels have been processed, they produce a set of curves in parameter space. If all pixels lie on the one curve, all curves in parameter space cross at one point, and the appropriate counter in parameter space will be maximum value, and co-ordinates of this cross point are the parameters of the curve that has been detected.

An algorithm of curve detection, based on GHT, must contain an additional step – generation of a curve in parameter space, so the performance is lower. It also has lower accuracy, since the cross point in parameter space must be found.

3.4. Corner detection

There are two common approaches to corner detection. One of them is local feature detection and uses the special filter. It operates with separate pixels using a set of masks like differential or smoothing filters (Fig.3.4). Each mask provides the detection of appropriate oriented angle [Bretschi, 1981]. Another approach is the global feature detection. It looks for corners as interception points of different lines or curves (Fig.3.5). In the case of the line segments, if co-ordinates of line segment ends are known, the corner position interception can be found as interception point of two lines passed through an appropriate line segment.

From the Fig.3.5, since

$$\tan \alpha = \frac{Y_A - Y_B}{X_A - X_B} = \frac{Y_A - Y_E}{X_A - X_E}$$
\[\tan \beta = \frac{Y_C - Y_D}{X_C - X_D} = \frac{Y_E - Y_F}{X_E - X_F}, \]

then:

\[
\begin{align*}
- X_E(Y_A - Y_B) + Y_E(X_A - X_B) &= - X_A(Y_A - Y_B) + Y_A(X_A - X_B) \\
- X_E(Y_C - Y_D) + Y_E(X_C - X_D) &= - X_C(Y_C - Y_D) + Y_C(X_C - X_D)
\end{align*}
\]

is a system to determine an angle vertex \((X_e, Y_e)\). The solution is:

\[
\begin{align*}
X_e &= \frac{[Y_A(X_A - X_B) - X_A(Y_A - Y_B)](X_c - X_D) - [Y_C(X_C - X_D) - X_C(Y_C - Y_D)](X_A - X_B)}{(X_A - X_B)(Y_C - Y_D) - (Y_A - Y_B)(X_C - X_D)} \\
Y_e &= \frac{[Y_A(X_A - X_B) - X_A(Y_A - Y_B)](Y_C - Y_D) - [Y_C(X_C - X_D) - X_C(Y_C - Y_D)](Y_A - Y_B)}{(X_A - X_B)(Y_C - Y_D) - (Y_A - Y_B)(X_C - X_D)}
\end{align*}
\]

The results of global corner detection are shown on Fig.3.6.

<table>
<thead>
<tr>
<th>0°</th>
<th>45°</th>
<th>90°</th>
<th>135°</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>5</td>
<td>5</td>
<td>5</td>
</tr>
<tr>
<td>4</td>
<td>5</td>
<td>4</td>
<td>5</td>
</tr>
<tr>
<td>4</td>
<td>4</td>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>4</td>
<td>4</td>
<td>4</td>
<td>4</td>
</tr>
</tbody>
</table>

Fig.3.4. Masks for local corner detection

Fig.3.5. Scheme of global corner detection
The first approach is faster and allows finding many of the local angles to be found. On the other hand, the second method is more accurate and reliable, and can process images of low quality. Usually a corner by itself cannot be used directly in the tasks of image comparison or recognition. They serve as additional information for further processing. So the presentation of image in a form of curve or line set is usually available for the stage of corner detection. In this case the second approach looks more preferable.

### 3.5. Line segment detection

A line segment can be considered as a curve and can be detected by GHT. A line segment requires four parameters to be uniquely described. The most comfortable form is co-ordinates of both ends, but it cannot be used with GHT since edge pixels have no information about line segment ends. So a less comfortable and less accurate form is used. One of the most popular forms is a set of appropriate line parameters \((\rho, \phi)\), segment centre position \(k\) on this line, and segment length \(l\).

Accuracy of line segment detection with this parameterisation depends directly on discretization step and accuracy of the gradient direction detection. The maximum accuracy that can be obtained using differential filters of practical suitable size is about \(1^\circ\) but due to the influence of such factors as image retrace with discretization, natural contour curvature, texture influence etc. is significantly lesser.
In Fig.3.7 a sample of ideal edge processing by a 5x5 filter size is shown. For the undistorted edge at 18.42° (arctg 1/3) an exact angle value cannot be calculated at any point. For 2/3 edge points a value of 21.8° (arctg 2/5) would be calculated – error is 3.38°. In other 1/3 points angle would be 11.31° – appropriate error is 7.12°. Because filter size of 3x3 is usually used in practice and real edges are far from ideal, the result can be much worse.

The error of parameter $\phi$ determination depends directly on the error in the calculation of brightness gradient direction and of error that appears due to angle discretization. The error of $\rho$ parameter calculation is less, and can usually be ignored (it is less than one pixel and hence cannot affect the image processing result).

Fig.3.8 shows a calculation scheme of error in a line segment end position. Due to presence of error $\varepsilon$ in normal angle determination, the real end of segment A will be situated in point A'. Triangles OAB and O'A'B' are equal (as rectangular with both equal legs), so OA=OA'=R and $\angle AOA'=\angle BOB'=\varepsilon$. Then the error, $\delta \lambda$, of a line segment end position can be found as the isosceles triangle AOA':
\[ \delta \lambda = 2R \sin(\varepsilon/2), \]  

(3.10)

where \( \varepsilon \) is error in \( \phi \) parameter determination,

\( R \) is distance from co-ordinate centre to the end of segment.

Thus, even the brightness gradient is calculated exactly, a shift of line end would be relatively large (about 1 pixel on the segment size of 64 per each error degree). For example, if there is an error of 3° in edge orientation (as in the previous example) for the images size of 128x128 a maximum error of the end segment position will be 8 pixels even for an ideal image. For many practical applications this value is too large.

The second disadvantage of the classical approach is related to the necessity of quantization. When the quantization step is large, an error of parameter determination is large too. If the quantization step is small, an interesting effect of line splitting arises: instead of one separate line a set of parallel lines is detected. These lines have the same pixels but are geometrically different. An additional factor that leads to the splitting can be seen on Fig.3.7: differential filter gives not one exact value of edge direction but a whole set. Each mode of this set would generate a separate set of parallel lines.

The additional negative effect of splitting is the fact that it is impossible to determine parameters of the original line by analysis of a split set. As a result, the decreasing of the quantization step causes splitting instead of growing of parameter extraction accuracy.

### 3.6. Proposed method of line segment detection

#### 3.6.1. Extended Hough transformation

The main problem of known line segment detectors is that the parameters used for line extraction provide poor line description and vice versa. The solution is to use different sets of parameters for the line segment identification and for the complete description.

At a first look, it seems impossible since a parameter set is used to create a detector parameter space and must be the same both for identification and description. The classical approach requires that all description parameters must be included into the set of co-ordinates of the search space. This causes significant increasing the dimension of search space and raises problems related with it, so in practical applications the use of these additional parameters is usually avoided if possible.

However, there is another way to take into account the set of additional parameters. It is required to extend the accumulator cell in such a way that each cell will keep not only accumulator value, but also all additional parameters related with current line.
The detection procedure will be changed in a following way. After the normal parameters are calculated and an appropriate accumulator cell is found, the accumulator value is increased, and all additional parameters are modified – the current gradient direction value and line end co-ordinates are corrected.

Line end correction is a very simple procedure. It is required to know the area co-ordinates, which completely includes the line. These co-ordinates are minimal and maximum values among all line points of co-ordinate values $x$ and $y$.

The correction of the brightness gradient direction along the line is also simple. A gradient vector is always orthogonal to line direction and parallel to the normal vector. Its direction is either the same or opposite of the normal. To compute the gradient direction, it is enough to introduce a counter that will increase each time gradient and normal directions are the same, and decrement when they are opposite. As a result, a value will be obtained which provides information about the prevailing gradient direction. It must be noted that obtained value depends on co-ordinate centre position. So the calculations must use angle of gradient direction instead of it to provide translation invariance.

In such a way another parameters can be calculated – the arc start and end angle, amount and average length of gaps in a dotted lines, and other statistics that can simplify the shape detection on the image.

We will refer to this approach as Extended Hough Transformation (EHT). It is equivalent to a group of HT-like transformations, when each of them corrects only one line parameter – amount of pixels, one of the end co-ordinates and so on. However, the cell co-ordinates are common for all of them and are calculated only once.

What result is obtained? Instead of straight line detection, which is described by the pair of parameters, a procedure of line segment detection is obtained. The classical approach requires 4-dimensional accumulator array, whereas extended HT uses only two dimensions. Reducing the parameter space dimensionality becomes possible due to the following. HT includes two processes – object identification and object description. These processes are similar but not equivalent. In the classical HT these processes are united in one, so the identification feature is also used for object description. However, it is required sometimes to use different identification and description parameters. For example, line segment is usually presented in co-ordinate form (via segment end co-ordinates) instead of normal form, used in classical HT (normal vector, segment length and centre). Normal form provides significantly less detection accuracy: small distortion in normal parameter determination can lead to significant displacement of a line segment. At the same time, small distortion in a segment end co-ordinates cannot cause significant displacement, so co-ordinate form is preferable. However, it is impossible to provide a HT scheme that can use segment end co-ordinates as detection parameters, because they are not dependent on edge pixel co-ordinates and brightness gradient.
To produce a new line segment detection algorithm, two considerations of Extended HT must be taken into account.

The first is that the counter cell can keep not only a counter value (amount of object with current parameters) but also any other information that can characterise an object. This additional information can be used for the accurate and complete object description. It can be, for example, co-ordinates of segment ends. In practice it is better to store the maximum and minimum co-ordinates of line segment pixels since end co-ordinates can easily be obtained from them. In this case the maximum error of a segment end position is dependent completely on the average width of the edge. Assuming that the edge width is one or two pixels, the accuracy of line segment localisation will also be a two pixels and not dependent on the line position on the image, in contrast to existing HT based algorithms.

The second idea is that the line segment on an image can be uniquely identified by less than four parameters. For example, two crossing line segments that belong to one line is one larger line segment. Non-crossing segments on the same line are often one segment with a breakout. In such a way, in much of cases the line segments on the one line can be considered as one segment and it can be identified by appropriate line parameters.

The problem of such approach arises when the different line segments (e.g. on the different sides of image) belongs to one line. But another hidden parameter can be used – a number of calculation iteration. Since the image processing goes in default order (e.g. from the left to right and from up to down), it can be used to monitor separate segments, which belong to one line. The monitoring can be easy performed by comparing current pixel co-ordinates and co-ordinates of segment ends stored in the counter cell. If the gap is large enough the current pixel belongs to a new segment and previous segment must be moved from the counter cell to a list of detected segments since all of its pixels are already processed.

To increase detection accuracy a following method can be used. Since the gradient direction is known, it is possible to directly examine the pixels, which probably belong to a current line. Each gradient value corresponds to a sector where such pixels can be found. So the exact direction of a pixel can be used instead the approximate gradient direction. The main drawback of this approach is that the several pixel pairs can be processed instead of one gradient value, but growth of accuracy is significant, and such an approach can be used in practical applications.
The modified algorithm of line segment extraction will look as follows. There are map of gradient magnitude |G| and phase F, empty list L of detected line segments and array C[\( \rho, \phi \)] of detector cells. Each cell is a structure with a field shown on Fig.3.9. At the first moment all fields filled by zeros.

I. For each pixel \((x, y)\) with brightness gradient \((g_x, g_y)\):
   1. Determine the sector on image, where the some other pixels of a current line probably can be found.
   2. For each edge pixel \((x_1, y_1)\) from this sector:
      2.1. Calculate the line parameters \((\rho, \phi)\) and look for counter cell C[\( \rho, \phi \)]
      2.2. If the cell is free, store current pixel \((x, y)\) as a begin and end of a new segment, and go to step 2.
      2.3. If the current pixel \((x, y)\) is near to one of stored segment ends, this end must be corrected, and counter field increased; then go to step 2.
2.4. If the current pixel \((x,y)\) is far from any stored segment ends, then the cell content must be moved into list \(L\), if the segment is long enough, and current pixel \((x,y)\) stored as a begin and end of new segment. Then go to step 2.

II. Move all long segments from detector cells \(C\) to a list \(L\).

III. ”Turning” the segments with direction in quarters I and III:

\[
\text{if } L[i] \phi \in [0..\pi / 2] \cup [\pi ..3\pi / 2] \text{ then exchange } L[i].Ymin \text{ and } L[i].Ymax
\]

3.6.3. Performance of proposed algorithm

Calculation time is easy to determine from the algorithm scheme. For each edge pixel a line parameter \(<\rho, \phi>\) must be calculated, and this pixel must be checked if it is a new segment end. According to (3) and (4), calculation of \(\rho\) and \(\phi\) requires:

- An integer operations: 4 multiplication, 2 addition and 1 dividing;
- Float point operations: 4 multiplication and 2 addition;
- Transcendental functions: 1 arctangents and 1 square root;

Ends processing require 4 integer comparisons;

For processor that was used\(^1\) (Duron\(^\text{tm}\) 750, 100MIPS/2MFLOPS, virtual 8086 mode) total calculation time would be:

\[
T1 = 2.54*10^{-5} \text{ – time of } \rho \text{ and } \phi \text{ calculation per each pixel;}
\]

\[
T2 = 0.4*10^{-7} \text{ – time for ends comparison per each pixel;}
\]

\[
\epsilon = 100\% \times \frac{T2}{T1+T2} = 0.15\% \text{ – fraction of comparison in total calculation time.}
\]

3.6.4. Gluing of line segments

As was noted before, the parameter space discretezation produces some negative effects. One of them is line splitting. If the line direction cannot be presented exactly in the parameter space, such a line would be splintered into one or several sets of parallel lines with rounded value of the normal angle (Fig.3.10). The same problem can arise also due to contour curvature (when contour is really not a direct line but it is required to approximate it) or when contour width is more than one pixel (or, more generally, when contour pixels cannot be exactly approximated by a single line).

\(^1\) This processor was available at the early beginning of this work, and now it looks old and slow. Further we have used more powerful hardware; however, all results were rescaled for this processor to represent data in a common basis. It is impossible to predict the speed of progress; so we placed the results in a form suitable for extrapolation to any system of known performance.
The problem is that the restoration of a source line from a set of splits is a very inexact when the normal representation of a line segments is used. But the proposed approach operates with end co-ordinate representation, and this allows the development of an accurate and effective gluing algorithm for splits.

All line splits can be classified in two groups. The first group is splits of direction near to an original line. This group contains most important information. All other splits form a second group. They are less important and can be ignored.

The use of normal parameters will not provide an ability to look for splits since segments of same parameters can be situated in different parts of the image. Instead of this, an approach based on line segment end co-ordinates can be used.

The two splits that belong to one line can be displaced one from another in perpendicular and co-axial directions. The shift in coaxial direction can be relatively large, and can be caused by gaps on a contour, which is caused by overlapping by another object, bad image quality or missing a part of a contour at the thresholding stage. The shift in perpendicular direction is limited by values of 2-3 pixels. This shift can be estimated via the distances from the ends of the shorter line segment (split) to a line passing through the larger segment (base line):

$$L = \sqrt{a^2 - \frac{(a^2 - b^2 + c^2)^2}{4c^2}}$$

where $a$ and $b$ are the distances from end of split to a appropriated ends of base line segment,

$c$ is the length of the base line segment.

If the distance $L$ is less than the critical value of 3 pixels the current segment is considered as a split of the base line.

The shift value $S$ in coaxial direction can be estimated as a projection of maximum end shifts to a direction of the base line:

$$S^2 = \Delta X^2 + \Delta Y^2 - L^2$$

where $\Delta X$ and $\Delta Y$ are the difference between co-ordinates of appropriate ends of the current and base line segments.

Due to a strict limitation to $L$ value, it can be avoided:

$$S^2 = \Delta X^2 + \Delta Y^2$$

Another principal question is whether the current line segment is a split of base line, or it is the far coaxial separate line segment. The segment is considered to be a split if one of its ends situated between the ends of the base line segment (with precision $\varepsilon$ ), and the following condition must be true for the any of the ends or for both of them:
where \((X,Y)\) is a end co-ordinate of a current segment, \\
\((X_1,Y_1)\) and \((X_2,Y_2)\) are the minimum and maximum co-ordinates of a base  
segment.

The gluing algorithm can be described as follows.

- **The first pass** is a first group processing. The line list must be sorted  
  according to line length (or alternatively, according to amount of edge pixels that  
  belong to the line, if this statistics is available). Then each line of the list is  
  processed. Potential splits for each line are found, checked and glued, if required.  
The glued segments, of course, are removed from the list. In such a way, each  
segment pair that has equal brightness gradient direction \(\gamma\) and normal length \(\rho\) is  
compared. If one of the end pairs is situated nearer than the required distance,  
such a pair considered as one broken line with end co-ordinates equal to  
appropriate maximum or minimum from both sets. If both ends coincide then a  
line splitting is considered. The correcting procedure is the same.

- **The second pass** is processing of the second group. For each line from list  
potential splits are looked for. The line is a split of the selected line if it has near \(\gamma\)  
parameters and its line ends situated near from the main line. Such splits must be  
simply removed.

The main known problem of the described gluing algorithm appears when  
there is an object on the image that produces line sets similar to the splitting  
picture (i.e. set of long parallel lines). In this case a whole set of parallel lines  
would be replaced by one line of the same direction in the middle position. Since  
the values of maximum and minimum end co-ordinates are used to restore an  
original line, all restored lines will look like rotated by a small angle: both ends  
will be shifted by a couple of pixels, but in different directions.

The gluing operation allows reducing the amount of segments to 20-45%  
without loss of information. Gluing effectiveness depends mainly on the type of  
image, quality of contour pixel detection and the total amount of lines extracted.

### 3.7. Experiments

In order to determine the characteristics of the proposed method of line  
segment detection, a series of experiments have been undertaken. The task is to  
examine both common characteristics (performance, accuracy and reliability), and  
additional characteristics that is specific for this method (metrics selection and  
minimal curvature radius that can be approximated by lines).
3.7.1. Performance analysis

One of the most important characteristics of line detector is a performance. The x86 family processors will be used to calculate the ratio of different operations.

An “equivalent operation” term will be used as a measure value. It considers all operations that require the same computation time as integer summation operation – e.g. subtraction, comparison, shift and logical operations. For the x86 family an integer multiplication and division requires ten times more computation time.

To simplify the total time calculation, the amount of computations per one pixel will be found at first, and then the results will be obtained for a whole image.

As it was described above, the line segment detector must include the following iterations (Fig.3.11). The scheme with a differential edge detector is considered.

- **frame capture and pre-processing.** This iteration provides a frame capture and copy into operative memory. The pre-processing is extraction of odd and even half-frames, and it can be included into frame copying, and do not require an additional time.

- **edge pixel extraction.** The Sobel filter size of 3x3 requires 20 additions (ten for each mask) per each edge pixel. It is also requires two module calculations and one division. The value of arctangent can be tabulated as it was described above, and one more integer division and comparison are required. So the total amount of \(20 + 2 + 2 \times 10 + 1 = 43\) operations to process each image pixel.

- **edge thinning and thresholding.** The adaptive thresholding with a block size of 12x12 requires one addition and one comparison per pixel. It is also requires one multiplication for an each block, or \(10/144 = 0.07\) equivalent operations per pixel. So the thresholding requires \(2.07\) equivalent operations per image pixel. On the edge thinning procedure each pixel must be compared with two neighbours and one logical “AND” to pixel suppression. In such a way, five equivalent operations must be performed to process one image pixel.

- **line segment extraction.** As it was found above, it requires about 500 equivalent operations per edge pixel. Since the edge pixels are about 25% of total image pixels, the approximate value of \(500 \times 0.25 = 125\) operations must be performed to process one image pixel.

- **segment gluing.** The time required for a gluing operation depends on the total splits amount, and cannot be easy recalculated in a term of the operation per image pixel. But since the gluing operation deal with small amount of lines, it requires a relatively small amount of operations, and can be neglected.
Fig. 3.11. Scheme of line segment detector
The full detection procedure without line gluing will require 43+5+125=173 equivalent operations per image pixel. For the Duron™750 processor in the virtual 8086 mode (96 MIPS) one frame size of 128x128 will be processed for a T=0.033 seconds (without time for image capturing – 0.04 seconds for standard video systems).

Results of experimental measurements of performance are shown on Fig.3.12.

In such a way, the performance of the proposed algorithm is high enough to use it as a base step for a wide range of practical applications.

**3.7.2. A metrics problem**

When the image is captured by hardware, a series of discretization processes are performed. The some of them, e.g. discretization by a time and by a signal level – are well known in an engineering practice and thoroughly investigated. But there is another process – a discretization by a position on the image – must be kept in mind.

The image capturing performed by a sequentially obtaining pixel rows and writing them into a two-directional array. So each brightness value is banded to place in the rectangular grid of the digitized image.

The several problems arise due to such kind of discretization when detecting the object edges (Fig.3.13). One of them is that only the lines of 0, 45 and 90 slope angles can be accurately presented. All other geometric lines will have gaps since they cannot pass through the neighbour pixels. If the arctangent of a line inclination angle is an irrational value (e.g. 30 degrees angle), such a line will exactly pass through only one image pixel (and of course cannot be detected). It

![Pie chart showing time required for line segment extraction stages, sec](image)

Fig.3.12. Results of experimental performance measurements
must be noted that the original line can be geometrically straight, but it can be a curve on the image.

To avoid this problem, the “line” term is extended: the image line is actually the set (may be even infinite set) of geometric lines, and the distance between the line set is less than one pixel. For example, middle line on the Fig.3.13 actually the set of two parallel lines (they are marked by a different brightness). Such an approach allows detecting the line of any orientation. Actually the line presentation as a set of geometric lines will raise a line splitting problem, but it can be easy solved.

Another problem is that line segments of same size will has a different amount of pixels (Fig.3.13). The horizontal line segment size of 16 pixels consists of 16 pixels, while the diagonal segment of the same size will have only 11 pixels (\(\frac{16}{\sqrt{2}} = 11.23 \approx 11\) pixels). The solution of this problem is use of chessboard distance instead of Euclidean metrics in all formulas of line detection:

\[
L = \max(X, Y)
\]

In this case all line segments of the same size will have the same amount of pixels, and detection will be anisotropic. The Fig.3.14 shows the difference between different metrics for a line detector.

It must be noted that the space anisotropy problem exists only on the line detection stage. When the image is already presented as a set of lines, the problem disappears, and all operation (e.g. line gluing) must use Euclidean metric.

![Fig.3.13. An illustration to a metrics selection problem](image)
3.7.3. An optimal discretization step selection

The discretization step of line normal parameters play a primary role in detection algorithm characteristics, and first of all – to its performance and accuracy. So the correct selection of discretization step is a very important task.

The problem of discretization step selection can be described as follows. The larger the discretization step, the lesser an accuracy of normal parameter
determination, and several different line segments will be detected as one line. On the other hand, the lesser discretezation step, the larger computation time, but after some level the accuracy grows insignificantly. The small discretezation step causes a line splitting effect: one line splits into a several ones. The danger is that if the splits are too short they all will be skipped and whole line will be missed regardless of its size.

One way to estimate the minimal discretezation step is to examine the normal parameter space in rectangle co-ordinate system instead of polar system:

\[ X_0 = \rho \cos \phi \]
\[ Y_0 = \rho \sin \phi \]

This parameterisation is known as foot-of-the normal scheme [Davies, 1990]. This scheme provides better performance and accuracy, but the accuracy goes down near the (0,0) point. The (0,0) point is a singularity point: it is impossible to determine a line direction if its normal co-ordinates are zero. It is easy to see that (\( \rho = 0, \phi = 0 \)) allows to determine a line, while (\( X_0 = 0, Y_0 = 0 \)) does not.

The normal parameter space in rectangular form will be looks as it shown on Fig.3.15. Due to discrete nature of parameter space, the lines will be placed non-uniformly. They will be located on several rays. The amount of rays depends on the contour detector type. For the used detectors described in chapter 2 (Sobel 3x3 filter and statistical detector) the amount of rays will be 96. The amount of rays is equal to a maximum possible amount of normal angle gradations that provides maximum accuracy. In the case of 96 rays the angle discretezation step will be \( 360^\circ / 96 = 3.75^\circ \).

The experimental result of different angle discretezation step using is shown on Fig.3.16. Image on Fig.3.16,A was used as a source. The minimal discretezation step of \( 3.75^\circ \) causes the loss of several lines due to splitting effect. The discretezation step of \( 7.5^\circ \) is looks as best variant: it allows detecting all significant lines while the splitting effect is minimal. Larger discretezation step \( 15^\circ \) does not produce any new lines and splits existing lines. Larger steps – \( 30^\circ \) and \( 60^\circ \) – cause appearance of false lines. The last case, where discretezation step is a huge value of \( 120^\circ \) is shows that method still works correctly and detects enough amounts of real lines.

For the \( \rho \) parameter representation, the best solution is to select the discretezation step of one pixel. The experiments show that lesser step causes significant line splitting, and some lines will be missed. Larger step value reduces the accuracy without any positive effect.
Fig. 3.15. A parameter space of normal parameters in rectangular coordinate system

Fig. 3.16. Dependence of line amount from the angle discretization step
3.7.4. Minimal object size that can be approximated by line segments

One more important characteristic of the line detector is a minimal size of object, which still can be approximated by a set of line segments. To determine this critical size it is possible to use an artificial test image shown on Fig. 3.17.

Fig. 3.17 contains a set of figures of periodically increasing size – circles and squares of different orientation. Number under the figure row means its size. The image shows that critical size for figures with straight edges is 8 pixels, and edge orientation is not affected to this critical size. In the case of arcs, the critical radius is 11 pixels. In this case the arc image will have straight segments of 8 or more pixels, which can be approximated by a line segments.

Right part of the test image contains a circle of large radius in order to show the algorithm ability to correctly approximate arcs and curves.

Fig. 3.17. Critical size for different objects
A) Source image
B) Presentation provided by a set of line segments

3.8. Conclusion

This chapter describes the Extended HT concept, and the new line segment detection method is developed based on it. The Extended HT allows separating the identification and description parameters of line segments. This allows selecting those parameters, which provide, on the one hand, high reliability and simple computation scheme of the detection, and on the other hand – provide accurate description and determination of additional characteristics with minimal computations (e.g. brightness gradient direction and line pixel density).
Described line detection algorithm has significantly better characteristics than classical approach can provide. It is faster, since it used simple computation scheme, does not require complex indexation of accumulator array, and uses only one pass. Algorithm does not require lot of memory since it uses two-dimensional accumulator array instead of four-dimensional. The parameterisation that uses the line segment end co-ordinates allows reaching one-pixel accuracy, which is unreachable by other methods, and allows using relatively simple and fast line merging algorithms.

In such a way, the developed algorithm is faster, more reliable and accurate. It can work in real time scale and can process real images. The method characteristics make it possible further image registration by matching of line segments, which are extracted by proposed method.

It must be noted that proposed concept of Extended HT could be successfully applied for the detection of other features like circles, arcs, curves, polygons etc. However, the discussion of this task is outside the boundaries of the current research and is the subject of further research.
Chapter 4. IMAGE REGISTRATION

This chapter is devoted to image registration using their representation as a set of line segments. Different methods of registration by line segments will be overviewed and analysed.

4.1. Introduction

This chapter is devoted to image registration using their representation as a set of line segments. Chapter examines both known and novel image registration methods, based on Hough Transformation, investigates their characteristics and compares each other in order to produce recommendations for proper method selection to be used in practical applications.

4.2. Statement of the problem

In the previous chapter an algorithm was described that allows representation of each image \(I_n\) as an appropriate list of line segments \(S_n\). The following information is available for each segment:
- Co-ordinates of ends;
- Direction and the average value of a brightness gradient along the segment points (it is determined during the line segment extraction procedure);
- The normal parameters (they are unambiguously defined via end co-ordinates, however they are usually known as a side result of segment extraction procedure)
- Length of a segment.

So there are two images presented as a set of line segments. The task is to estimate parameters \(p\) of transformation \(L(p)\) that passes one image \(S_1\) into another \(S_2\) and produces maximum coinciding of both sets, i.e. maximises some quality functional \(\max J(p) = J(L(p,S_1),S_2)\).

A set of possible transformations \(L(p)\) will be limited by simple translation \((\Delta x, \Delta y)\), or – where it will be marked specially – it can contains also rotation \(\phi\) and scaling \(m\).

The quality functional \(J\) will be either amount of line segments or amount of pixels in all line segments that will match on both images when transformation \(L(p)\) is implemented. First variant is less accurate, but allow reducing the amount of computations in several methods.
4.3. Image registration strategies

As it was reviewed in chapter 2, there are three slightly different image registration strategies – exhaustive search, graph matching and Hough transformation.

Exhaustive search and its variations are usually used in area-based methods, where a number of elements to be compared are greater than amount of possible positions. In feature-based methods, an exhaustive search produces a lot of void passes and thus requires too many computations.

Most popular strategies of feature-based image registration methods are based on graph matching. Their main disadvantage is significant computation accuracy, but they require a lot of computations. This makes them difficult to implement in real time applications.

Alternative to a graph approach is strategy based on Hough transformation. This approach has promising characteristics (high performance, in first of all). However, it is poorly investigated, and known implementations have low reliability. So HT strategy is rarely used in a practice.

Here a short description of these strategies.

4.3.1. Exhaustive search

For each possible position of the current frame on the base frame the amount of matching elements are calculated. The maximum amount corresponds to a most probable position of the current frame on the base frame.

A disadvantage of this strategy is a huge amount of calculations, because it process large amount of position that is proportional to image area in pixels. The amount of operations is proportional to amount of all possible positions of one image on another. So it can be used only in a limited set of applications where the size and dimension of search space are small. This strategy is well investigated and approaches to reduce the size of searching parameter space are known (first of all – hierarchical, or pyramidal search). However, significant reducing of computations is still unreachable, so other strategies must be used.

Another approach is to compare lines instead of checking for all possible positions. Two variants how to do this are known – graph-to-graph mapping and HT for the point features. Since the small amount of lines (or other elements) is used instead of large amount of positions, these strategies are much faster.

4.3.2. Graph matching

This approach can be described as follows. Each pair of features that probably can correspond to each other produces a space transformation, which transforms one feature to another. So it is required to find a transformation that
leads to matching of maximum amount of features. A systematic way to represent this is to construct a match graph, in which the nodes represent feature assignments, and the arcs joining nodes represent pair-wise compatibility between assignments. To find the best match it is necessary to find regions of the match graph where the cross-linkages are maximum.

A complete sub graph, where arcs connect all pairs of nodes, is called clique. Hence, the maximum cliques are taken as they lead to the most reliable matches between the sample and observed image.

The main problem of matching graph approach is an exponential growth of required computations in dependence on feature amount. A quantity of all possible feature combinations that define a search tree depends exponentially on amount of structural elements. The task of looking for sub graph in a graph belongs to NP-complete tasks class (like a travelling salesman task). For such tasks, there is no known means of ensuring that they are executed in polynomial time. Thus, given a graph of n nodes, it is not known how to find the maximum cliques in a time that is bounded by a polynomial of n, current indications being that it is at least exponential in n. For example, to process a graph of n nodes, it is required to check about $2^n$ nodes in it. So the exhaustive search is acceptable only in a limited group of tasks. More popular searching methods are the stochastic methods or finite searching methods like dynamical programming, annealing methods, genetic algorithms etc. However, they are not guaranteed that the best image fitting (i.e. real position) would be found.

This strategy and its variations became especially popular in the last time, so its description it is not included in the current research.

4.3.3. Hough Transform strategy

The basic idea of HT strategy, or clustering, for the case of point-like features was proposed in [Davies, 1992], and can be briefly described as follows.

There is two images presented as sets of point features (specific points, corners, line crossings, circle centres etc.), and it is required to estimate translation of one image in relation to another. In this case, the search space has two dimensions, and can be represented by two-dimensional array of accumulator cells. Each pair of corresponding features on one and another $r$ images produces uniformly defined translation vector ($\Delta X=X_2-X_1, \Delta Y=Y_2-Y_1$) that can be marked in search space by increasing the accumulator cell $A[\Delta X, \Delta Y]$ value. Different feature pairs can produce different translation vectors (since inappropriate pairs of features are also taken into account), but commonly these elementary translations will be grouped in a tight cluster around the real translation value. So this cluster can be easily located and its centre can be considered as a desired translation value.

Unlike the graph approach, the strategy requires polynomial amount of computation, and it is significantly faster. Amount of computations is proportional to $n^2k$, where $n$ is amount of features in image representation and $k$ is number of features required to produce point feature. In the case when image represented as a
set of point features, \( k \) is one. If the lines are used, \( k \) will be 2 for pure translation, 3 for translation and rotation, and so on.

Generic disadvantage of HT-based strategies is significant amount of memory required to hold accumulator array. So direct use of HT is possible only for the case of translation. In the case when complex transformations are present, different methods of a search space reducing can be used. It can be, for example, multi-resolution methods like pyramid approach, or hashing of accumulator array. They can reduce memory requirements, however, they are also require many additional computations.

4.4. Hough Transform strategy variations

Basic HT strategy can be easily expanded for the case of non-point features, or case when image deformation differs from simple translation. There are several ways to do this. In this chapter a following extensions will be taken into account.

4.4.1. Point features formed from non-point ones

This approach considers forming a multi-dimensional composite point features from basic non-point features in order to use them according the basic algorithm. If the lines used as an image representation features, point features can be formed as a line combination – line pair crossing. In the case of line segments two more point features became available – corners and line segment middles (line segment ends cannot be used as point features due to low detection accuracy).

It must be also noted that the point feature can be selected in such a way that it will have no clear geometric interpretation. For example, any line pair will produce point feature with three co-ordinates (for example – crossing point co-ordinates and angle between lines), three lines produce four-dimensional object, and so on. Furthermore, these lines can belong to different images. So different line combinations will produce different registration strategies with different characteristics. Theoretically, it is possible to extend this approach for the any number of deformation parameters. However, the amount of objects that can be formed by \( n \) lines from the set of \( N \) is \( N^n \), and image registration computational complexity became \( O(N^{2n}) \). This is too large for a many real-time applications, and other strategies must be kept in mind.
4.4.2. The search space decomposition

This approach considers reducing the search space dimensionality by its decomposition. The example of this strategy is our stripe comparison method described further. This is a fastest possible strategy, since search space automatically decomposed into a set of one-dimensional subspaces. However, the known disadvantage of search space decomposition is significant reducing reliability, so the examination of this direction was limited by only one method – stripes comparison.

4.4.3. Standard Hough Transformation approach

This approach considers direct usage of non-point features that will produce non-point responses in a search space. Since it looks like the Standard Hough Transformation (SHT) scheme for feature detection (not for image registration itself), it will be referred as SHT-strategy.

The idea of this strategy is similar to a HT. Each pair of corresponding elements on the both images produce a whole curve in the parameter space instead of one response, as it was in previous case. The interception point of curves means the possible position of one image on another. The interception point that belongs to most of curves is a most probable coordinates of image position on another image, and it corresponds to an accumulator cell with maximum value.

If the parameter space has more than two coordinates, each pair of corresponding elements produce a surface (or hyper-surface), and amount of calculations also increases, but the amount of computations remains linear function of elements amount.

In comparison to previous case, this strategy requires only $n^2$ computations, and it must be faster for the larger images with larger $n$ values.

The known disadvantage of this strategy is a low accuracy due to a low accuracy in the curve interception point location. Since all calculations performed in the image-like discrete parameter space, specific problems arise. These problems have the same nature as ones described in the previous chapter, and they will be thoroughly analysed.
4.5. Exhaustive search

The simplest solution is based on looking for maximum of quality functional by exhaustive searching in the shift and rotation parameter space, taking into account the limitations related to the aspects of the object dynamics. The algorithm looks as follows:

\[ J^* \leftarrow 0; \]
For each admissible rotation \( \phi \):
  For each admissible horizontal shift \( dx \):
    For each admissible vertical shift \( dy \):
      - Calculate object parameters \( St(\phi, dx, dy) \) of the current frame for rotation \( \phi \) and shift \( (dx, dy) \);
      - Calculate a functional of quality \( J(So, St) \);
      - If a calculated functional is greater than current maximum \( J^* \), then store a shift and rotation values \( (\phi^*, dx^*, dy^*) \), and set a calculated value \( J(So, St) \) as current \( J^* \).

It is values \((\phi^*,dx^*,dy^*)\) found which are shift and rotation parameters of the current frame in relation to the basic frame.

The advantage of this method, apart from its obvious simplicity, is a maximum possible accuracy; the methods described below are less accurate, but have greater speed.

The main disadvantage of this method is the significant amount of surplus calculations: in the most points \( J(\phi,dx,dy)=0 \) (no matching objects).

4.6. METHODS OF ENDLESS LINE COMPARISON

4.6.1. Line matching based on Hough transform

It is considered that there is a line \( l_1 \) from the basic frame and an appropriate line \( l_2 \) from the current frame. The passage \( l_1 \rightarrow l_2 \) generates the whole set of possible values of a translation vector \( L \) (Fig.4.1).
Thus, in the space of translation parameters each pair of lines of the basic and current frames will generate some curve $L(p,w) = (p/Cos(w), w)$. The cross point of a maximum amount of such curves will give a required translation value.

Such a method is simple and fast enough and it proves itself in another applications. However, experiments show a curious phenomenon, the idea of which is the following.

Due to the discrete nature of the parameter space, axioms of Euclidean geometry are not applicable to discrete lines. For example, distinct parallel lines can intersect, and nonparallel lines can have no cross point (Fig.4.2). As a result, images with lines of dominant direction will produce maximums at the point of dominant parallel lines “crossing” instead of crossing points of other lines. To solve this problem, another approach of line drawing must be used: each cell, which contains current line, must be marked by a number that is proportional to length of line segment inside the cell. This process is generally called as aliasing. However, the aliasing still does not allow separating neighbour parallel lines, and they still will be crossing in some points.

The problem becomes especially complex when curves are present instead of direct lines, as in the current algorithm. The proper aliasing for curve requires many additional computations, and gives less effect.

However, this problem can be moderated by proper selection of parameters representation. Further when line segment matching will be discussed, the rectangular coordinates for translation parameter representation instead of polar will be used. They transforms the arctangent curves into direct lines, that allow easily find the cross points.
The experiments show that this method could not process correctly even simple images. However, additional research shows that it is possible to change the computation scheme, so that it will produce linear responses in parameter space instead of sinus. We will return this scheme in the line segment matching methods.

4.6.2. Method based on line pair comparison

The key disadvantage of the previous method is the overstating of the quality functional value for the points of parameter space, so the quality functional is calculated incorrectly. To eliminate this problem it has been decided to accumulate only those points of the translation parameter space where real curves crossing exist.

To do this, a whole line set is divided into pairs. Each pair contains a line from one frame and an appropriate line from another frame, if these lines seem to be the same. To compare lines, information about line position, brightness, thickness and gradient direction is used. Each line may be included into several pairs.

When pairs are formed, appropriate translation parameters are calculated for each two pairs and the value calculated is marked on the translation parameter space, as before. A position of maximum value in the parameter space will traditionally be considered as the desired translation parameters.

The scheme of the translation parameter calculation is presented at Fig.4.3.
When line $l_1$ converts into $l'_1$ and $l_2$ into $l'_2$, a point $C$ of $l_1$ and $l_2$ crossing translates into point $D$ of $l'_1$ and $l'_2$ crossing. The $CD$ vector is desired translation $L$.

It is denoted $\Delta \rho = \rho_2 - \rho_1$, $\Delta \phi = \phi_2 - \phi_1$.

$\angle CEA = \Delta \phi$ as angles with perpendicular sides. Appropriately, $\angle CBF = \Delta \phi$. $ADBC$ is a parallelogram, because its appropriate sides are parallel. So its appropriate sides are equal: $AD = CB$ and $AC = DB$.

$\triangle CFB$ is a right-angle triangle, so $AD = CB = \Delta \rho_2 \cos \Delta \phi$; $FB = \Delta \rho_2 \sin \Delta \phi$.

Similarly $DB = AC = \Delta \rho_1 \cos \Delta \phi$; $AE = \Delta \rho_1 \sin \Delta \phi$.

Then $DE = AD - AE = \Delta \rho_2 \cos \Delta \phi - \Delta \rho_1 \sin \Delta \phi$.

$\triangle CED$ – is a right-angle triangle, so $CD^2 = CE^2 + ED^2$; then distance $L$ can be found as follows:

$$ |L| = \sqrt{\Delta \rho_1^2 + \Delta \rho_2^2 - 2 \Delta \rho_1 \Delta \rho_2 \cos \Delta \phi \\ |\sin \Delta \phi|} $$

(4.0)

From the same reason:

$\angle ECD = \arctg(ED/EC) = \arctg(\cos \Delta \phi / \sin \Delta \phi - \Delta \rho_2 / \Delta \rho_1 \sin \Delta \phi)$;

Appropriate angle of inclination can be found as follows:

$$ \arg(L) = \phi_1 + \arctg\left( \frac{\Delta \rho_2}{\Delta \rho_1 \sin \Delta \phi} - \frac{\cos \Delta \phi}{\sin \Delta \phi} \right) $$

(4.0)

Our experiments show that the algorithm processes correctly both artificial and natural images. The main disadvantage is relatively large computation time (maximum processing time of 128x128 256-grayscale image for Duron™750 processor achieves 1.5 seconds – in comparison to 0.3...0.5 sec for other methods that are discussed below). But this is compensated by greater stability. The method is suitable for practical applications due to its reliability.
4.7. SECONDARY FEATURE EXTRACTION METHODS

The basic idea of this approach is to combine lines into more complex objects in order to use them for the registration. A pair of matching objects on both images provides exact information about the transformation, which cannot be obtained from matching line pair. Furthermore, more complex object allows more accurate and reliable identification procedure to separate true matches. This cause reduction of pairs to be taken into account and hence significantly decreasing amount of computations. However, the more complex object, the rare it can be found on the image, so this can cause fault of registration process when images contain little amount of required object.

As a rule, this group of methods allows radical reduction of computation time, but usually leads to less reliability. These methods cannot be used by itself, but it can be used together with more reliable methods in order to increase its performance or accuracy.

4.7.1. Corner comparison

The calculation scheme is the same as on the pair comparison method, but due to changing of comparison order a significant increasing of performance can be achieved.

At the first stage lists of corners of the base and current frame are constructed. A corner is a line segments pair, which crosses under 30°-120° angle and has a pair of neighbour ends. The amount of corners, extracted in such a way is many times less than the appropriate amount of identical line pairs. Since the amount of comparisons (and appropriate computation time) depends on the squared amount of objects compared, this leads to radical increasing of performance.

The algorithm is able to process artificial images with high accuracy and speed, but fails on certain aspects of natural images – for example, town landscapes. This kind of images contains a huge amount of little objects (buildings) and curves (streets) so it does not allow enough corners to be extracted for reliable processing.
4.7.2. **Stripe comparison**

The basic idea is the same. Lines are combined into more complex object — stripe, and then the stripes are used for frame comparison. The stripe can be considered as two parallel neighbour lines with opposite directed brightness gradient. The idea of the method is to use stripes of one direction to determine translation in perpendicular direction that can be found exactly. Then this translation can be used to calculate a common translation.

At the first step a set of stripe are extracted. This is a fast procedure because only the gradient direction information is used for each line.

At the second step translations $\Delta \rho_i(\gamma)$ for each stripes set with orientation $\gamma$ are calculated. Then doubtful values are eliminated: only values with quality criteria $J(\Delta \rho_i)$ not less than $C*J_{\text{max}}(\gamma)$ remain, where $C=0.5..1$ is threshold level and $J_{\text{max}}(\gamma)$ is maximum criteria value for stripes with orientation $\gamma$.

At the third step, accumulated values of $\Delta \rho(\gamma)$ are used to calculate translation in according to formula (4.1) and (4.2). As before, amount of appropriate values $L$ of translation are accumulated in the parameter space, and the position of maximum in parameter space considered as the desired translation parameters.

The experiments show significant speed of calculation, but the accuracy was too low: about a half of the natural images were processed incorrectly. The situation becomes even worse when small geometric distortions (e.g. rotation) are present. Due to line extraction features, a stripe width will depend on the rotation angle, making it impossible to compare stripes exactly.
4.7.3. Comparison of line segment middles

This method uses the middle point position of the appropriate line segments to determine the translation parameters. The pair of line segments from base and current frame allows us to determine the whole translation vector \( L \), not only its projection, which is orthogonal to the appropriate line direction. Therefore it is possible to take this feature into account.

The method works as follows. For each line segment of the current frame an identical segment of the base frame is looked for (they must have approximately equal size and brightness). For each of these pair the translation vector coordinates are calculated as translation of segments middle:

\[
L_x = \frac{x_1 + x_2}{2} - \frac{x_1' + x_2'}{2} \tag{4.0}
\]
\[
L_y = \frac{y_1 + y_2}{2} - \frac{y_1' + y_2'}{2} \tag{4.0}
\]

Then the traditional procedure is used: calculated values \((L_x, L_y)\) are accumulated in the translation parameter space, and position of maximum in this space considered as the desired translation parameters.

The method is faster than the pair comparison due to calculation simplicity (compare formulas (4.1-4.2) and (4.3-4.4)). The experiments show that the method has both high performance and adequate accuracy. However, a significant drawback for this method was found. The maximum value is extremely low and equal to 2 in many cases. This feature reduces the method reliability. Although most of the images were processed correctly, this effect can cause a serious problem. To reduce such a negative effect, a slide frame was used instead of pure maximum: maximum position is the position of slide frame, when sum of elements
inside the frame is maximal. This increases the reliably, but in any case an additional investigation is required. Nevertheless, this simple method shows that invoking a line segment end co-ordinates in an image registration process allows obtaining significant advantages in comparison to endless line usage.

4.8. METHODS BASED ON LINE SEGMENT COMPARISON

Line segments, in comparison to lines, provide additional information – co-ordinates of ends – that can be used both for more exact selection of corresponding pairs and for reducing a search space size. This allows significant increasing of performance, reliability and even accuracy.

4.8.1. Image registration by line segment pairs

This method will be further referred as Line Segment Pair Comparison (LSPC). The idea of this method is the same as in line pair comparison method: two pair of corresponding line segments on both images produces a pair of reference points and provides information about projection of translation vector on two directions – direction which orthogonal to corresponding pair direction. If these directions are not parallel, it is possible to reconstruct a whole translation vector.

When translation vector for the current pair is found, an appropriate counter in parameter space is increased. When all pairs are processed, the indexes of counter with maximum value are most probable co-ordinates of real translation.

In such a way, the algorithm of image registration will look as follows:

1. Build a list of possible line segment pairs: each pair consists of line segment on the first image and another line segment on second image that probably corresponds to it (each line segment can correspond to several line segments on another).
2. For each couple of non-parallel line segment pair:
   - Calculate the translation vector co-ordinates,
   - Mark it in the search space.
3. Look for a counter in search space that hold a maximum value. The indexes of this counter are the estimate of the real translation parameters.
4.8.1.1. Looking for a line pairs

The following notation will be used. Objects of the first image will be marked by an asterisk «*» symbol. Values that relate to first line pair will be marked by one prime, to second – by two primes. And the lower index will be used to mark one of the line segment ends (1 for left and 2 for right). Lower zero indexes will be used for line normal parameters. A delta (Δ) will mark a difference in co-ordinates of line segment ends, or, in application to normal parameters – difference in appropriate normal parameters of one line segment pair.

The line segments of one pair must obey following requirements:

1. Requirement of equal direction of the brightness gradient: γ ≈ γ. This condition means also an equal orientation of line segments.
2. Requirement of length equality: \((x_2^* - x_1^*)^2 + (y_2^* - y_1^*)^2 \approx (x_2 - x_1)^2 + (y_2 - y_1)^2\).
3. Requirement of pixel amount equality: \(m^* \approx m\). Amount of edge pixels in line segment is determined on the line segment extraction stage.
4. Limitations to a translation value. These limitations are determined from priory knowledge about task. For example if it is known that the first image is contained into another, the translation value can be positive only and limited by differences in image sizes. In this case lines of one pair must obey the following conditions: \(x_1^* + x_2^* \leq x_1 + x_2\), \(y_1^* + y_2^* \leq y_1 + y_2\)

4.8.1.2. Matching pairs

Two line segment pair suitable for the computation of translation vector must obey the following conditions:

1. **Lines must have a cross point.** Two parallel pairs cannot provide information about translation component that is orthogonal to line direction, and such pairs must be skipped. Furthermore, if the angle between lines is not a zero, but it is small, the accuracy of translation vector computation will be low. In such a way it would be better to take into account only pairs with angle between lines in interval of \(30^0...150^0\).

2. **Bindness condition.** The correspondence of line positions must remain the same for both images. The simplest way to check this condition is to use central points of line segments: \(\Delta x_{2^*} + \Delta x_{1^*} \approx \Delta x_{2^*}'' + \Delta x_{1^*}''\), \(\Delta y_{2^*} + \Delta y_{1^*} \approx \Delta y_{2^*}'' + \Delta y_{1^*}''\), where \(\Delta y\) and \(\Delta x\) are differences of appropriate line segment end co-ordinates. The lines passing through line segments must coincide when translation is performed. To provide a line segment overlapping that is not less than a half of its length, it is required that the centre of lesser segment pass in position between ends of greater segment:

\[
X_1 - (X_2^* + X_1^*)/2 > S_x > X_2 - (X_2^* + X_1^*)/2,
\]
\[
Y_1 - (Y_2^* + Y_1^*)/2 > S_y > Y_2 - (Y_2^* + Y_1^*)/2,
\]

where \((S_x, S_y)\) are the translation co-ordinates.
4.8.1.3. The translation value computation

As in the case of line pair comparison, there is a two line couples \((l^*, l')\) and \((l^{*''}, l'')\). When translated, line \(l^*\) goes to \(l^*'\), and \(l^{*''}\) goes to \(l^{*'''}\). Then line crossing point \(C\) goes to point \(D\), and translation value is \(CD\) (Fig.4.6).

If the co-ordinate system origin is placed into point \(C\), then translation vector is determined by point \(D\) co-ordinates, i.e. crossing of lines \(l'\) and \(l''\) in the new co-ordinate system. Equations of these lines in the new co-ordinate system will be look like that:

\[
l' : \Delta \rho' = x \cos \phi' + y \cos \phi',
\]
\[
l'' : \Delta \rho'' = x \cos \phi'' + y \cos \phi'',
\]

where \(\Delta \rho = \rho - \rho^*\) is the difference of normal length, \(\phi\) – normal inclination angle (since appropriate lines of two frames are parallel a priory, both lines has equal \(\phi\)).

![Diagram](image)

**Fig.4.6. Translation parameter determination using two line pairs**

If \(\Delta \rho' \neq 0\) and \(\Delta \rho'' \neq 0\), both line equations can be multiplied by \(\Delta \rho'\) and \(\Delta \rho''\) appropriately:

\[
l' : \Delta \rho'^2 = x \Delta \rho'\cos \phi' + y \Delta \rho'\cos \phi',
\]
\[
l'' : \Delta \rho''^2 = x \Delta \rho''\cos \phi'' + y \Delta \rho''\cos \phi'',
\]

Since \(\Delta \rho\cos \phi = \Delta X_0\), \(\Delta \rho\cos \phi = \Delta Y_0\), \(\Delta \rho = \Delta X_0\cos \phi + \Delta Y_0\cos \phi\), where \(\Delta X_0 = X_0 - X_0^*\), \(\Delta Y_0 = Y_0 - Y_0^*\) – appropriate foot of normal coordinate \((X_0, Y_0)\) differences, a follow system can be obtained for the line crossing point \(D(x_0, y_0)\):

\[
\begin{cases}
\Delta X_0^2 + \Delta Y_0^2 = x_0 \Delta X_0' + y_0 \Delta Y_0' \\
\Delta X_0''^2 + \Delta Y_0''^2 = x_0 \Delta X_0'' + y_0 \Delta Y_0''
\end{cases}
\]

The solution of this system will be:
\[
x_u = \frac{\Delta Y_0''(\Delta X''_0^2 + \Delta Y''_0^2) - \Delta Y_0'(\Delta X''_0^2 + \Delta Y''_0^2)}{\Delta X_0'\Delta Y_0'' - \Delta X''_0 \Delta Y'_0}
\]
\[
y_u = \frac{\Delta X_0'(\Delta X''_0^2 + \Delta Y''_0^2) - \Delta X_0''(\Delta X''_0^2 + \Delta Y''_0^2)}{\Delta X_0'\Delta Y_0'' - \Delta X''_0 \Delta Y'_0}
\]  

(4.0)

This is a desired translation co-ordinate that obeys the conditions \(\Delta \rho' \neq 0\) and \(\Delta \rho''' \neq 0\). Denominators of (4.6) and (4.7) becomes zero either lines \(l'\) and \(l''\) are parallel, or one of \(\Delta \rho\) is zero.

In order to raise computation accuracy, these formulas can be rewritten via line ends co-ordinates instead of normal parameters. The normal co-ordinates can be expressed from line ends co-ordinates by the following way.

For each line segment, a scalar multiplication of normal vector and \((\Delta X, \Delta Y)\) vector must be equal to zero since these vectors are orthogonal. Since the end of normal \((X_0, Y_0)\) belongs to line segment, then for each line segment points – for example, for the left end \((X_1, Y_1)\) – the scalar multiplication of normal vector to \((X_0 - X_1, Y_0 - Y_1)\) vector must be zero also (Fig.4.7). In such a way, the equation system for normal co-ordinates \((X_0, Y_0)\) is obtained:

\[
\begin{cases}
\Delta X \cdot X_0 + \Delta Y \cdot Y_0 = 0 \\
\Delta Y \cdot X_0 - \Delta X \cdot Y_0 = X_1 \Delta Y - X_1 \Delta X
\end{cases}
\]

The solution is:

\[
\begin{align*}
X_0 &= \Delta Y \cdot \nu \\
Y_0 &= -\Delta X \cdot \nu
\end{align*}
\]

(4.0)

where \(\nu = \frac{X_1 \Delta Y - Y_1 \Delta X}{\Delta X^2 + \Delta Y^2}\)

Fig.4.7. Scheme of normal parameter calculation using line segment end co-ordinates

If line segments inside the pair is supposed to be exactly parallel. Then \(\Delta X^* = C \Delta X, \Delta Y^* = C \Delta Y\), where \(C\) is non-zero coefficient. Then:
\[\Delta X_0 = X_0 - X_0* = \Delta Y \cdot \nu - C \Delta Y \cdot \nu* = \Delta Y \left( \frac{X_1 \Delta Y - Y_1 \Delta X}{\Delta X^2 + \Delta Y^2} - C \frac{X_1* C \Delta Y - Y_1* C \Delta X}{(\Delta X^2 + \Delta Y^2)C^2} \right) =\]

\[= \Delta Y \frac{(X_1 - X_1*) \Delta Y - (Y_1 - Y_1*) \Delta X}{\Delta X^2 + \Delta Y^2}\]

\[\Delta Y_0 = Y_0 - Y_0* = -\Delta X \cdot \nu + C \Delta X \cdot \nu* = -\Delta X \left( \frac{X_1 \Delta Y - Y_1 \Delta X}{\Delta X^2 + \Delta Y^2} - C \frac{X_1* C \Delta Y - Y_1* C \Delta X}{(\Delta X^2 + \Delta Y^2)C^2} \right) =\]

\[= -\Delta X \frac{(X_1 - X_1*) \Delta Y - (Y_1 - Y_1*) \Delta X}{\Delta X^2 + \Delta Y^2}\]

Denote: \(\Delta \nu = \frac{(X_1 - X_1*) \Delta Y - (Y_1 - Y_1*) \Delta X}{\Delta X^2 + \Delta Y^2}\)

Then:

\[\Delta X_0 = \Delta Y \Delta \nu,\]
\[\Delta Y_0 = -\Delta X \Delta \nu\]

If the formulas (4.8) and (4.9) are inserted in (4.6) and (4.7), the following formulae for the translation vector would be obtained:

\[x_\nu = \frac{\Delta X' \lambda' - \Delta X' \lambda''}{\Delta X'' \lambda' - \Delta X' \lambda''}\]
\[y_\nu = \frac{\Delta Y' \lambda' - \Delta Y' \lambda''}{\Delta Y'' \lambda' - \Delta Y' \lambda''}\]

(4.0),

where \(\lambda = (X_1-X_1*)\Delta Y-(Y_1-Y_1*)\Delta X\) appropriately for the first and second pair.

The formulas (4.10) and (4.11) depends on line segment ends only. The limitations \(\Delta \rho' \neq 0, \Delta \rho'' \neq 0\) are eliminated, and formulas can be used for the line segments of any position. Other limitation – a line from different pairs must be non-parallel (if lines are parallel, the denominator becomes zero) – is processed on the previous stage.

4.8.1.4. Method characteristics

In common, the method provides relatively high performance and accuracy. It is also much more reliable than methods described above. The method drawback is that the amount of computations grows as a fourth degree of line segments amount for pure translation, sixth – for translation with rotation and eighth for a whole isometric group. When amount of lines is relatively small (about 100 per image), the performance is high enough. But it can be a serious problem when larger line amounts are processed or when additional transformation parameters must be found.

To avoid these problems, a next method was developed. It requires amount of operation growing only as a second order of line amount.
4.8.2. Image registration by line segments

This method will be further referred as Line Segment Comparison (LSC). The idea of this method can be described as follows. For each pair of probably matching line segments, for each possible transformation, an amount of line segment pixels are marked, which will match for this transformation. When all possible pairs and transformations are examined, the parameter space will be filled by amount of line pixels, which will match if the transformation with current parameters would be performed. The parameters that provide a maximum amount of matching pixels, as before, are the most probable parameters of real transformation.

It is easy to note that most of transformations cannot provide common pixels of two line segments, and only remaining transformations must be actually taken into account.

In such a way, for each pair of line segments (one – from first image and another – on the second) following actions must be performed:

- Determine, whether these line segments can coincide, taking into account the limitations to the search space.
- Determine the set of transformation parameters that can cause the line segment matching.
- Determine the amount of coinciding pixels of line segments for transformation parameters found, and mark them in the search space.

Let us examine these actions more thoroughly.

The possibility of line segment matching is determined as it is described for previous method.

4.8.2.1. Determination of transformation parameters

It is easy to found that all translation parameters that provide line segment matching are situated on the line segment, which connects the AD and BC vectors (Fig.4.8). On the B⇒C and A⇒D translations the only one pixel pair will match (B=C or A=D appropriately). For other translations, belonging to this line segment, the number of matching pixels will be greater.

In such a way, the task of translation parameters calculations can be reduced to a building of BC – AD line segment in search space.
4.8.2.2. Determination of matching pixel amount
Firstly the case AB<CD will be examined. The point A moves from A₁ to A₄=D position (Fig.4.9).

A₁A₂ segment: when A→A₁, the point B goes to point C and it is only one matching pixel. While point A moves to C, the amount of matching pixels grows until reaches a maximum – a size of AB. This happens when point A reaches point C(A₂ position).
A$_2$A$_3$ segment: when point A situated between A$_2$ and A$_3$, whole AB line segment situated inside CD. The amount of matching pixels is maximum and equal to AB size.

A$_3$A$_4$: segment: when point A passes A$_3$ position, point B passes the D and became outside the CD line segment. Amount of matching pixels begins reducing until reaches the minimum – one pixel. It happens when point A goes to D (A$_4$ position), and it is only one pair of matching pixels.

For the case of AB>CD the calculations are appropriately the same, but intervals of pixel amount increasing and decreasing are changed. Amount of pixel grows when point A moves from A$_1$ to A$_2$. At this time point B moves from C to D. On the segment from A$_2$ to A$_3$ amount of pixels remains constant, till point A reaches C. On the segment from A$_3$ to A$_4$ (from C to D) amount of pixels decreases till became one.

In the case AB=CD the (A$_2$,A$_3$) interval is empty. This case can be reviewed as a particular case of any of the variants above.

In such a way, the amount of pixels in search space is a trapezoid with a width equal to size of lesser line segment.

4.8.2.3. Drawing a trapezoid in the search space

The search space can be represented as a rectangular array. The array indexes are the translation parameters, and values are amounts of matching points. The base of trapezoid is a line segment that is build up on the AB and CD vectors. It is obviously impossible to exactly build any desired line in the search space of this type, and approximate methods must be used. The task is to find array cells nearest to desired line segment, and to fill them by values of matching pixels.

There are two approaches for a line segment building on the orthogonal grids: the Brezenham approach and using of independent parameter. The Brezenham approach is faster, but less accurate. But in this case the accuracy will play a primary role. Furthermore, the independent parameter can be also used for computation of pixel amount.

The generation of line points based on independent argument can be described by formulas:

\[
\begin{align*}
x_i &= x_0 + \Delta x \cdot p_i \\
y_i &= y_0 + \Delta y \cdot p_i,
\end{align*}
\]

where \((x_0, y_0)\) is a co-ordinates of start point (point A$_1$ in this case), \(p_i\) is independent argument.

The diapason of argument \(p\) changing is diapason from zero to a maximum difference of line segment end co-ordinates. In this case integer values of argument will generate a line without gaps and overlapping pixels (since one of co-ordinates will be also integer value).

The line segment building can be performed using recurrent formulas:

\[
\begin{align*}
x_{i+1} &= x_i + \Delta x \\
y_{i+1} &= y_i + \Delta y
\end{align*}
\]

The co-ordinates of drawing pixels are rounded to a nearest integer value.
The argument $p$ can be also used for a trapezoid building:

On the $A_1A_2$ segment the following value is added to a segment cells:

$$\frac{|AB|}{\max(|x(A_2) - x(A_1)|, |y(A_2) - y(A_1)|)} \cdot P_i$$

On the $A_2A_3$ segment the constant value $|AB|$ is added to a segment cells.

And on the $A_1A_2$ segment the following value is added to a segment cells:

$$\frac{|AB|}{\max(|x(A_2) - x(A_1)|, |y(A_2) - y(A_1)|)} \cdot (\max(|x(A_4) - x(A_1)|, |y(A_4) - y(A_1)|) - p_i)$$

4.8.2.4. Method characteristics

The method characteristics are appropriately the same as in previous case. In the series of experiments that are described below, the method shows itself as little more accurate, has the same reliability, but little slower.

4.9. Method comparison

Characteristics of all methods described above are placed into a Table 1. The measurements were performed on the following video sequences:

Set 1: A moving rectangle. This is a base test, which is used mainly for debugging purposes. All methods must process it as a necessary minimum.

Set 2: The agricultural landscape (fields and a river), which contains a lot of simple objects and lines. This is a main set.

Set 3: The town landscape (streets and buildings) that contains a lot of tiny details and curve lines that cannot be approximated by line segments. This set is useful to test the algorithm reliability.

It can be seen from the Table 1 that only method of line pair comparison and its assessors – line segment comparison and line segment pair comparison – can be used in practice since they provide enough reliability. All these methods have an average performance and accuracy. These characteristics can be amplified using other methods – for example, by angle comparison.

Since characteristics of two last methods are approximately the same, the additional measurements were performed. The aim is investigate the primary characteristics of these methods: performance, accuracy and reliability in order to compare with each other and with other popular methods.
Table 1. The performance and reliability of tested algorithms

<table>
<thead>
<tr>
<th>Method</th>
<th>Max. time, s</th>
<th>Reliability</th>
<th></th>
<th></th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Set 1</td>
<td>Set 2</td>
<td>Set 3</td>
<td></td>
</tr>
<tr>
<td>Exhaustive search</td>
<td>&gt;10</td>
<td>100%</td>
<td>100%</td>
<td>100%</td>
<td>Maximum exactitude. Unacceptable because of low performance.</td>
</tr>
<tr>
<td>Line comparison</td>
<td>0.3</td>
<td>0%</td>
<td>–</td>
<td>–</td>
<td>Extremely low reliability. Doesn’t work in the pure state.</td>
</tr>
<tr>
<td>Line pairs comparison</td>
<td>1.5</td>
<td>100%</td>
<td>90%</td>
<td>80%</td>
<td>Acceptable ratio of efficiency and reliability.</td>
</tr>
<tr>
<td>Corners comparison</td>
<td>1.2</td>
<td>100%</td>
<td>90%</td>
<td>40%</td>
<td>Doesn’t work for some image types</td>
</tr>
<tr>
<td>Stripes comparison</td>
<td>0.5</td>
<td>100%</td>
<td>70%</td>
<td>20%</td>
<td>Low reliability</td>
</tr>
<tr>
<td>Segment middles comparison</td>
<td>0.5</td>
<td>100%</td>
<td>90%</td>
<td>80%</td>
<td>High performance and accuracy. Low reliability.</td>
</tr>
<tr>
<td>Comparison of line segment pairs</td>
<td>0.6</td>
<td>100%</td>
<td>100%</td>
<td>100%</td>
<td>Acceptable ratio of efficiency and reliability.</td>
</tr>
<tr>
<td>Comparison of line segments</td>
<td>0.9</td>
<td>100%</td>
<td>100%</td>
<td>100%</td>
<td>Acceptable ratio of efficiency and reliability.</td>
</tr>
</tbody>
</table>

4.9.1. Performance

When the LSPC method is used, each line segment pair on first image is compared with pair on another. The amount of computations in a first method is equal to $P_1=O(N_1^2*N_2^2)$, where $N_1$ and $N_2$ are amount of lines on first and second images appropriately.

The LSC method requires each line segment to be compared with each one on another image. The amount of computations in this case is $P_2=O(N_1*N_2)$. This is significantly lesser. However, LSC method requires additional operations to build trapezoids in the search space instead of simple points for LSPC method.

Relation between object amount and method performance is shown on Fig.4.10. It can be seen that both methods provide the same results when amount of lines in frame is small (50 and lesser): higher computation complexity of LSPC method is compensated by trapezoid building in LSC. Furthermore, larger part of computation is related with line detection process. However, the situation significantly changes when amount of lines is above hundred: LSPC method becomes in more than ten times slower than LSC due to its higher computational complexity.

Table 2. Relation between number of objects and performance of the algorithm

<table>
<thead>
<tr>
<th>Number of lines in frame</th>
<th>56</th>
<th>68</th>
<th>80</th>
<th>92</th>
<th>114</th>
<th>136</th>
</tr>
</thead>
<tbody>
<tr>
<td>Segment comparison, ms per frame</td>
<td>40.1</td>
<td>42.2</td>
<td>45.1</td>
<td>49.1</td>
<td>54.1</td>
<td>59.6</td>
</tr>
<tr>
<td>Pair comparison, ms per frame</td>
<td>64.4</td>
<td>92.5</td>
<td>149.7</td>
<td>249</td>
<td>532</td>
<td>846</td>
</tr>
</tbody>
</table>
Fig. 4.10. Relation between object amount and method performance
Let us look more thoroughly for the separate stages of the line-based image registration in order to determine the part of computation they taken in different conditions.

Image registration algorithm with line matching includes several heterogeneous components. The fraction of computation time taken by each stage depends on slightly different factors that indirectly affect one to another.

In common case line-based image registration includes two general stages – line detection and line matching. The amount of computations on the line extraction stage depends firstly on the size of image to be processed. Other important characteristics are amount of edge pixels found and their spatial distribution. They depend on the image type and detection parameters (filtration and thresholding parameters etc.) and can be neither set nor known priory.

The amount of computation on the line matching stage depends mainly on the amount of lines used for matching. The dependence on other factors like image size and type is indirect and sophisticated. This makes a direct research of computation time ratios a hard task.

To solve this task, the following way can be used. We will change the amount of edge pixels on the image, and mark the amount of lines found. The amount of edge pixels affects directly to the duration of the line detection stage, whereas amount of lines affects to the duration of the line matching stage (when all lines are used for matching). Since the dependence of lines found on amount of edge pixels is monotonous, it can be inverted. This allows investigating the dependence of line matching and line detection duration from line amounts by control the amount of edge pixels.

The dependence of line detection and line matching computations from amount of lines found is shown on Fig.4.11. The ratio of computation times in detection and matching stages is shown on Fig.4.12. All time measurements were performed for computer system based on Duron™ 750 processor.

It can be seen that line detection takes insignificant part of computations in LSPC method, and this part quickly decreases with growth of line amounts and image size. This happens since LSPC method has a relatively high computational complexity.

The situation is different in the LSC method. The line detection stage takes larger part of computations. This part slowly decreases with growth of line amount; however, it still prevails.
Fig. 4.11. Computation time of the stages of line-based image registration methods

Fig. 4.12. Fraction of line segment detection stage in common computation time
4.9.2. Accuracy

To investigate the accuracy, the same image was taken as both images, but the line amount was limited. Three characteristics were reviewed: amount of failures (when the position was found incorrectly with error larger than a given limit – seven pixels), maximum error value (failures was not taken into account) and standard deviation. The measurement results are placed into Table 3.

The amount of lines on second image (the map) was constant (217), and amount of lines in a frame was changed from 15 to 70. The amount of frames in one measurement is 23168.

Table 3. Dependence of reliability and accuracy from number of lines

<table>
<thead>
<tr>
<th>Line amount</th>
<th>Max. error $E_{\text{max}}$</th>
<th>Variance $D_x$</th>
<th>Variance $D_y$</th>
<th>Failures amount</th>
<th>Failures, $%$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Line segment pair comparison</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>15</td>
<td>-</td>
<td>0.108</td>
<td>0.358</td>
<td>3550</td>
<td>18%</td>
</tr>
<tr>
<td>20</td>
<td>-</td>
<td>0.0566</td>
<td>0.273</td>
<td>1265</td>
<td>7%</td>
</tr>
<tr>
<td>30</td>
<td>3</td>
<td>0.0414</td>
<td>0.221</td>
<td>76</td>
<td>0%</td>
</tr>
<tr>
<td>35</td>
<td>2</td>
<td>0.0493</td>
<td>0.228</td>
<td>22</td>
<td>0%</td>
</tr>
<tr>
<td>40</td>
<td>2</td>
<td>0.0585</td>
<td>0.237</td>
<td>15</td>
<td>0%</td>
</tr>
<tr>
<td>50</td>
<td>2</td>
<td>0.0687</td>
<td>0.244</td>
<td>15</td>
<td>0%</td>
</tr>
<tr>
<td>70</td>
<td>2</td>
<td>0.0695</td>
<td>0.245</td>
<td>15</td>
<td>0%</td>
</tr>
<tr>
<td>Line segment comparison</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>15</td>
<td>-</td>
<td>0.0496</td>
<td>0.147</td>
<td>2090</td>
<td>11%</td>
</tr>
<tr>
<td>20</td>
<td>-</td>
<td>0.0174</td>
<td>0.0579</td>
<td>696</td>
<td>4%</td>
</tr>
<tr>
<td>30</td>
<td>3</td>
<td>0.00468</td>
<td>0.0256</td>
<td>24</td>
<td>0%</td>
</tr>
<tr>
<td>35</td>
<td>4</td>
<td>0.00291</td>
<td>0.0172</td>
<td>2</td>
<td>0%</td>
</tr>
<tr>
<td>40</td>
<td>1</td>
<td>0.00161</td>
<td>0.00853</td>
<td>2</td>
<td>0%</td>
</tr>
<tr>
<td>50</td>
<td>1</td>
<td>0.000573</td>
<td>0.00271</td>
<td>2</td>
<td>0%</td>
</tr>
<tr>
<td>70</td>
<td>1</td>
<td>0.000364</td>
<td>0.00224</td>
<td>2</td>
<td>0%</td>
</tr>
</tbody>
</table>
Fig. 4.13. Comparison of accuracy and reliability

A) Dependence of faults from line amount;
B) Dependence of the error distribution variance from the line amount.

It can be seen that line segment comparison method has a lesser deviation (in 2 and more times), lesser maximum error (1 pixel instead of 2) and lesser amount of faults (in 1,5 and more times).

The interesting fact that the line pairs comparison method has a higher accuracy if the line amount is limited by about 30 lines per frame (Fig.4.13,B). However, the reliability is monotonous function, and it is higher for the larger line amounts (Fig.4.13,A)

4.9.3. Reliability

To compare the method reliability, another group of experiments has been done. The source image was rotated and scaled, divided into 128x128 pixel frames, and frames position was looking for on the original image. The angle and scale correction was disabled for both methods, and they could operate by translations only. The measurement results are shown on Table 4.

It can be seen that although line segment comparison method shows higher reliability on original and turned images, it appears less reliable on scaling. The accuracy of line segment comparison methods appears higher in all cases.
Table 4. Comparison of reliability and accuracy of tested algorithms in the presence of geometric distortions

<table>
<thead>
<tr>
<th></th>
<th>Line segment pair comparison</th>
<th>Line segment pair comparison</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$D_x$ $D_y$ $E_{\text{max}}$</td>
<td>$N$ $D_x$ $D_y$ $E_{\text{max}}$</td>
</tr>
<tr>
<td>1. Original image</td>
<td>0.0318 0.116 2</td>
<td>5 ($&lt;$1%)</td>
</tr>
<tr>
<td>2. Rotation on 1°</td>
<td>0.757 1.73 5</td>
<td>291 (1%)</td>
</tr>
<tr>
<td>3. Rotation on 5°</td>
<td>3.94 4.14 7</td>
<td>5040 (26%)</td>
</tr>
<tr>
<td>4. Scaling 90%</td>
<td>31.4 38.3 6</td>
<td>1380 (7%)</td>
</tr>
<tr>
<td>5. Scaling 110%</td>
<td>22.5 32.2 11</td>
<td>5291 (27%)</td>
</tr>
<tr>
<td>6. Scaling 125%</td>
<td>- - - -</td>
<td>- -</td>
</tr>
</tbody>
</table>

4.10. Conclusion

In this chapter possible methods of image registration using presentation in a form of line set are examined. This method in common has a highest performance, on one or two degrees higher than other methods allow. To this moment, the method accuracy and reliability was poorly investigated. This chapter proposes two methods – line segments comparison and line segment pair comparison – which have a high ratio of performance, accuracy and reliability. This allows using these methods in a wide range of practical applications.

Another interesting fact is that the reliable method – line segments comparison – can have computation complicity of $O(n^2)$. This looks like a theoretically achievable limit. All existing approaches known to this moment provide larger computation complexity [Brown, 1992].

Nevertheless, all reviewed methods have a disadvantage: they require that image must contain direct lines. There are number of important situation when direct lines are absent on the image. In this case, a line comparison algorithm must be combined with another methods, for example – area-based correlation.
CHAPTER 5. DISCUSSION

The method of image registration by line comparison, which was proposed and analysed in previous chapter, has significant advantage in comparison with known image registration techniques: it requires significantly lesser amount of computations. This effect is even more sensitive when larger images are processed. This happens because the computational complexity of the algorithm is $O(n^2)$ while known methods requires more computations (e.g. dynamical programming provide only $O(n^3)$). So the method can be considered as a good alternative to be used in real time systems.

However, the method has its own disadvantages and unsolved problems, which is either specific for the whole class of feature-based methods, for the HT approach or for proposed method itself.

Class-specific problems are associated with reduced reliability and accuracy of feature-based image registration in comparison to area-based exhaustive search methods. Some problems arise on the feature detection stage, and also related with feature extraction accuracy and reliability for the low quality images. Another important problem is significant memory requirements to keep the results of HT.

This chapter provides comparison of proposed method with known ones in order to determine the area of application. Proposed method can be divided into two important parts – image registration itself and feature detection. Since an original algorithm is used, it is also necessary to compare its characteristics with characteristics of known algorithms. It is important because quality of line detection plays key role in a whole image registration algorithm, and more sensitive to a changing of image type.

5.1. Characteristics of image registration methods

5.1.1. Performance considerations

Amount of computations in the proposed algorithm of line-based image registration depends on several parameters such as the image size, the amount of edge pixels found, the amount of line segments found, the amount of line segments selected for matching, the ability of priory vectorisation of one image and others. These parameters affect to the amount of computations in separate stages of whole algorithm and to the ratio of these computations.

The proposed algorithm contains three large stages: pre-filtration, line segment extraction and line matching.
**Pre-filtration**

The pre-filtration stage is necessary for the images of low quality, which is not enough to reliable and accurate vectorisation and further image registration. Pre-filtration includes noise suppression, brightness and contrast balancing. In some cases more sophisticated operations are required, for example – partial image restoration. Current research does not analyse the pre-filtration stage since it is well investigated, and images used have enough quality to skip the pre-filtration. However, practical applications often require it, and appropriate computations must be included into total amount of computations. The main parameter affected to the computational time is the image size, i.e. total amount of pixels. The amount of computations for the typical approaches is well known and can be easily estimated. One of the most popular solutions to implement the pre-filtration is mask filter. Their computational complexity linearly depends on amount of image pixels, and it is usually required about 10% of whole vectorisation time. However, in the several cases more expensive filtration is required.

**Line segment extraction**

The amount of computations on this stage depends on several parameters, such as image size, amount of edge pixels, amount of lines found. The image size is primary parameter since it is determined by developer and does not depend on the image content. Other parameters depend both on image type and on tuning parameters like thresholding level and minimal line size. As it can be concluded from Chapters 2 and 3, amount of computations on this stage linearly depends on the image size. The dependence of computations from other parameters can be also easily obtained but it is useless due to the difficulty of the parameter estimation.

**Line matching**

As it was shown in the previous chapter, the computational complexity of proposed line matching algorithm is proportional to n1*n2, where n1 and n2 are line amounts on both images to be compared. Theoretically, it is enough to know only two lines on each image. However, additional influences make this rule incorrect. For example, line on one image can be absent on another, since it can be situated outside the visual field, can be noised, distorted or overlapped by something. Furthermore, image can contain several lines that look simultaneously, which makes uniform matching impossible. Another case is the curve approximated by direct lines: there are several ways to do this, depending of detalisation level and even curve orientation on the digital image. So in a practice the more lines are used the better results will be obtained. The more lines in the image description, the better chances to find matching line pairs.

On the other hand, increasing of line amount lead to increasing amount of computations, so only limited amount of lines can be used. Amount of lines in the image description can play a role of regulator to set balance between reliability and performance, and can be considered as a tuning parameter.
As it was showed in Chapter 4, the amount of lines involved in registration process determine the ratio of computation times in line extraction and line matching processes. When amount of lines is relatively small (100 or lesser), the larger part of computation is spent to a line extraction. This amount of computation is proportional to amount of pixel on images. The situation became simple if one image can be converted in a set of lines in off-line mode, and vectorisation of only one image is required in real time. This can be, for example, when one of image is a priori known map in visual navigation tasks, or when there is a set of a priory known samples in image recognition tasks. Furthermore, in these tasks a larger set of images can be vectorized in off-line mode.

In such a way, proposed algorithm of image registration contains three components – pre-filtration, vectorisation and line matching. Amount of computations in pre-filtration and vectorisation stage depends linearly on image pixel amount, whenever amount of computations in line matching is proportional to multiplication of line amounts on both images.

Existing area-based algorithms of image registration in common case require the amount of computations that is proportional to \( p_1^2 p_2^2 \), where \( p_1 \) and \( p_2 \) – amounts of pixels on first and second images. Using the pyramidal scheme can reduce this amount to \( p_1 p_2 \log(p_1) \log(p_2) \), however, it is critical to the image quality. Using of correlation via fast Fourier transformation gives the same result, but it has its own specific problems.

Best of feature-based graph matching methods requires amount of computation that is proportional to third order of line amount. They are also requires the same computations for vectorisation and pre-filtration.

In such a way, the proposed algorithm of line matching has significant advantages in performance when dealing with images of large area and high detalization (large amount of lines, required for correct image description). This can be, for example, in a task of image position determination on the map in the visual navigation task, when relatively small video frame is compared with a large map, which is vectorized in off-line mode.

When the image size is relatively small, the advantages in performance are reduced significantly due to overwhelming of vectorisation computations over line matching, and classical methods became more preferable here.

### 5.1.2. Memory requirements

Primary disadvantage of HT-based image registration methods is significant memory requirements. Memory is required to keep accumulator array, which represents the search space. The size of accumulator array is proportional to a number of all possible positions of one image on another – i.e. amount of all
possible combinations of transformation parameters with required discretization step.

When the number of parameters to be determined is small, the memory requirements are acceptable. For example, it is often required to determine the pure translation value only. In this case the accumulator array size is approximately equal to a larger image. So if one of images is a map of 10,000x10,000 pixel size, the accumulator array will require about 200Mb (if each cell is represented by two bytes) to achieve one-pixel accuracy.

However, situation changes dramatically when additional parameters are added. These parameters are usually rotation angle and scale. For example, if it is required to determine the rotation with one degree accuracy, the memory requirements will grow in 360 times (about 72Gb for the previous example). If it is also required to determine scale in the typical small range ±25% with a step of 1%, the memory requirements will be increased in 50 more times – about 3.6 Tb. It is clear that such a huge value is unacceptable not to on-board computers but even most powerful of existing machines.

The search space is significantly rarefied, so traditional methods can be used to reduce memory requirements.

One of the most popular techniques is hashing with losses. It allows keeping huge virtual array in a significantly lesser physical memory by ignoring void cells and insignificant responses. But the price of hashing is significant growth of computation to perform an array addressing operation. The more complex algorithm is used and more the memory compression level, the more additional computation will be required.

Another approach is iterative search with reducing the search area and increasing accuracy on each step. Since algorithm begins from rough discretization step, the accumulator array size can be significantly reduced. However, the specifics of HT-based strategies is that the reducing of the search area or parameter determination accuracy does not lead to significant reducing of required computations (unlike to area-based methods). So amount of computations is growing proportionally to amount of iterations.

Popular alternative to a hashing and iterative search is search space decomposition. This method decomposes the search space to a set of subspaces of lesser dimensionality. Instead of marking each response in a search space, now it is needed to mark its projection to appropriate subspaces. The traditional solution is to use subspace of pure translation and rotation-scaling subspace. In this case the amount of memory in the example above will be only 200Mb + 72Kb. So the decomposition provides a most effective memory usage. However, our experiments show that using of decomposed search space leads to significant reduction of reliability. This happens because different cluster situated far one from another in initial search space can be projected into a same area in subspaces, and will produce false responses. The effect will be greater for a lesser subspace dimensionality. So this approach cannot be recommended for use in practical applications.
In such a way, significant disadvantage of HT-based image registration method is significant memory requirements. The typical approach to lower these requirements can be hashing accumulator array, implementing iterative search and search space decomposition. First two approaches requires significant amount of additional computations, whenever last one lowers the probability that desired position is found correctly. The selection of one or another approach in a specific application will be determined by task conditions and requirements. For instance, the hashing can be best choice for the task of merging separate air photos into complete map. This task will produce a huge search space but is not significantly sensitive to computation amount. Another typical task is image registration for the tasks of autonomous navigation. The most popular and suitable strategy here is hierarchical search since it provides good trade-off between reliability and performance. Many machine vision applications like object localisation and inspection can also use search space decomposition, since the light conditions and background can be amplified in order to provide correct line extraction and image matching.

5.1.3. Accuracy

Modern theory of image registration counts the HT approach as having lesser accuracy (several pixels) than other method can provide. So experiments were done in order to analyse thoroughly an accuracy of proposed approach.

Experiments, described in Chapter 4, demonstrate that accuracy depends mainly on amount of lines involved in matching process. When amount of lines are small (20 pixels and lesser), the standard deviation became greater than one, so the error of several pixel can be obtained. This situation is typical for the object recognition applications, where relatively simple object can be described by small amount of lines. The inaccuracy value depends also on the prevailing line orientation: as a rule, prevailing vertical and horizontal lines provides better accuracy in comparison to lines of other orientation.

The situation significantly changes, when larger amounts of lines are used. In this case the standard deviation becomes lesser 0.01, and accuracy is in the one-pixel range. The prevailing line orientation does not significantly affected to accuracy, because statistical properties of line distribution compensate most of other factors.

There are several ways that can help to achieve even sub-pixel accuracy. It becomes possible since more complex object (lines) is used for matching instead of separate key points. One of approaches that allow obtaining sub-pixel accuracy related with an analysis of cluster shape in the search space. In the simplest case it is enough to calculate a mass centre of the cluster. However, in some cases more sophisticated analysis is required, that takes into account moments of higher level.
The task of obtaining sub-pixel accuracy lies beyond the ranges of this research, and this task is not reviewed here: sub-pixel accuracy can be more easily obtained by some other methods like graph relaxation when approximate position is already estimated by proposed method.

The accuracy that can be achieved is sensitive to another factor – uncompensated geometric distortion of one image to be registered. When such distortions are present, the uniform matching becomes impossible in the terms of line sets. It is impossible to propose such a translation that causes exact line matching on both images. In this case the response distribution in the parameter space becomes significantly different from normal (it have a shape of deformed rings), and cannot be analysed by standard approaches. More complex (and more computationally expensive) methods are required in this case.

So, the proposed method has accuracy in the range of one pixel if no uncompensated image distortion is present.

5.1.4. Reliability

As a rule, feature based method has lesser reliability in comparison to area-based ones – this is a price for its significant speed. The reliability depends on many factors, and most important of them are the follows: changing of brightness and contrast, high spatial frequency noise, geometric distortion and image content.

5.1.4.1. Brightness and contrast variations

The changing of brightness and contrast causes the difference in the brightness value of appropriate pixel of both images, even the images are matched exactly. In in-door scenes it can be compensated by proper selection of light conditions whenever it is impossible to do this in out-door scenes due to daily changes of light conditions. The more complex problems rise in airspace image registration: additional brightness changing is caused by season and weather variations. Furthermore, the colour of local areas can change, so the brightness transformation becomes non-global, and cannot be characterised by only two parameters of brightness and contrast.

Brightness changing is a serious problem for area-based methods, and often it is one of the main reasons to use feature-based method instead. Feature-based methods (and proposed method as well) are practically insensitive to the brightness and contrast changing – neither global nor local – until it allows extracting of appropriate features and does not corrupt image (e.g. like solarization does).

5.1.4.2. High spatial frequency noise

A noise on the image, as a rule, has significantly high-frequency nature. This makes possible to implement noise filtration, since objects on the image have lower frequencies. Noise, as a rule, appears on the image reception and it is
specific for different types of receiving devices. The sample of specific noise can be a speckle noise in infrared images.

The high-frequency noise can make more difficult (or even impossible) the feature extraction, because the feature extraction process requires the edge pixel extraction, and edge pixels itself are the high-frequency object. So when filtration is used, edges can be suppressed or distorted as well: corners become rounded, tiny details are disappeared, lines are shifted from its real position etc.

The using of direct lines in comparison to other possible features (corners, reference points, arcs etc.) is more useful when significant high-frequency noise is present, since line detection is less sensitive to such a noise (lines are more low-frequency objects). In common case, the larger and more complex object is used, the less its extraction sensitive to noise. So, for example, it will be better to use rectangles instead of separate line segments, line segments – better than reference points and arcs, and so on. However, the more complex is used, the rare it will be present on the image completely or even partially. So the line segments seem to be a good compromise between stability to noise and rarity on the image.

If the static filtration with constant parameters is used, the problem of the line shifting has a simple solution: it is enough to implement the same filtration to the both images. This causes the same line shifting on the both images. It is not a problem that lines will be shifted from its real position – it is important that both images will match exactly in their line set representation form.

In such a way, presence of significant high-frequency noise makes more difficult to extract object contours and further line extraction, and use of filtration causes line shifting and reduce both reliability and accuracy of the image registration.

5.1.4.3. Geometric distortions

Geometric distortions are such a spatial transformation, which parameters are not determined in the image registration process. A presence of uncounted geometric distortions causes the images cannot be matched exactly using accounted transformations. For example, when one image is rotated in relation to another, it is impossible to achieve image matching by implementing shifts only.

Like the high-frequency noise, the geometric distortions played insignificant role when area-based methods were used: regardless of geometric distortions and noise, the most of image pixels will have matching pair. The situation changes when feature-based methods are used, since amount of features in the image description is significantly lesser (from 100 to 1000 times) than amount of pixels on it. As it was showed in the experiments in the Chapter 4, presence of even insignificant geometric distortions causes serious reducing of reliability of the image registration process. As a result, all geometric distortions must be accounted as full as it possible.

If the parameters of geometric distortions are known – a priory or obtained from other sources – their compensation is relatively simple. It is enough to implement needed transformations over the vector description of the image. For example, when the image is presented as a set of line segments, the
transformations must be applied to line segment end co-ordinates. The amount of line segments significantly small (in comparison to overall amount of pixels that would be corrected if the area-based methods are used instead), so amount of computations required for performing a correction is also small (it takes lesser than 1% of total computation time).

Such a correction can be implemented, for example, to a projective distortions in the case when camera is looks to a scene under an angle or when scene is not flat, or when non-linear distortions caused by wide-angle camera usage are present. Correction can be also implemented to rotations and scaling when their parameters are estimated from other sources.

The problems arise when information about distortions is not available. In this case the task complexity is significantly arises, and acceptable solutions are not known for such kind of tasks.

One of the possible approaches is using of invariant features. The idea is to find an invariant for each pair of features that probably match each other, so that invariant will be free from distortions. The distortion parameters are determined by feature pair analysis, and appropriate correction is done. After that the feature can be accounted in the search space by the ordinal way. It is easy to see that using of invariant features is a modification of search space decomposition method and hence it ascends all of its negative properties. Nevertheless, discussion of invariant feature using method is outside the bounds of this research and it is a theme of separate research.

In such a way, the significant factor affected to the reliability of the proposed method is a presence of unaccounted geometric distortions. If their parameters are known, these distortions must be compensated in the vectored scene description, since it is not required significant computations in comparison to other methods.

5.1.4.4. Image type and contents

Image content is always a complex task when a wide range of images must be processed. The problem is that the image can contain too many objects of interest, which makes image analysis too complex for real time mode, or – vice versa – image can contain no objects, and image processing becomes useless. The objects on the image can also have slightly different quality. So the proper selection of image representation becomes very important task. Representation must allow dealing with objects that partially invisible, overlapped by other objects or frame bounds, or have bad quality.

Quality of extraction process of selected features is determined by used vectorization algorithm. For example, when the image must be represented as a set of line segments, the contour tracing approaches will better extract shorter lines. Longer lines, which probably have gaps or which are not exactly direct lines, will be separated to shorter ones. And vice versa, HT-based vectorisation will highlight longer lines regardless gaps and curvature, and will suppress shorter lines. Since proposed vectorisation method uses HT approach, it will also have a problem when facing images that mainly consist of short line segments.
There are several important classes of aerospace scenes, which mainly consist of short lines. Firstly, there are town scenes of some scale: long objects (streets etc.) on the image are produced by short ones (like houses and buildings). They cannot be united in one long object due to non-coaxiality and significant gaps between them. Same problems appeared with images of complex coastal line. One more complex kind of scene is large-scale images: they have no man-made straight objects but there is many nature objects of complex shape instead (e.g. mountain chains).

When images of such kind are prevailing, the proposed vectorisation method requires additional tuning or it even need a serious modification. For example, it is possible to make the vectorisation algorithm to extract shorter lines that usually suppressed (like separate building walls in large-scale town scenes). This causes significant increasing of line amount in the scene description. This large amount of lines will significantly slow down either line matching or line gluing. Such a modification will also slow down the processing of all other scenes that does not need small lines for a reliable image registration.

In such a way, one more factor that affected to the image registration reliability is contents of image itself and quality of objects on it. Significant deviation of object characteristics (e.g. prevailing of objects of small sizes or complex forms) can require re-tuning or modification of the proposed vectorisation algorithm.

5.2. Comparison of image registration techniques

5.2.1. Line segment matching vs control point matching

Proposed method of line segment matching has significant advantages in comparison to popular and commonly used control point matching method. Line segments are more regular and common features on the image, than control points. Algorithm of line segment detection, proposed in the chapter 3, is a more formal and more reliable, than control point detector. A set of line segments can be easily structured in order to separate features of different importance, that is hard to do for control points – they carry less information. As a result, registration algorithm can use exactly such amount of lines that allowed by a hardware performance. In the same case the control point matching algorithm would skip some points regardless their importance.

Older algorithms of HT-based line matching [Stockman et al 1982, Davies 1991] were extracting control points as points of line joining and interception, with related problems. Proposed algorithm allows avoiding this.
The line segment matching works slower, when equal quality of lines and control points are used, since it requires additional computations. The computational complexity of both algorithms is the same.

5.2.2. Line segment matching vs. graph matching
Graph matching, in comparison to the line segment matching, allows reach higher accuracy and reliability. However, they also have higher computational complexity, which makes harder to use them in real time applications.

Both accuracy and reliability of line segment matching can be raised using cluster analysis for the responses in parameter space, as it was proposed in [Stockman et al 1982].

5.2.3. Line segment matching vs. area-based methods
Area-based method (least square exhaustive search, extreme correlation, phase correlation in frequency domain etc), in common, allows using all information about image and thus they are potentially more reliable. However, they also requires significant amount of computations. Even if specific hardware is used (e.g. for Fast Fourier transform computation or optical correlation), the speed of area-based registration remains low in comparison to line segment matching.

Another serious disadvantage is sensibility to the image brightness variations, whenever line matching is free from it. Another specific factor is that area-bated methods cannot be modified to find complex transformations of the image.

However, another factor was found that affects on line segment matching reliability – presence of unaccounted geometric distortions. These distortions are insignificant factor in area-based registration since matching areas of both images will change insignificantly. However, object contours will change significantly and it can happen that it is impossible to match both contours using accounted group of transformations. Therefore, the feature-based methods must use geometric corrections always if it is possible.

Area-based methods are less sensitive to the high frequency noise, since the line segment matching uses derive operation that highlights such a noise. However, there is some kind of specific noise, which causes crush of area-based methods. The example is the presence of clouds on the aerial images. Area-based methods require the clouds to be detected and replaced by white noise for the correct image registration. The line segment matching does not require additional processing since cloud images contain insignificant amount of ling and direct lines.
5.3. Conclusion

The primary advantage of the proposed method is its significant performance that becomes especially sensitive when images of the large size are processed. Therefore, the method can be the best alternative in the applications where high performance is the primary task.

The accuracy of the method has the range of one pixel if no uncompensated image distortion is present. This is enough for the most of usual applications.

However, specific problems arise due to significant memory requirements and lower reliability on the some classes of images.

The significant disadvantage of the method is its significant memory requirements. The typical approach to lower these requirements can be hashing accumulator array, implementing iterative search and search space decomposition. First two approaches require significant amount of additional computations, whereas last one lowers the probability that desired position is found correctly. The selection of one or another approach in a specific application will be determined by task conditions and requirements.

The lower reliability of the image registration is another important limitation of the method. The method is not sensitive to noise presence and brightness variation. However, two new factors appear that affected to reliability and accuracy: influence of unaccounted geometric distortions and affections of the image content.

If the parameters of geometric distortions are known or can be estimated, they must be compensated in the vectored scene description, since it does not required significant computations in comparison to other methods. On the opposite case, their presence causes significant reducing of reliability and accuracy.

Variations of the image contents become more important factor when the proposed method is used. A significant deviation of object characteristics (e.g. prevailing of objects of small sizes or complex forms) can require re-tuning or modification of the proposed vectorisation algorithm. However, parameter tuning can be performed automatically by using measured dependences between threshold level, actual and desired number of lines, and primary characteristics of performance, accuracy and reliability.

In such a way, the method of the image registration proposed in this research can be considered as a good alternative in real time applications. The primary area, for which the method was developed, is the aerial and space orientation using visual images. However, the method can be successfully applied in other areas that require the image registration in real time or over large images. However, the method limitations must be kept in mind. They are related with loss of reliability if unaccounted geometric distortions are present on the image or when image does not contain enough amounts of long direct lines required for uniform image matching.
CONCLUSION

This research is devoted to a task of image registration for real time applications.

Existing image registration techniques were examined, and structure of the new method was proposed. The method for fast and reliable image registration can be implemented using fast scene vectorisation and further line matching.

Existing image vectorisation techniques was also examined in order to select methods for a new vectorisation approach. The method must use edge extraction and line detection.

An edge detection method that combines enough speed, reliability and accuracy was proposed and compared with existing techniques. Although the components of this method are already known, the proper selection of components is a complex task. So it required additional analysis to construct proper edge detection.

A new approach – extended Hough transform – is proposed and used to develop a novel line segment detection method. The method characteristics were investigated. These characteristics provide significantly better ratio of speed, accuracy and reliability, than that is possible using known method. So obtained line detector was selected as a base for a next stage of image registration – line matching.

Possible variants of line matching were analysed in order to find an approach with better characteristics. It was found that method of line segment matching based on Standard Hough Transform could provide significant speed while keeping accuracy and reliability high, as it is possible for selected class of the methods. It was found that time complexity of the method is only $O(n^2)$, that is better than known methods can provide. Low computation complexity makes the method very useful in processing large images.

The accuracy and reliability of the method was also analysed. It was found that accuracy of the method is relatively high and allows method to be used on practice. The method shows itself to such influences as image brightness variations and presence of noise, which is a problem for a many popular methods. However, new factors affected to the reliability and accuracy was found: unaccounted geometric distortions and changing of image content.

In such a way, this research presents a novel image registration method that has a good ratio of primary characteristics: speed, accuracy and reliability. These characteristics allow method to be used in the applications where both speed and reliability of image registration play an important role. There are, first of all, the applications related with orientation using visual information in real time.

The proposed method itself or its components can be also used in different areas of computer vision and image processing, since they have good characteristics. Theoretical results like introduction of extended HT or using SHT
for non-point feature matching can extent existing theory of image analysis and can find an application in the different areas of this theory.

**Further research**

Each of parts of the present research – contour extraction, feature detection, and image registration – contains both unsolved tasks and new ways for the theoretical research and practice. So the further additional and deeper research is supposed.

**Contour detection**

The theory of contour detection is relatively well developed, and it is uneasy to find here something fundamental. However, there are set of tasks which have practical importance that must be analysed. There are tasks of image pre-filtration, further development of non-maximum suppression approach, and research of statistical detection characteristics.

**Feature detection**

The key direction of future work here is further development of Extended HT concept. This is an integration of EHT approach with other known feature detection approaches, and application of EHT for the more complex feature detection (arcs, ellipses, polygons etc).

One more separate direction in this area is development of more reliable and effective algorithm of line merging.

**Image registration**

The primary problem of proposed method that does not solved in the current work is a problem of search space expansion. It is often required on the practice to find not only translation but also rotation and scale of one image in relation to another and even more complex transformations. Direct accounting of these factors according to proposed scheme will lead, as noted, to significant memory requirements and loss of accuracy and reliability. Therefore, new approaches must be found and tested. These approaches can include different ways of accumulator array organisation (e.g. hierarchical search schemes, hashing, projection and their combinations) and methods of reply cluster analysis.

Another interesting direction is investigation of combining of proposed method with other image registration methods with different properties and different feature used, in order to characteristics amplification and raising of reliability. The features that can be used in addition to line segments can be large spots and specific contour elements. They can be used when line detection progress is impossible due to environment influence or image character. These
image registration methods itself are well researched, however, usage of their combinations are separate important and complex task.

**Analysis of image registration method in the physical environment**

First stage of analysis is research of proposed methods on the dynamical complex of semi-physical modelling. The suitable tool for this purpose is the dynamical modelling complex based on robot-manipulator PM-01 (PUMA-560 type). This stand was developed on the Automatic Control Department of Bauman MSTU, Moscow for researching the automated control system component behaviour in conditions close to the real environment.

Modern systems of control and data processing consist of many components that must interact in the complex and changing environment conditions and without operator’s help. It is almost impossible to imitate a whole complex of conditions that affected to the separate components. It is often too expensive to produce and lunch working prototype of the system. The solution is to use semi-physical modelling. It allows keeping most significant information links between components and to research influences that will affect to the system in the real environment.

The image analysis system will be affected by different factors:
- Variation of the image characteristics due to the plant motion;
- Variation of observe conditions due to daily, season and weather changing of lightning conditions;
- Raster corruption due to mechanical influences to the camera and vibrations;
- Presence of noise both on image data and synchronization signals in the image receiver and influence of electrical influences.
- Geometrical distortions due to non-linear properties of optical system and 3-dimentional character of the image;
- Asynchronous character of processes of image obtaining, analysis, decision acceptance and plant control.

The structure of the dynamical modelling stand is described in Appendix A.

Next stage is research of image registration methods on the base of space images of earth surface obtained from small-class satellite “Baumanets”. It contains four-channel video camera that operates in the visual and near infrared spectrum diapason. On the target trajectory, camera can provide 4096-pixel width stripe of image data with resolution of 50 meters per pixel. Camera transceiver can send obtained images to the fly control centre for its further analysis. The video system will be used in the task of ecological situation monitoring for the area where the trajectory is going over. Another task is research different variants of position determination methods on the base of comparison of earth images with the stored map. The lunch of satellite is planned on February 2006.

Research of position determination methods is performed by stationary tools of the fly control centre. The desired result of this part of work is conclusion about suitability of proposed methods for position determination tasks, and requirements to the aboard hardware for implementation of these methods.
Related publications


4. Bobkov A.V. Line segment extraction on the image in a task of orientation by visual information. *Vestnik MGTU. Priborostroenie*, №3(48), 2002


References


Ventura J.A. Accurate matching ow two-dimensional shapes using the minimal tolerance zone error *Image and vision computing* 16(1997) pp 889-899


Appendix A. Equipment for dynamical modelling

This research presents a part of more complex work devoted to artificial intelligence technologies in automated control applications, which is carried on IU-1 department of Bauman MSTU, Moscow. The core of the project is a modelling complex based on universal robotic manipulator PM-01.

Purpose of modelling equipment

The important stage of the complex algorithm development in the control systems is the testing of control system component interaction on the dynamic modelling complex. This is necessary because it is impossible providing required quality of system functionality without carrying a large number of research and tests.

The dynamic modelling includes research of properties of environment and different measurement devices – video cameras, ultrasonic sensors, contact sensors – that are included into information processing system, in the well-controlled conditions and with keeping most important links between these devises and other parts.

One of most important factor for the on-board equipment on the moving object is the 3-dimentional motion with 3 rotation axis, and the modelling complex must provide this kind of motion. The robot-manipulator PM-01 (PUMA-560 type) was chosen as a basis of modelling complex, since it has enough number of freedom degrees.

Since the control algorithms and cinematic scheme of the robot does not completely account the specifics of modelling complex, they were significantly modified during modelling complex development.

Modelling complex is used for research work and natural modelling in the following tasks:

- Research of image processing algorithms and determination motion parameters with using parallel computing architectures
- Developing and researching of automated control systems for moving objects in complex situations.
- Modelling of spacecraft docking systems
- Modernisation of robotic control systems for the complex trajectory realisation

Modelling complex includes manipulator PM-01 of six degrees of freedom, manipulator control device “Sfera-36” and control computer (Fig. A.1). The manipulator hand holds a video camera and can move it over the surface model, which contains different kinds of landscape. The control computer contains transputer board (INMOS T805d and T425b-25 processors) with installed frame grabber device. The scheme of modelling complex is shown on Fig. A.1, and common view is shown on Fig. A.2.
Fig. A.1. The scheme of modelling complex

Fig. A.2. Common view of modelling complex
The image of surface model, received by video camera, is digitised by frame grabber and goes to transputer board. The image can be either processed by transputer board, or passed to the control computer, depending on the experiment requirements. The image is compared with the map of surface model, and position of the camera over the model is determined. The correction signal is calculated, and passed through the COM-port to the control device in a form of directive of the ARPS language. The control device determines required trajectory of manipulator, and passes control signals to the manipulator motors. Camera moves into another place, and the described cycle is repeated again.

**Video capturing and processing system**

Video capturing system includes video camera, frame grabber and transputer board. The camera is standard video camera that passes analogue PAL-coded video image to the frame grabber. The camera has a wide visual angle and observes a whole surface model at once, so the only central part of image of lesser size is used. Camera sends images in interlaced retrace mode, so the odd and even frame halves must be separated on the software layer.

The image from the camera is passed to the transputer board INMOS TTFG-4. Frame grabber is the separate board based on T805 transputer and Bt252KPJ20 image digitising micro scheme. It allows capturing grey scale images with 256 grey gradations. The maximum frame size is 512x512 pixels; capturing rate is 18 frames per second. Frame grabber board is inserted into slots of transputer board, and can pass captured images by four separate links, that allows passing video stream with a speed of 24 frames per second.

Transputer board includes four transputers T800 that can provide image processing in a sequential or parallel mode. The software developer toolkit includes program libraries, translators and debugger for languages OCCAM, ANCI C and 3LC.

**Host computer**

Host computer is a personal computer. The transputer board is installed into ISA slots of host computer, and root transputer uses one of its links to connect it. Host computer can obtain either result of image processing or whole image. Host computer provides processing operations that cannot be presented in parallel mode. That is commonly high-level stages of image analysis.

Another purpose of host computer is to produce and send commands to a control device.

**Control device**

Control Device “Sfera-36” is a two-level multiprocessor control device. It is used for contour-positional control of PM-01.

Higher layer is an Advanced Robot Programming System (ARPS). ARPS includes central computer with specialised operational system, user console, remote control device, and storage device. Central computer is the “Electronica 11100.1” (based on K1801 processor). It provides the user interface
for the robot control, trajectory calculation and interaction between low level and peripheral devices. The operational system is Nokia ARPS/M B05.RM-1.

ARPS provides the robot control by one of the following ways:
- By sending directives from the console,
- By running the user program
- By positioning from the remote control device.

Control device “Sfera-36” can process commands in two forms: directives on Nokia ARPS language from the terminal device or required joint parameters from the hand switcher. The host computer can be connected to a terminal connector through COM-ports and emulates terminal signals.

Lower layer includes microprocessor controllers for the robot motors control.

**Manipulator**

Manipulator PM-01 is anthropomorphic robot with six degrees of mobility. The manipulator was modified from its original configuration by turning the sixth joint axis by 90 degrees in order to eliminate the singularity points that original configuration has. The joint characteristics are shown in table 5, and technical characteristics – in table 6.

<table>
<thead>
<tr>
<th>№ of joint</th>
<th>Motion range, degrees</th>
<th>Max speed, degrees/sec</th>
<th>Max moment, Н/м</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>320</td>
<td>1,4</td>
<td>67</td>
</tr>
<tr>
<td>2</td>
<td>260</td>
<td>0,9</td>
<td>113</td>
</tr>
<tr>
<td>3</td>
<td>284</td>
<td>2,1</td>
<td>57</td>
</tr>
<tr>
<td>4</td>
<td>280</td>
<td>4,0</td>
<td>14</td>
</tr>
<tr>
<td>5</td>
<td>200</td>
<td>4,2</td>
<td>12</td>
</tr>
<tr>
<td>6</td>
<td>532</td>
<td>4,0</td>
<td>14</td>
</tr>
</tbody>
</table>

Table 6. Technical characteristics of manipulator

<table>
<thead>
<tr>
<th>Number of joints</th>
<th>6</th>
</tr>
</thead>
<tbody>
<tr>
<td>Driver</td>
<td>DC motors</td>
</tr>
<tr>
<td>Static force in the working point</td>
<td>Up to 60 Н</td>
</tr>
<tr>
<td>Carrying capacity</td>
<td>Up to 2,5 Kg</td>
</tr>
<tr>
<td>Accuracy</td>
<td>0,1мм</td>
</tr>
<tr>
<td>Speed</td>
<td>Up to 1 м/с</td>
</tr>
<tr>
<td>Working space</td>
<td>0,92м</td>
</tr>
<tr>
<td>Robot control</td>
<td>Two-level microprocessor device</td>
</tr>
<tr>
<td>Teaching and programming devices</td>
<td>Console, remote control device</td>
</tr>
<tr>
<td>Programming language</td>
<td>Advanced Robot Programming System Language</td>
</tr>
<tr>
<td>Power supply</td>
<td>220 V, 50 Hz</td>
</tr>
</tbody>
</table>
Software

The main difficulty of software development is need to synchronize the work of several components – frame capturing, image analysis and robot control, where each of them has its own time characteristics, modes and requirements to hardware.

The module of frame capturing is executed on the transputer environment. It must be implemented on the high level language (OCCAM or 3LC) and compiled into the executable program of BTL format.

The loading and starting of BTL-module is performed by loader program, which is distributed with INMOS transputer SDK. The loader is implemented on the host computer under the operational system of MS-DOS type, and cannot be modified or rewritten without additional documentation. On the other hand, the BTL module executed in the transputer board and cannot get access to all required resources of the host computer. Furthermore, debugging of BTL module is uneasy task, since it is possible using build-in tools only.

Therefore the following model of program complex and data exchange was developed.

The frame capturing module periodically acquires a picture from camera and stores them as files of raw data using the host file system. Files can be placed either on the RAM-disk (to perform fastest data exchange), on the hard disk drive (for the further off-line analysis) or passed to the remote host using NetBEUI protocol (e.g. if the speed of the local host is not enough, and more powerful tools are required). Host computer uses multitask operational system, and executes loader program, image analysis module and robotic control module in the pseudo parallel mode.

Image analysis module periodically checks the directory where images are stored in order to find latest image from the camera. Then the image analysis is performed according to the methods presented in the current research, and translation value of the current frame in relation to the sample is calculated. The sample can be either the first frame marked as a map (map traveling mode) or previous frame (frame by frame traveling mode).

Using the translation value, the robot control module determines required co-ordinates \(A \begin{bmatrix} X, Y, Z, \phi_x, \phi_y, \phi_z \end{bmatrix}\) of manipulator, and sends the command \textit{TEACH} \(A\) through the COM port to the robot control device. This command determines the new position of the point \(A\), which robot must reach. Reply from the COM port and data accept is performed in the separate thread.

The robot control device performs simple cycled program that makes robot to follow into the point \(A\):

1. \textit{GO} \(A\)
2. \textit{GOTO} 1

When the \textit{TEACH} \(A\) directive is passed, the position of point \(A\) is changed, and robot begins moving into the new position. As a result, the image obtained by the camera is changed, and process is repeated.
Appendix B. Software for the off-line image analysis

This mode uses the sequence of images either stored on the disk when the robot moves by the given finite trajectory, or sequence of artificial images.

RoboVision Module

This module was developed in co-operation with Anton Medvedev, Bauman MSTU. The aim of the module is research of edge detection filters and basic filtering operations. Module contains basic tools of the image filtration, different edge detectors (both known and newly developed for the research purpose), line and angle detectors, and statistics collection. This module was used to obtain significant part of experimental results of Chapter 2.

Fig. B.1. Common view of RoboVision module interface
WinLoader Module

This module was used for research of line detection algorithms and development of image registration methods. It allows researching all stages of the line extraction and investigating different methods of image registration. The module contains tools for time measurement of all stages duration, and has an ability to collect and store the statistic data about image registration for further analysis by external tools. This module was used to obtain experimental results of chapters 3 and 4.

Fig. B.2. Common view of WinLoader module interface
**StatCheck Module**

This module was developed for analysis of statistics collected by WinLoad module in order to research the accuracy and reliability of different image registration approaches in the different conditions. The StatCheck module oriented to the processing of large data volumes, which cannot be effectively processed by traditional tools. This module was used to get experimental data on accuracy and reliability of different image registration approaches in the end of Chapter 4.

![Common view of StatCheck Module interface](image)

**Fig. B.3.** Common view of StatCheck Module interface