

Digitization of Handdrawn Diagrams

Vorgelegt von:

Regina Altmann

Matrikelnr.: 4597011

e-Mail: Regina_Altmann@yahoo.de

Masterarbeit

im Studiengang Informatik

Freie Universität Berlin

Gutachter: Prof. Dr. Raul Rojas

Dr. Hamid Mobalegh

Betreuer: Dr. Ronny Hänsch

Eidesstattliche Erklärung

Ich erkläre hiermit an Eides statt, dass ich die vorliegende Arbeit selbstständig und ohne Benutzung anderer als der angegebenen Hilfsmittel angefertigt habe; die aus fremden Quellen direkt oder indirekt übernommenen Gedanken sind als solche kenntlich gemacht. Die Arbeit wurde bisher in gleicher oder ähnlicher Form keiner anderen Prüfungskommission vorgelegt und auch nicht veröffentlicht.

Berlin, den 07. April 2015

Regina Altmann

Contents

Abstract	III
Zusammenfassung	III
1 Introduction	1
1.1 Motivation	1
1.2 Structure of the thesis	2
1.3 Related Work	2
2 Materials and Methods	3
2.1 Preprocessing	3
2.2 Median Filter	4
2.2.1 Adaptive Thresholding	5
2.2.2 Skeletization	7
2.3 Recognition	10
2.3.1 Generalized Hough Transformation	10
3 Implementation	14
4 Results	17
5 Conclusion	20
List of figures	22
References	23

Abstract

For scientific documents, which nowadays are usually created in digital form, it is often necessary to include diagrams or models. The faster and more convenient method to create those diagrams however, is still drawing them by hand. Scanned or photographed, those drawings are rarely in such a neat quality that they smoothly integrate into the digital document. For this thesis a program was created to recognise the individual shapes in a diagram using a generalised Hough transformation and redraw the diagram with a drawing function.

Zusammenfassung

Für wissenschaftliche Texte, die heutzutage oft in digitaler Form angefertigt werden, benötigt man häufig Diagramme oder Modelle, welche sich jedoch bequemer und schneller mit der Hand gezeichnet anfertigen lassen. Eingescannt oder ab fotografiert sind diese Zeichnungen aber selten von so guter Qualität, dass sie sich nahtlos in ein digitales Dokument einfügen lassen. Im Rahmen dieser Arbeit wurde ein Programm entwickelt, dass für das Erkennen der einzelnen Formen eines Diagramms die generalisierte Hough Transformation verwendet und das Diagramm mit einer Zeichenfunktion neu erstellt.

1 Introduction

1.1 Motivation

In the present time a lot of things are created in digital form: electronic books, electronic appointment calendars and address books. Not only is that more convenient but also very compact and can be organized very easily. Before the age of notebooks, smart phones and e-book readers everything was written down on paper by hand or typewriter. Some people and companies were keeping huge amounts of paper documents and working with them was a skill on it's own because it was not so easy to find the desired information. In the case of text the digitization was fairly easy because it is limited to a certain set of letters. With the help of a keyboard or touch screen it is very comfortable to type in the desired text. The computer recognition also has the advantage that the limited set of letters will usually form a limited set words. With this information false recognition can easily be minimized. A lot of non-fictional texts however also contain diagrams, tables and illustrations.

The one to one digitization of these images can be simply realized by scanning or photographing, what on the downside is messy, impractical and unprofessional. For a uniform clean look it was necessary to invest some time in creating the digital version of these diagrams, tables or illustrations a digital image editor by hand.

The computer program developed for this work tries to automatically generate digitized diagrams from an input image. The main applications will probably be for transcripts of lectures or seminars in universities and companies.

The most commonly used device to capture the image in the present time is the smart phone because it is always at hand. The condition under which the diagrams are drawn and also the capturing mode render his task into a problem that is not as trivial as it seems at first glance.

The first problem is that these images will carelessly be drawn in a hurry because they need to follow the speed of explanations. Therefore we can not simply look for a perfectly formed shape but in fact a circle will look different even for every instances in the same image.

The second problem is that the background medium can range from paper, to blackboard and whiteboard resulting in writing utensils like ball pens and pencils, as well as chalk and whiteboard markers. Dark background with light drawings have to be handled as well as the opposite.

Because smart phone photographs in closed rooms with bad lighting are not very hight quality there can be a lot of noise, low contrast and a irregular

illumination in the captured image.

The size of the image and also the size of each shape is unknown and can vary considerably. This does not allow the exclusion of a large range of possible shape sizes.

At last the presence of text inside the different objects in the diagram complicates the recognition problem again. All of these factors need to be taken care of in the process of digitization.

1.2 Structure of the thesis

The thesis is structured in five chapters. After the first introductory chapter, a detailed information about the methods used to create a diagram from the input image is given in chapter 2. Chapter 3 then describes the design and implementation of the created computer program. The results from that program will be presented in chapter 4 and finally, chapter 5 draws conclusions and discusses possible faults and improvements.

1.3 Related Work

There is a lot of research done in the field of object recognition, however none for the offline digitization of diagrams. There are approaches for online recognition for example by Stoffel and Rojas [Stoffel] of commutative diagrams which have the advantage of receiving stroke information and the knowledge about the size and characteristics of the input medium.

Another field that is very popular is the digitization of text. The advantage of text recognition is the opportunity to examine the resulting word with a dictionary of words. One approach that uses an offline technique is described in [Blum].

2 Materials and Methods

The following chapter describes the methods used for the digitization of hand drawn sketches. In the first part it is described how to improve and alter the original image to make the recognition of the shapes in the drawing easier. This part of the processing chain is called preprocessing. The second part is the recognition part and deals with finding and assigning the shapes in the image. After finding the shapes in the image the drawn reproduction of them is trivial and shortly explained later in the implementation part.

2.1 Preprocessing

Colour noise, reflections, low contrast, smudge and other artefacts are commonly present in photographed images and lower the quality of the recognition. It is desired to create images, which are as similar in quality to each other as possible.

Because this part operates automatically it is important to find one good solution that performs well for a set of different problems. There is a set of methods to enhance contrast, remove artefacts and noise and improve the edges of the drawn shapes. As the image quality, object size, colour and background are unknown, operations have to be selected carefully. One operation that improves an image can produce negative effects in another.

For example: a low contrast image can profit from an increased contrast if the background is mostly uniform. The presence of noise, smudges or structures in the background can induce the highlighting of those even more.

The following processing chain is used for the training and recognition images in the same way. At first the image is converted into grayscale and then smoothed with a median filter. The next step is explained in chapter 2.2.1 and creates a binary image using adaptive thresholding. Finally, one pixel wide skeletons of the shape are created using a skeletonization algorithm, which is explained in chapter 2.2.2. The skeletons are used to reduce processing time and compensate small drawing irregularities, while keeping only the important information about an object.

2.2 Median Filter

The special characteristics of the median filter is that it removes noise, while preserving edges. It basically locates extreme intensities in the image and replaces them with more balanced values. For this a window slides over the image and replaces the current intensity with the median of the neighbouring intensities. The size of the neighbourhood taking into the calculation depends on the size of the sliding window. If the window is chosen to large the smoothing is very strong and can erase important object information.

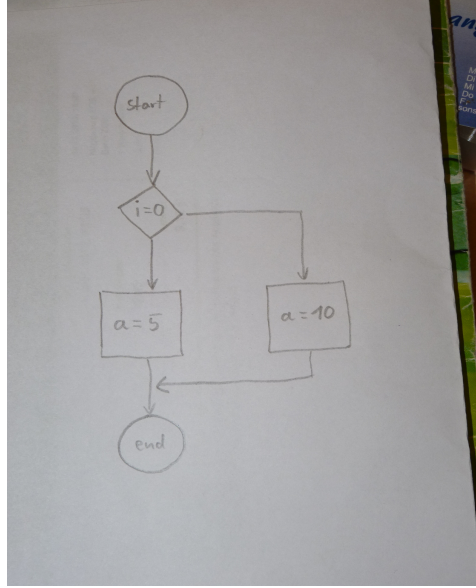


Figure 1: The original image

The three images (1 to 3) illustrate the importance of the smoothing filter. The first figure shows the original image without any alteration. The two images on the next page show only a section of the processed image to clarify the details.

In the left part, showing the result after the application of adaptive thresholding, the differences are not so obvious. The background noise is gone but the object looks the same on the first glance. With a closer look it can be seen, that the lines are more solid after using a median filter. As an effect this creates a clear result of the skeleton algorithm.

The fractured line in the top image creates a very messy result for the skeletonization and would lead to faulty entries in the R-Table and unstable recognition results.

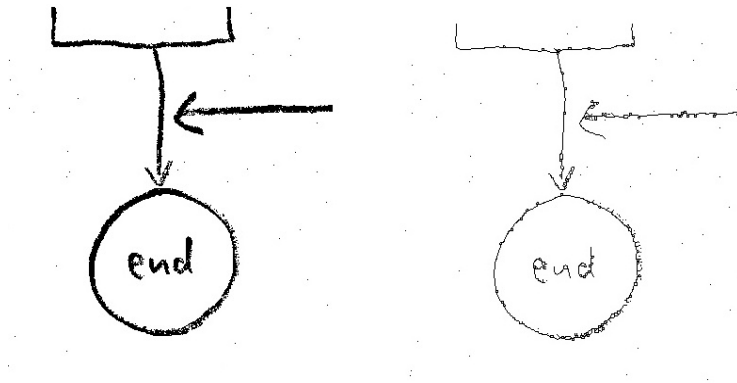


Figure 2: A section of figure 1 with thresholding (left) and skeletization (right) but without smoothing by a median filter.

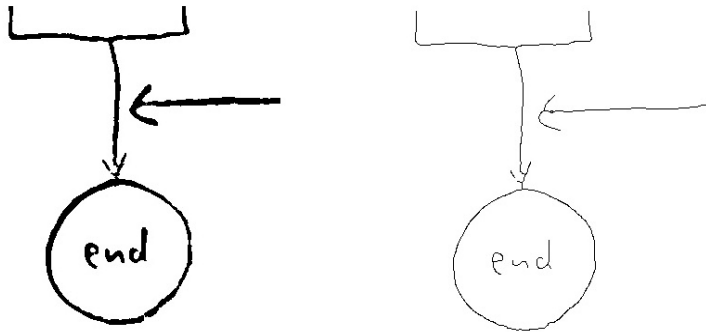


Figure 3: A section of figure 1 processed with the use of the median filter.

2.2.1 Adaptive Thresholding

Thresholding is a segmentation technique to separate the image into objects of interest and background regions. This technique is mainly used when objects and background do not have a lot of detail. The output is a binary image with black pixels corresponding to foreground and white pixels to background, or vice versa. Each pixel $g(x,y)$ in the image is compared to a threshold T and set to black or white, corresponding to their intensity in relation to the threshold.

$$g(x, y) = \begin{cases} 0 & \text{if } f(x, y) > T \\ 1 & \text{if } f(x, y) \leq T \end{cases}$$

The threshold can be set by a user or determined by the image's histogram. One method is to calculate the mean threshold, simply by adding all intensity values i and dividing them by the number of pixels (n).

$$T = \frac{\sum i}{n}$$

Figure 4 points out the limitations of the global threshold. The result in the middle was calculated with a global threshold. Under uniform illumination the background and regions of interest form two peaks in the histogram. Those peaks can be easily separated by a threshold. The uneven illumination in the original image leads to overlaps of fore- and background regions in the histogram and therefore to a faulty segmentation.

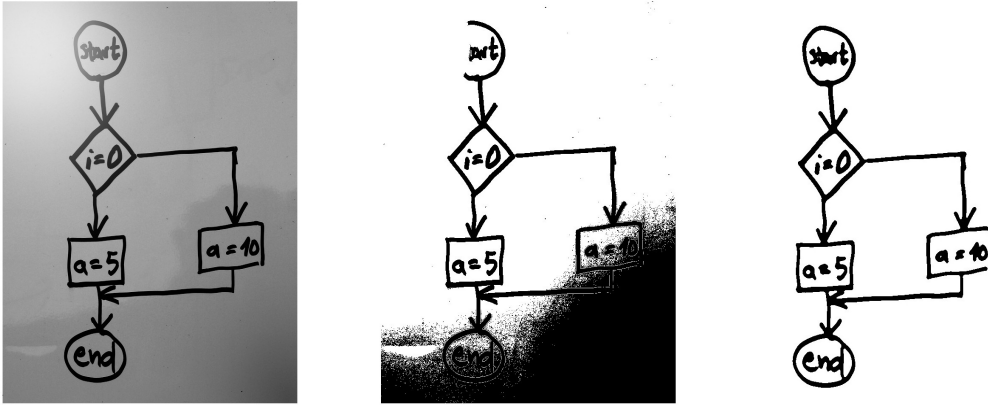


Figure 4: Thresholding of the original image (left) with a global threshold (middle) and adaptive thresholding (right)

This problem can be eliminated if the threshold is permitted to adapt dynamically to different locations in the image.

One of the different ways to achieve this is to create a model of the background and subtracts it from the image. The idea of the other approach which was used to create the solution on the right in picture 4 is to partition the image into $m \times m$ subimages and compute a threshold $T_{i,j}$ for each

of them. The calculation for the threshold remains the same as discussed previously. For the final result the regions are united [Gonzalez02].

2.2.2 Skeletization

The word skeleton (from Greek skeletos "dried up" Oxford Dictionary of English 2nd edition 2005) refers to an inner supporting structure.

With a thinning algorithm, objects in a binary image can be reduced to one pixel wide lines, called skeletons. Those skeletons or core lines contain the essential information of the components that improve the recognition and redundant information are erased. The thickness of the drawn line in our case is irrelevant as it carries no information which are helpful for recognition.

This reduction of information for each shape is able to significantly accelerate the computation in the recognition part and reduce allocation of memory. It is also able to reduce small drawing irregularities

To ensure the characteristics of each component remains unchanged, the following rules have to be met.

1. The number of connected lines must be equal to the original number of connected regions.
2. The obtained lines should have a minimal number of pixels with 8-connectivity.
3. Spurs are undesired and should be minimized, as they can originate from noise in the image.
4. The pixels removed by each thinning iteration should not be the end pixels of a line because that would shorten the line and change its location.
5. For a best possible solution the resulting lines should approximate the medial lines of the original region. The true medial lines can not be archived because they could run through half pixel space which can not be represented digitally.

There are different algorithms for skeletization. In this work the approach from Zhang and Suen: "A Fast Parallel Algorithm for Thinning Digital Patterns" is used [Zhang84] and essentially strips away the outer layers until only the skeleton remains.

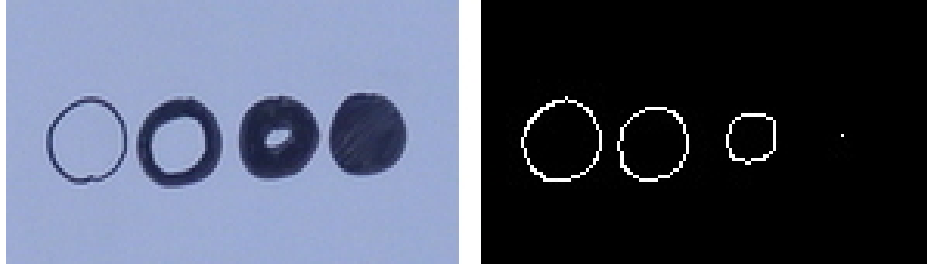


Figure 5: The result of skeletization using the described algorithm

```

While points are deleted do
  For all pixels P1(i,j) do
    if (a)  $2 \leq B(P1) \leq 6$ 
      (b)  $A(P1) = 1$ 
      (c) in odd iterations:  $P2 \times P4 \times P6 = 0$ 
      (c') in even iterations:  $P2 \times P4 \times P8 = 0$ 
      (d) in odd iterations:  $P4 \times P6 \times P8 = 0$ 
      (d') in even iterations:  $P2 \times P6 \times P8 = 0$ 
    then
      Delete pixel P1(i,j)
    end if
  end for
end while

```

Where P1 is the current contour point, P2 to P9 are the neighbours according to figure 6, B(P1) is the number of non-zero neighbours and A(P1) is the number of 0 to 1 transitions in clockwise direction of the neighbourhood.

P_9 $(i-1, j-1)$	P_2 $(i-1, j)$	P_3 $(i-1, j+1)$
P_8 $(i, j-1)$	P_1 (i, j)	P_4 $(i, j+1)$
P_7 $(i+1, j-1)$	P_6 $(i+1, j)$	P_5 $(i+1, j+1)$

Figure 6: The 3x3 window around examined contour point P1 [Zhang84]

By the first condition (a) endpoints of lines are preserved, while condi-

tion (b) prevents the disconnection of segments. If condition (c) and (d) are met, P1 is a south-east boundary or north-west corner point. P1 might be a north-west boundary or south-east corner point if conditions (c') and (d') are satisfied. Those point are not relevant for the skeleton and can be deleted. As long as points are still being removed the iteration continues. This algorithm was chosen because it is very fast and yields good results. Image 5 shows the result of a skeletization with the proposed method.

2.3 Recognition

After transferring the image to a binary representation containing only skeletons the next step is to find and identify each shape.

There exists a lot of scientific work for finding shapes or certain objects in an image, especially for automated industrial production and quality control. In contrast to local detection methods, a global method such as the Hough transform does not require previous calculation to group edge points together.

The Hough transform uses a voting mechanism where each contour point only contributes a small fraction to the recognition of an object. Practically this means that a shape can still be recognized with less than 50 percent of the contour points. This makes the technique relatively robust to noise, gaps and partial deformations and is in this aspect superior to other techniques including regression.

The basis for identification of lines was introduced by Paul V. C. Hough as early as 1962. Richard Duda and Peter Hart extended the method for arbitrary shapes in 1972 and called it a "generalized Hough transform". But it was only later in 1981 that the generalized Hough transform was popularized by Dana H. Ballard through an article called "Generalizing the Hough transform to detect arbitrary shapes" [Ballard80].

This generalized Hough transform is the technique used in this work for the detection of shapes in a hand drawn diagram and described in the following section. The original Hough transform, which works very similar but uses the unique properties of a simple analytic shape, described by an equation with a low number of parameters, like the line equation. It will not be explained in detail in this work.

The generalized Hough transform was chosen to create the possibility to extend the recognition for arbitrary customized shapes, like clouds. Another reason was to find shapes that are composed of several lines directly and without the need of grouping detected lines together.

2.3.1 Generalized Hough Transformation

In contrast to the original Hough transformation, which was aimed at detecting simple analytic forms, the generalized Hough transform can directly detect arbitrary shapes. Basically it is an improved form of template matching.

One point in the template image, which can be the center, is chosen to be the reference point (x_c, y_c) for the object.

The generalized Hough transform requires the exact definition of a shape to be translated into a structure called R-table. The table is constructed from all displacement vectors (d_x, d_y) and their associated gradient angles ϕ (see image 7).

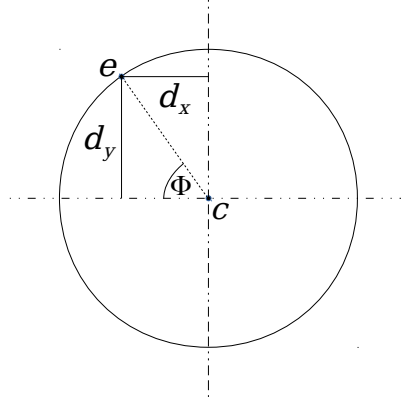


Figure 7: An example for generation of R-table information from a template

The gradient angles $[0, 2\pi]$ are discretized by dividing them into sub intervals called bins. Each index in the R-table represents one bin and contains all the displacement vectors for these gradients.

The number of R-tables in total correspond to the templates used to create the database.

For the recognition, an accumulator table is created in the size of the image and initialized with zero. The created space is also referred to as Hough space. The votes from each point will be recorded there later.

The new image is searched for contour points and if one was found, their gradient is computed and used to look up the deposited displacement vectors. It is now possible to compute the location of all possible reference points c^* in the following way:

$$c^* = (x + d_x^n, y + d_y^n)$$

,with n being all displacements in the current bin.

Figure 8 illustrates an example of two edge points with the same gradient and the calculation of possible reference points. The location in the center would receive two votes and become the highest value. Each calculated point, represents a vote for a reference point and the value at it's location in the

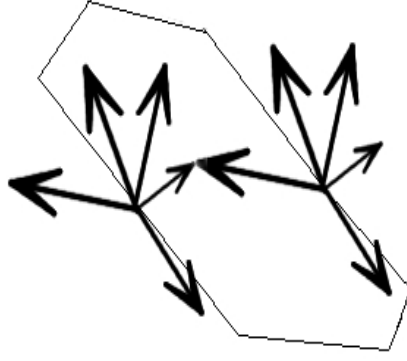


Figure 8: Visualization of the R-Table information for two edge points

accumulator table is incremented. After all contour points have been visited a shape can be found by locating the peak in the accumulator table.

A peak in the Hough space is the estimate for the location of the object. A lot of noise in the original image will result in region with high values. In this case the center of the region provides an estimate for the object location. If there is more than one object matched by a model, there will be several peaks in the accumulator table. It is possible to find all peaks if all the high values are selected up to a border where a peak is not considered significant any more. However, chapter 4 illustrates that it is not trivial to determine whether a high value is significant or not.

Another problem with the generalized Hough transform is that computation time and memory allocation increases exponentially with dimensionality and makes it unsuitable for real time detection.

So far we have only considered objects which have the exact same orientation and size. However this is seldom the case and for the recognition it is crucial to extend the method with scaling and rotation. The accumulator matrix is now no more 2-dimensional but 4-dimensional.

$$a = (x, y, s, \theta)$$

,with s being the scaling and θ the rotation.

The construction of the R-table is not effected by the extension but only the formula to calculate possible reference points.

R-table information now need to be scaled and rotated. In the following equation p is the scaled displacement, R is the rotation matrix and v is the

shift to the contour point.

$$p = \begin{bmatrix} d_x * s \\ d_y * s \end{bmatrix} \quad ; \quad R = \begin{bmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{bmatrix} \quad ; \quad v = \begin{bmatrix} x \\ y \end{bmatrix}$$

$$c* = \begin{bmatrix} x_c \\ y_c \end{bmatrix} = Rp + v$$

After incrementing all possible points in the accumulator matrix the space will be searched for peaks, as explained before.

3 Implementation

Because the program should be usable for all different kinds of people it was necessary to provide a graphical user interface. An easy and effective way to create such a GUI for the program, was to use QT. QT is a C++ development framework with the advantage that it is platform independent. However the code still needs to be compiled for the different platform separately. This is important because the program should not only be usable on Unix systems, which the author uses on her machine.

The connection with C++ also created the opportunity to use the OpenCV (Open Source Computer Vision) library. As the name suggest, it mainly contains algorithms for image processing, pattern recognition and machine learning.

The whole program is made up of two main parts:
The training of the Database and the digitization. The processing chain of the first part is shown in image 9 and described in the following section.

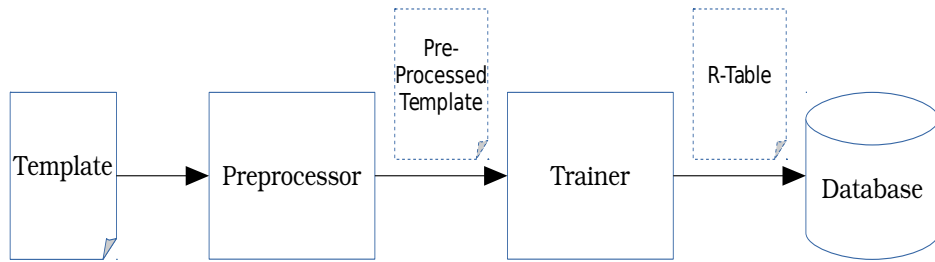


Figure 9: The processing chain of the proposed digitization program for filling the database.

At first the selected template images are run through the Preprocessor, which will convert it into grayscale, smoothing it with a median filter, apply adaptive thresholding and finally thin the image to a skeleton.

From the preprocessed images, R-Tables are extracted and assigned to the specified Shape object. All R-Tables are stored in the Database, to assure an easy access in the recognition part. If the program is closed, the information from the Database are saved into a file.

If the program is started for digitization later, this file is loaded into the internal Database. A filled database already comes with the program but in case it is missing, the recognition part is blocked and R-Tables have to be created first by selecting "Training" in the menu.

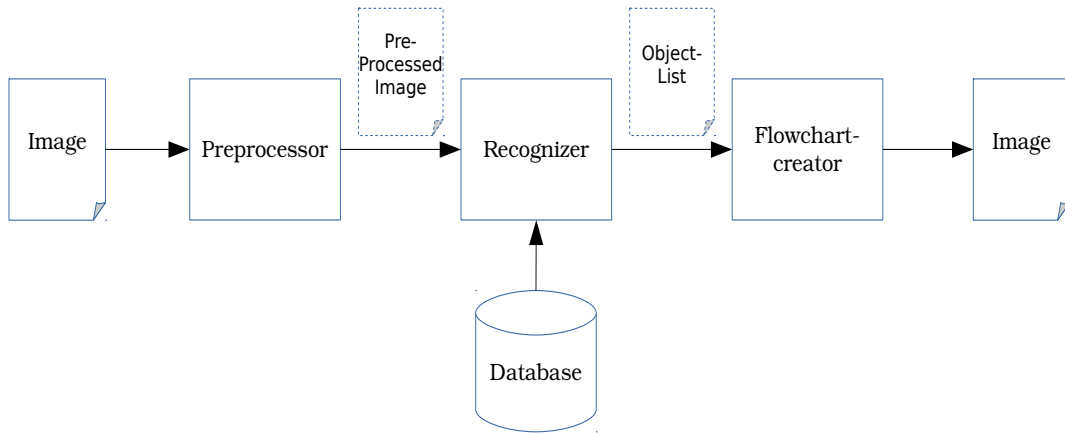


Figure 10: The processing chain of the digitization part.

The processing chain for the digitization part of the program is illustrated in image 10.

At first the image, that is selected for digitization, goes through the same preprocessing steps as the template. Because the generalized Hough transformation is a brute force approach it is very computationally expensive. To reduce the load as much as possible all black edge pixel are saved in a Matrix and background pixel are left out.

In the Recognizer a hough space is created for each R-Table and it's scaling and rotation.

The scaling is set to 0.3 until 3 times the size of the image with a step size of 0.025 and the rotation ranges from -20 to 20 degree in 5 degree steps.

Because many passes through the edge image would be necessary the elimination of background pixels reduces the iterations significantly.

After the creation of a hough space it will be analysed for maximum values. The obtained maxima are saved into the object class together with the corresponding position, intensity, scaling- and rotation information.

As a result the hough space can be overwritten in the next iteration to save memory space. Each object is appended to an object list.

The recreation of the found shapes is handled by the FlowChartCreator.

It receives the object list and initially tries to eliminate similar duplicates of an object, which vary just a little in position, scaling or rotation.

The reason of this is that noisy and distorted objects do not result in one single peak but rather an area of peak values. The highest peak values is selected from the list as a correctly found shape. All shapes with high values

that intersect with the area of the previously found object are deleted from the list.

Afterwards the cleaned up object list can be used to draw the shape.

The functions for drawing each of the different shapes are adjusted with the scaling and rotation information. The position of the curve is calculated from the center point and the the size information. A second scene in the main window is used to display the digitized image.

Finally, the user can adjust the result to a certain degree. By a left click on an object it is selected and can be dragged to a different position. With the deletion key "Del" on the keyboard the selected object will be deleted permanently. To change the size of an object it can be selected and adjusted with the "+" and "-" key.

If the user is satisfied with the end result it can be saved into an vector image (.svg).

4 Results

After all details about methods and implementation have been pointed out, the results can be discussed in this chapter.

As already shown earlier, the methods of the preprocessing produce quite satisfying results. The source image in figure 11 has a low contrast and irregular lighting and is segmented and thinned correctly.

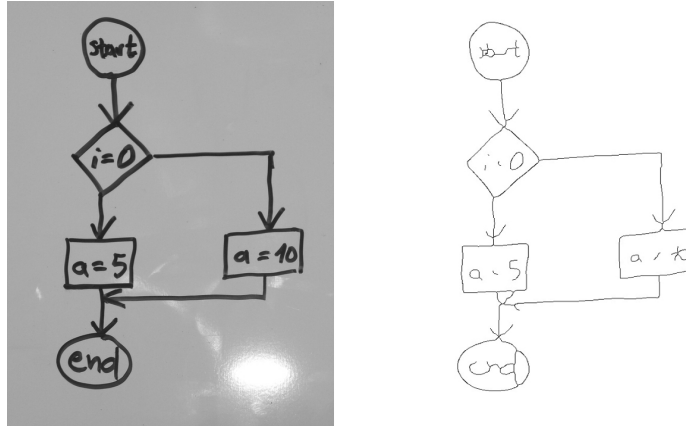


Figure 11: A photograph after the preprocessing (right).

The automatic detection if an image has a dark or a light background was not possible because the information of the amount and thickness of lines is unknown. Therefore the histogram does not give any reliable information about which area corresponds to which area, partly also because of low contrast and irregular lighting.

It would be possible to just look at border pixels to determine the background colour. The author decided against this, as the border of an image could also consist of surrounding area around the drawing medium or .

For this reason there was a selection button integrated in the GUI.

The whole recognition process did not meet the expectations in every aspect. Diagrams created by an image editor program yield quite good results even for distorted shapes. The result of the recognition of circles with different scaling and a distortion can be seen in figure 13.

Because the circle is symmetrical and not effected by rotation, there is another example shown in figure ???. One false positive was detected in the image, an arrow pointing upwards, but everything else is digitized correctly.

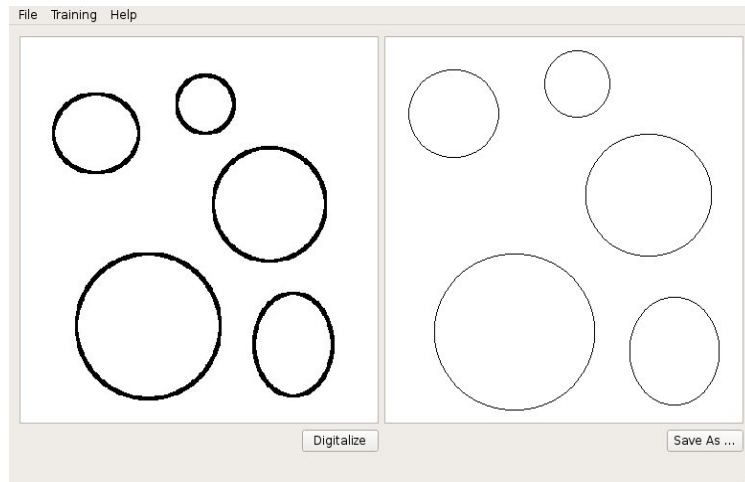


Figure 12: Digitization result (right) for circles created with a digital image editor (left) including varying scaling and distortion.

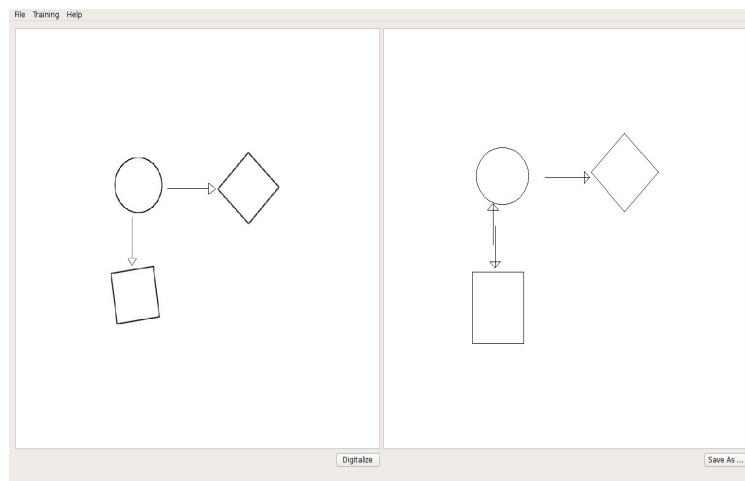


Figure 13: Digitization result (right) for a diagram created with a digital image editor (left) including rotation.

For a handdrawn image with a digital image editor the results still seem very acceptable (see figure 14).

However for handdrawn diagrams, which are captured with a camera, the results is not satisfying at all. Unfortunately it was not possible the find and eliminate the reason for this behaviour. For the result see figure 15 on the next page.

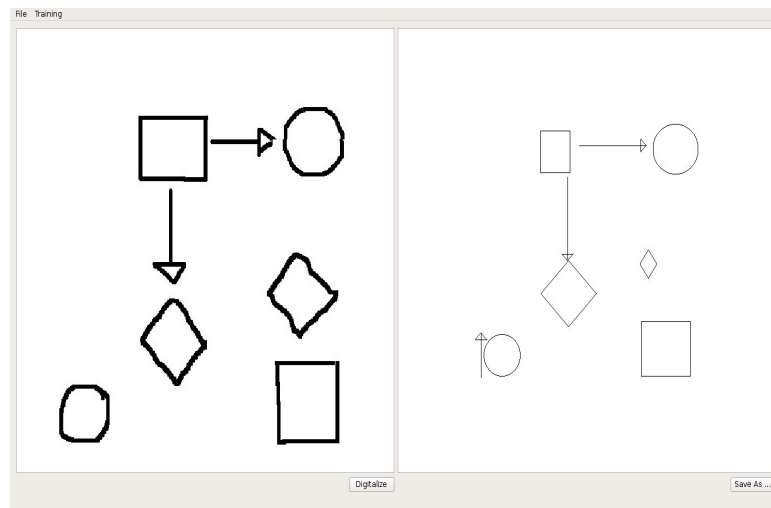


Figure 14: Digitization result (right) for handdrawn shapes with a digital image editor (left).

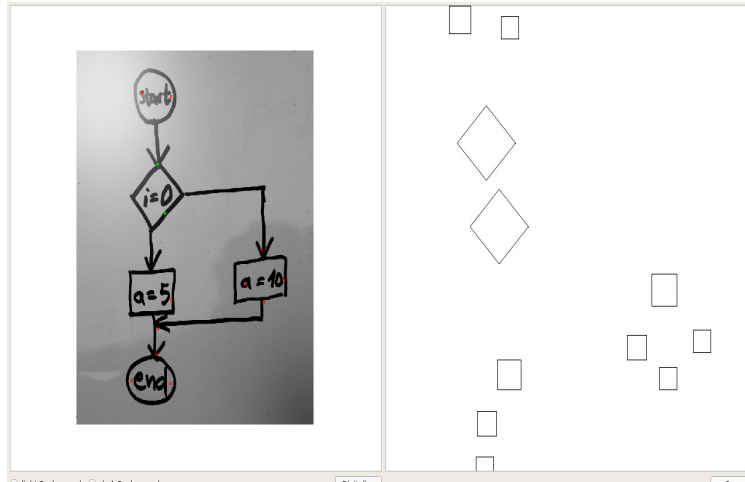


Figure 15: Digitization result (right) for handdrawn diagram (left).

5 Conclusion

In this work it is shown that the generalized Hough transformation is not suitable for digitizing hand drawn diagrams.

One of the major drawbacks of this method is the processing time and memory needed for execution. For this reason large images, which usually photographs are, require so much time that it is not suitable for the intended purpose.

Several acceleration methods, like iterating only through edge points instead of the whole image for each run of scaling, rotations and all R-tables or saving only the important data from each accumulator table have been implemented without effecting in an acceptable speed.

One technique that could further improve the computational load would be to translate the image into Fourier space and perform all operations in the frequency domain.

The other and probably most important drawback is the success of the recognition.

While it is in fact possible to recognize all of the shapes in the diagram correctly but only with the cost of setting the threshold very low. In return the low threshold allows a lot of false positives to be accepted.

The program includes a function to adjust sizes and positions as well as deleting redundant objects and is done very conveniently but does not change the fact that the recognition does not perform very well.

In the future two more options could be implemented. The first is to find the the highest value in the accumulator table and add the object to the acceptance list. Then decrease all locations in Hough space, that the particular shape voted for. This could decrease noise by eliminating false possibilities. This could not be tested because only the values exceeding the threshold are saved and all redundant information discarded to save computations and memory space.

The other technique would involve to look at the original image again. The curve of the detected shape could be mapped onto the corresponding area and reviewed for a match.

Even if there are still some idea to improve the result of the digitization program it has to be said that the generalized Hough transform does not seem suitable for the proposed problem. There are too many possibilities that an object could look like to adjust it to a stable recognition. The occurrence of many template objects in the processed image has a negative influence as well. Suppose there are four objects in a square arrangement and connected by arrows. Then the space in the middle will be recognized as a rectangle because the location will receive a lot of votes from the surrounding arrow tails.

The generalized Hough is a very powerful and robust tool for the detection of one specific object in an image but it is just not suitable for the proposed problem of this work.

List of Figures

1	The original image	4
2	A section of figure 1 with thresholding (left) and skeletization (right) but without smoothing by a median filter.	5
3	A section of figure 1 processed with the use of the median filter.	5
4	Thresholding of the original image (left) with a global threshold (middle) and adaptive thresholding (right)	6
5	The result of skeletization using the described algorithm	8
6	The 3x3 window around examined contour point P1 [Zhang84]	8
7	An example for generation of R-table information from a template	11
8	Visualization of the R-Table information for two edge points	12
9	The processing chain of the proposed digitization program for filling the database.	14
10	The processing chain of the digitization part.	15
11	A photograph after the preprocessing (right).	17
12	Digitization result (right) for circles created with a digital image editor (left) including varying scaling and distortion.	18
13	Digitization result (right) for a diagram created with a digital image editor (left) including rotation.	18
14	Digitization result (right) for handdrawn shapes with a digital image editor (left).	19
15	Digitization result (right) for handdrawn diagram (left).	20

References

- [Gonzalez02] GONZALEZ, Rafael C.; WOODS, Richard E.: Digital Image Processing, 2nd edition. Prentice-Hall, Inc., 2002 page 123-124; 595-610 ISBN: 0-201-18075-8
- [Zhang84] Zhang, T. Y.; Suen, C. Y.: A fast parallel algorithm for thinning digital patterns. Communications of the ACM, Vol. 27, No. 3. (1984), pp. 236-239
- [Davies90] Davies, E.R.: Machine Vision: Theory, Algorithms, Practicalities. AP Academic Press Limited, 1990, ISBN 0-12-206090-3
- [Jain95] Ramesh Jain, Rangachar Kasturi, Brian G. Schunck: MACHINE VISION. McGraw-Hill, Inc., 1995, ISBN 0-07-032018-7
- [Ballard80] Ballard, D.H.: Generalizing the Hough transform to detect arbitrary shapes, Pergamon Press, 1980
- [Ballard82] Ballard, D. H.; Brown, C. M.: Computer Vision; Prentice-Hall, Inc., 1982, ISBN: 0-13-165316-4
- [Bow02] Bow, Sing-Tze: Pattern Recognition and Image Processing; Marcel Dekker Inc., 2002, ISBN 0-8247-0659-5
- [Bonnet01] Bonnet, Noel: An unsupervised generalized Hough transform for natural shapes; Pattern Recognition 35 (2002) 1193 – 1196, 2001
- [Phillips94] Phillips, Dwayne: Image Processing in C, Second Edition; R and D Publications, ISBN 0-13-104548-2, 1994
- [Samal97] Samal, A.; Edwards, J.: Generalized Hough transform for natural shapes; Pattern Recognition Letters 18 (1997) 473-480, 1997
- [Blum] M. Blumenstein, C. K. Cheng and X. Y. Liu: NEW PREPROCESSING TECHNIQUES FOR HANDWRITTEN WORD RECOGNITION, School of Information Technology, Griffith University
- [Stoffel] Stoffel, Andreas; Rojas, Raul: Recognition of On-Line Handwritten Commutative Diagrams; Universitaet Konstanz, Fu Berlin
- [Kara] Levent Burak Kara, Thomas F. Stahovich: Hierarchical Parsing and Recognition of Hand-Sketched Diagrams