A Portable DARC Fax Service

Thesis project done at Data Transmission,
Linköping University
by

Björn Husberg

Reg nr: LiTH-ISY-EX-3326-2002
A Portable DARC Fax Service

Thesis project done at Data Transmission,
Linköping University
by

Björn Husberg

Reg nr: LiTH-ISY-EX-3326-2002

Examiner: Ulf Henriksson
Supervisor: Robin von Post
Linköping, October 28, 2002
En Bärbar Faxtjänst För DARC
A Portable DARC Fax Service

Björn Husberg

Abstract
DARC is a technique for data broadcasting over the FM radio network. Sectra Wireless Technologies AB has developed a handheld DARC receiver known as the Sectra CitySurfer. The CitySurfer is equipped with a high-resolution display along with buttons and a joystick that allows the user to view and navigate through various types of information received over DARC.

Sectra Wireless Technologies AB has, among other services, also developed a paging system that enables personal message transmission over DARC. The background of this thesis is a wish to be able to send fax documents using the paging system and to be able to view received fax documents in the CitySurfer.

The presented solution is a central PC-based fax server. The fax server is responsible for receiving standard fax transmissions and converting the fax documents before redirecting them to the right receiver in the DARC network. The topics discussed in this thesis are fax document routing, fax document conversion and fax server system design.

Nyckelord
DARC, fax routing, fax conversion, character recognition, document deskewing, bi-level images, image compression
# Table of Contents

1  Introduction.................................................................................. 1
   1.1  Background .............................................................................. 1
   1.2  Goal ........................................................................................ 1
   1.3  Method ..................................................................................... 1
   1.4  Limitations ............................................................................. 2
   1.5  Disposition ............................................................................ 2

2  Acronyms .................................................................................... 3

3  DARC Fax System Overview....................................................... 5
   3.1  Fax Technology ........................................................................ 5
        3.1.1  Introduction................................................................. 5
        3.1.2  Fax Protocol .................................................................. 5
   3.2  DARC ....................................................................................... 6
        3.2.1  Introduction................................................................. 6
        3.2.2  DARC Network Overview............................................. 6
        3.2.3  DARC Protocol Overview............................................. 6
   3.3  CitySurfer.................................................................................. 7
        3.3.1  Introduction................................................................. 8
        3.3.2  Sectra Paging System ................................................... 8
        3.3.3  CitySurfer Paging Client............................................... 8
   3.4  Fax Server Overview ............................................................. 9
        3.4.1  Receiver ......................................................................... 9
        3.4.2  Router .......................................................................... 9
        3.4.3  Converter .................................................................... 9
        3.4.4  Transmitter .................................................................. 10

4  Fax Document Routing ............................................................. 11
   4.1  Optical Recognition............................................................. 11
        4.1.1  Form Field Extraction.................................................. 12
        4.1.2  Mark Recognition......................................................... 16
        4.1.3  Single Character Recognition ...................................... 17
   4.2  Alternative Methods ............................................................ 22
        4.2.1  Line Based Routing....................................................... 22
        4.2.2  DTMF Interaction......................................................... 22
        4.2.3  Direct Inward Dialing................................................... 23

5  Fax Document Conversion ....................................................... 25
   5.1  Image Conversion................................................................. 25
        5.1.1  Image Noise Reduction ................................................ 25
        5.1.2  Re-Sampling ................................................................ 27
5.1.3 Generic Document Deskewing ............................................. 30
5.1.4 Grayscale Conversion .......................................................... 32
5.2 Image Compression ..................................................................... 33
5.2.1 Limitations ............................................................................ 33
5.2.2 Transmission and File System Compression ......................... 33
5.2.3 SBM Compression ................................................................. 33
5.2.4 Bi-level SBM Format............................................................. 34
5.2.5 Dictionary Compression ......................................................... 35
5.2.6 Alternative Compression Techniques ................................... 36
5.3 OCR Conversion ................................................................. 37
5.3.1 OCR System Considerations ................................................. 37

6 Fax Server System Design .......................................................... 39
6.1 Fax Server Software Design ..................................................... 39
6.1.1 Software Overview ............................................................... 39
6.1.2 Fax Receiver ........................................................................ 39
6.1.3 Image Processor ................................................................. 40
6.1.4 Additional Functionality....................................................... 40
6.2 Fax Server Maintenance Application ....................................... 41
6.3 Fax Viewer System ................................................................. 41

7 Conclusions and Further Studies .................................................. 43
7.1 Conclusions ............................................................................. 43
7.2 Further Studies ....................................................................... 43

8 References .................................................................................. 45

A CCITT Reference Images .......................................................... 47
1 Introduction

This document is a report of a Master of Science thesis in Computer Science and Engineering at the Department of Electrical Engineering at Linköping University. The task was performed in cooperation with Sectra Wireless Technologies AB.

1.1 Background

DARC - Data Radio Channel, is a technique for data broadcasting over the FM radio network. Sectra Wireless Technologies AB has developed a handheld DARC receiver known as the Sectra CitySurfer. The CitySurfer is equipped with a high-resolution display along with buttons and a joystick that allows the user to view and navigate through various types of information received over DARC.

Sectra Wireless Technologies AB has, among other services, also developed a paging system that enables personal message transmission over DARC. The background of this thesis is a wish to be able to send fax documents using the paging system and to be able to view received fax documents in the CitySurfer.

1.2 Goal

The main goal of the thesis is to evaluate the possibilities for creating a fax service for the Sectra Paging System and to, based on the results of the evaluation, create a pc-based prototype version of the system. During the work on the thesis, three sub-goals were defined in agreement with Sectra Wireless Technologies AB:

- **Present a solution to the fax document routing problem.**
  Fax documents that are received by the fax server must be redirected to the right receivers. A number of solutions to this problem are presented in Chapter 4.

- **Present a solution to the fax document conversion problem.**
  Received faxes must be converted and reformatted to be able to be displayed in the CitySurfer. A suggested solution is presented in Chapter 5.

- **Present a suggested fax server system design.**
  The functionality of the fax server should be implemented in a flexible and efficient way. Suggestions for the system design are presented in Chapter 6.

1.3 Method

The work on the thesis is divided into a number of separable steps:

The theoretical studies make up one of the most important steps of the process. Before the goal of the thesis can be determined, it is important to have a good understanding of the actual problem. A rough outline of the report is also preferably created at this time.
The practical nature of the task makes it necessary to create a development environment to use as a base for testing different approaches. For the work on this thesis, a simple fax modem communication tool has to be found, an image processing system has to be developed and the connection to the DARC network has to be handled.

The sub-goals presented in Chapter 1.2 make up a good division of the work, and these three problems can also be approached individually. The resulting solutions are finally combined into an evaluation version of the fax service.

The last step of the process is to sum up the results of the work and to finish the report.

1.4 Limitations
Sectra Wireless Technologies AB has already sold many paging-system-enabled CitySurfers. Since changes in the CitySurfer software would require these to be brought in for reprogramming, it would be desirable to be able to use current CitySurfer firmware for receiving and displaying fax documents. Due to this, the possibilities for different system solutions are significantly decreased, since only the link between the sender and the receiver can be modified.

1.5 Disposition
The outline of the rest of this document is as follows:

Chapter 2: Acronyms – Lists acronyms commonly used in this document.

Chapter 3: DARC Fax System Overview – Gives a brief introduction to the DARC fax system and the technology behind its components.


Chapter 5: Fax Document Conversion – Presents the evaluated fax document conversion methods.

Chapter 6: Fax Server System Design – Gives recommendations for how the fax server should be designed.

Chapter 7: Conclusions – Presents the conclusions drawn from the results of this thesis.
## Acronyms

<table>
<thead>
<tr>
<th>Acronym</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>API</td>
<td>Application Programming Interface</td>
</tr>
<tr>
<td>BIC</td>
<td>Block Identification Code</td>
</tr>
<tr>
<td>CCITT</td>
<td>Consultative Committee for International Telegraph and Telephone</td>
</tr>
<tr>
<td>CRC</td>
<td>Cyclic Redundancy Check</td>
</tr>
<tr>
<td>DARC</td>
<td>Data Radio Channel</td>
</tr>
<tr>
<td>DID</td>
<td>Direct Inward Dialing</td>
</tr>
<tr>
<td>Dpi</td>
<td>Dots Per Inch</td>
</tr>
<tr>
<td>DTMF</td>
<td>Dial Tone Multi Frequency</td>
</tr>
<tr>
<td>Fax</td>
<td>Facsimile</td>
</tr>
<tr>
<td>FM</td>
<td>Frequency Modulation</td>
</tr>
<tr>
<td>GPS</td>
<td>Global Positioning System</td>
</tr>
<tr>
<td>HTML</td>
<td>Hypertext Markup Language</td>
</tr>
<tr>
<td>ISDN</td>
<td>Integrated Services Digital Network</td>
</tr>
<tr>
<td>ITU-T</td>
<td>International Telecommunications Union – Telecommunications Standardization Sector</td>
</tr>
<tr>
<td>JBIG</td>
<td>Joint Bi-level Imaging Group</td>
</tr>
<tr>
<td>LCD</td>
<td>Liquid Crystal Display</td>
</tr>
<tr>
<td>MPX</td>
<td>Multiplex</td>
</tr>
<tr>
<td>NWS</td>
<td>Network Server</td>
</tr>
<tr>
<td>OCR</td>
<td>Optical Character Recognition</td>
</tr>
<tr>
<td>OMR</td>
<td>Optical Mark Recognition</td>
</tr>
<tr>
<td>PBX</td>
<td>Private Branch Exchange</td>
</tr>
<tr>
<td>RAM</td>
<td>Random Access Memory</td>
</tr>
<tr>
<td>RDS</td>
<td>Radio Data System</td>
</tr>
<tr>
<td>RLE</td>
<td>Run Length Encoding</td>
</tr>
<tr>
<td>RTOS</td>
<td>Real-Time Operating System</td>
</tr>
<tr>
<td>SBM</td>
<td>Sectra Bitmap</td>
</tr>
<tr>
<td>TCP/IP</td>
<td>Transmission Control Protocol / Internet Protocol</td>
</tr>
<tr>
<td>TIFF</td>
<td>Tag Image File Format</td>
</tr>
<tr>
<td>TSE</td>
<td>Transmitter Station Equipment</td>
</tr>
</tbody>
</table>
3 DARC Fax System Overview

This chapter gives an introduction to the DARC fax system and the technology behind it. The design of the fax system, as requested by Sectra Wireless Technologies AB, is shown in Figure 1. The fax document is sent over the telephone line from the fax machine to the fax server. The fax server processes the fax images and forwards the result into the DARC network, which is responsible for transmitting the data by air to the CitySurfer DARC receivers.

![Figure 1 - DARC fax system overview](image)

The technology behind fax transmissions is briefly presented in Chapter 3.1, followed by a DARC introduction in Chapter 3.2. The CitySurfer DARC receiver is presented in Chapter 3.3 and the central unit of the system, the fax server, is presented in Chapter 3.4.

3.1 Fax Technology

This chapter covers the basics in fax technology. Since the fax server receives fax documents using existing software solutions, the details of the fax technology and its protocols are actually not of big importance to this report. Therefore only a brief presentation will be made.

3.1.1 Introduction

A fax machine is a device that can send or receive digitalized copies of documents over a telephone line. Digitalization means that the document is divided into a grid and then scanned into a series of zeroes and ones, where the value of each point is determined by the darkness of the corresponding spot on the document. This binary representation is well suited for data communication transfer and after the data has been received, it is a simple task for the receiver to reverse the process and reprint the original image.

3.1.2 Fax Protocol

In the 80’s a third generation standard fax protocol was developed by the CCITT (later renamed to ITU-T) standards organization. The protocol is named CCITT Group3 [1] and is still in use today. The protocol supports two resolutions - low (203 by 98 dpi) and high (203 by 196 dpi) - and also defines its own compression schemes. Most fax machines and modems support the CCITT Group3 protocol but there are also newer protocols available, such as Super G3 (Group3 fax with increased transfer speed and compression, using the JBIG
compression standard) and CCITT Group4 (ISDN telephone line support with increased transfer speed and compression).

3.2 DARC
This chapter covers the basics of the technology behind DARC.

3.2.1 Introduction
DARC – Data Radio Channel, is a system for data broadcasting over the FM radio network. In the typical case, where the FM radio network already has nationwide coverage, the cost for setting up a large scale DARC system is comparably low. According to [2], the technique that is used in DARC is similar to the technique that is used in the RDS system.

3.2.2 DARC Network Overview
A DARC transmitting network typically consists of some Service Providers, a NWS - Network Server, and some transmitter stations equipped with DARC signal encoders called TSE - Transmitter Station Equipment (see Figure 2).

The Service Providers are providing the data that is to be transmitted over the network, such as financial data, news or differential GPS information.

The NWS is responsible for collecting the data and redirecting it to the transmitters, using overflow prevention mechanisms along with packet priority calculations.

The TSEs are modulating the data before adding it to the FM signal so that it can be transmitted by air to the receivers.

![Figure 2 - DARC network](image)

3.2.3 DARC Protocol Overview
The DARC protocol is specified using a multiple layer model. In this overview only the two lowest layers are presented, since they are interesting for a basic knowledge of the DARC architecture. The information in this chapter solely relies on the specifications given in [2].

3.2.3.1 Physical Layer
The physical layer is the lowest layer of the protocol and is therefore the layer that is restricting the transfer speed of the entire system. In this layer the DARC data is modulated onto a sub-carrier of 76kHz, which is in turn added to the FM MPX - multiplex signal. The MPX signal, which might also contain mono or
stereo audio as well as RDS data, is finally modulated onto a FM carrier before it is transmitted. The gross bit rate of the DARC signal is 16,000 bits/s, which is about 13.5 times the gross bit rate of the RDS system.

![Multiplex signal spectrum](image)

**Figure 3 - Multiplex signal spectrum**

### 3.2.3.2 Data Link Layer

The data link layer is the second lowest layer of the DARC protocol and handles framing and error correction. The data is first divided into blocks of 288 bits, each block starting with 16 bits BIC - Block Identification Code. Depending on block type, the rest of the block can either contain vertical parity bits (in case of a Parity Block) or 176 bits information, 14 bits CRC – Cyclic Redundancy Check, and 82 bits parity (in case of an Information Block). The CRC is used for error detection and the parity bits are used for error correction.

The blocks are finally assembled in frames. There are currently four types of frames – A0, A1, B and C, which differ in block type organization. Frame type A0 is the type that is used by the Sectra Paging System and its underlying architecture. It consists of 272 blocks, of which the first 190 blocks are information blocks and the last 82 blocks are parity blocks.

![Frame type A0](image)

**Figure 4 - Frame type A0**

The amount of error detection and correction bits in the A0 frame is considered to be enough to guarantee that all received frames are either successfully received or detected as faulty.

### 3.3 CitySurfer

This chapter gives a quick overview of the CitySurfer DARC receiver.
3.3.1 Introduction
DRT-4000, also known as the CitySurfer [3], is a handheld DARC receiver developed by Sectra Wireless Technologies AB. The CitySurfer is equipped with a LCD screen capable of displaying a four-level grayscale at a resolution of 240*160 pixels and is controlled by a micro joystick and two pushbuttons. It is running a RTOS - Real Time Operating System, and has a file system using both RAM and FLASH memory. The CitySurfer is used to receive, store and display various information transmitted over DARC.

![Figure 5 – The CitySurfer](image)

3.3.2 Sectra Paging System
Sectra Wireless Technologies AB has developed a paging system [4] for the CitySurfer that allows personal message transmission to the CitySurfer receivers. Messages can be broadcasted or sent to receivers addressed either individually or by groups. The heart of the paging system is the paging server, which is acting as a service provider for the DARC network (see Chapter 3.2.2) and is responsible for receiving messages from the paging terminals and redirecting them to the right receivers via the NWS. The system is restricted to a maximum decompressed message data size of 32,768 bytes.

![Figure 6 – Paging system network](image)

3.3.3 CitySurfer Paging Client
When the CitySurfer receives a message, the user is informed of the reception and is then able to view the message using the paging client software. The paging client currently supports text files and HTML files with images. The built-in HTML browser is limited to a subset of the HTML language and can handle images of the 4-level grayscale SBM format developed by Sectra. The conversion between large fax images and the HTML wrapped images of a paging message is one of the topics of this thesis.

Since paging messages are sent over the DARC File and Fragment protocol [5], they are automatically compressed by the underlying architecture using the `zlib`
compression software. According to [6], the message data is also compressed, using the same compression method, when stored in the flash memory of the receiver.

### 3.4 Fax Server Overview

The fax server is the central part of the portable DARC fax service system (see Figure 1 on page 5). The system model presented in this chapter is based on requests from Sectra Wireless Technologies AB and is in turn the base for all later development decisions. The server is a PC-based solution running Windows 2000. The server is connected to the telephone line using a fax-enabled modem and accesses the DARC network via TCP/IP, as described in [4]. The block diagram in Figure 7 shows a documents way through the fax server. Note that the block diagram only shows an abstract model of the system functionality. The actual system design is allowed to differ from this model as long as the functionality is the same. Each block is further explained in the following subchapters.

![Figure 7 – Block diagram of a fax document operation](image)

#### 3.4.1 Receiver

The fax server is first of all responsible for receiving incoming fax documents. In the service evaluation system, this is done using an external freeware application named StupidFax by Dan Llewellyn [7]. StupidFax is a standalone application capable of receiving fax documents over a fax modem and storing them in a number of image formats. In the evaluation system the TIFF format is used. The TIFF format is commonly used in fax programs, due to its fax compression support. In a sharp version of the system, the StupidFax application should preferably be replaced by a professional solution.

#### 3.4.2 Router

Before the fax server starts preparing the fax document for DARC transmission, the address of the receiver must be determined. The address is later used to route the document in the DARC network. In the evaluation system, the address parsing is done using optical recognition techniques. These techniques are further presented in Chapter 4, followed by a presentation of a number of alternative methods.

#### 3.4.3 Converter

The fax server is also responsible for converting the documents so they can be displayed in the Paging Client (see Chapter 3.3.3) of the CitySurfer. In the evaluation system, the fax images are retrieved in TIFF format and first have to be decoded into some intermediate form. After that, a number of image conversion techniques are used to process the images before they are converted into a format suited for display in the CitySurfer. The image conversion plays an important part of this thesis and is further presented and discussed in Chapter 5.
3.4.4 Transmitter

The transmitter part of the fax server uses TCP/IP to connect to the Paging Server in the DARC network and sends the fax images wrapped in HTML code, as a Paging Message over the Paging Terminal Protocol, see [8] and [6].
4 Fax Document Routing

This chapter covers different approaches to the fax document routing problem. The fax server is acting as a link between the sender and the receiver of each document. Therefore, the sender must be able to inform the server what to do with incoming documents and where to send them. This functionality is referred to as fax document routing and makes up a major part of this thesis. A number of different approaches to the routing problem are presented in the following chapters. The approach initially requested by Sectra Wireless Technologies AB is based on optically readable fax forms. The optical recognition method, presented in Chapter 4.1, is therefore the most thoroughly evaluated solution. Alternative solutions to the routing problem are briefly discussed in Chapter 4.2.

4.1 Optical Recognition

This chapter covers fax document routing solutions based on optical recognition. The idea behind the optical recognition approach is to have an optically readable form sent as a cover page of the fax document. The form should be holding information necessary for further management of the fax document in the fax server. The form must at least contain the address of the receiver, but it could also handle any other possible kind of settings, like compression level or OCR usage (see Chapter 5.3). Sectra Wireless Technologies AB wants to be able to use the nine-digit serial number of each CitySurfer as address identification. An example of a form of this type is shown in Figure 8.

![Figure 8 – Optically readable form example](image)

The form is supposed to either be filled in by hand or by machine in such a way that the fax server can interpret it automatically. Allowing hand written forms does of course add a more user-friendly touch to the system, but does also introduce problems in terms of interpretation difficulties. A big problem is the reliability of the system. Since optical methods are usually rather error prone, there is an increasing risk of failing to identify a receiver or, even worse, redirecting a document to the wrong receiver.

There are several professional systems available for managing forms and other optical recognition variants. These systems are typically both very complex and expensive. For the evaluation purposes of this thesis, a simple, customizable and inexpensive system is requested. Therefore it was decided to develop an own specialized fax form reading system. The system shall handle field extraction, mark recognition and to some extent also be able to recognize characters and
symbols. In the following subchapters, the results from the development of such a system are presented.

4.1.1 Form Field Extraction

This chapter presents a method for form field extraction. The predefined fax form contains fields where marks and symbols are supposed to be entered. The exact positions of these fields are known, relative to the edges of the form. The problem is that since documents are often misaligned when inserted into a fax machine, the resulting image is usually both skewed and misplaced. Different machines might even produce images of slightly different sizes. Since the extraction of the fields on the fax cover page is vital to the success of our optical methods, we are interested in the transform that straightens an incoming cover page image and normalizes its size and position. Therefore, a realignment method dealing with translation, scaling and rotation was developed. The method is presented in the following subchapters.

4.1.1.1 Alignment Marks

Before performing any modifications on the image, we need to find out which transformations that are needed to deskew the image and restore its original alignment. One way to do so is to locate certain points, called feature points, in the image and then deduce the translation, scaling and rotation transformations that are needed to restore their original positions. It should be possible to pick feature points from patterns that are part of the actual form layout, but it seems more reliable to use specially designed alignment mark symbols instead.

4.1.1.1.1 The Alignment Mark Symbol

The actual alignment mark symbol should be chosen with care. Ideally, an alignment mark symbol should have features that are scaling, translation and rotation invariant. Otherwise it risks to get distorted by the fax scanner, which in turn makes it harder to recognize.

In the solution presented here, a cross symbol is used as alignment mark. It is a simple symbol that is commonly used for alignment purposes. It has a well-defined center and it can easily be located using a corner detection filter (see [9]), even after it has been enlarged or shrunk. Its rotation variance might however be a problem. Any rotation angle other than \(\pi/2\) will tilt the symbol and thereby make it harder to recognize using a simple filter. However, this should be only a minor problem, since the maximum possible tilt angle of a scanned fax paper is limited to just a few degrees.

4.1.1.1.2 Positioning the Alignment Marks

Intuitively the alignment marks should be positioned as far away from each other as possible and in areas where no other patterns can confuse the corner detection filter. Hence, the best locations are near the corners of the form. A single mark in the upper left corner is sufficient for calculating the translation of the image. The addition of a mark in the lower right corner makes it possible to calculate the scaling factors. Since the scaling is not necessarily horizontally and vertically
proportional, a third mark must be used to be able to calculate the rotation transform of an image. This mark could for example be placed in the upper right corner of the form. Leaving the fourth corner empty leaves room for a simple extra feature: An image containing a mark in the lower left corner instead of in the upper right has most likely been accidentally turned 180° in the scanning process. A case that is easy to correct once it has been recognized.

![Figure 9 – Alignment mark positions](image)

4.1.1.1.3 Locating the Alignment Marks

Locating the alignment mark crosses in a received image can be done simply by applying a corner detection filter in the area of the supposed positions of the marks, and picking the coordinate where the filter output has its peak value. Since the actual size and line width of the cross are varying, it is easier to search for a corner than to search for the entire cross mark. There are many types of corner detection filters, with varying level of complexity. For example, the Plessey Feature Point Detector – also known as the Harris Corner Detector or Harris-Stephens Corner Detector (see [9]), can be used to find spots where the underlying image gradient is pointing in two separate directions, hence forming a corner of arbitrary color and direction.

However, the Plessey Feature Point Detector is a quite complex and calculation intensive method. In our case it proves to be sufficient with one of the simplest filters available. The filter in Figure 10 is a 3*3 filter that is stretched to 9*9 and used to find the center of the alignment mark cross symbols. As can be understood by the figure, the filter is actually constructed to find upper left 90° corners that are perfectly horizontally and vertically aligned. The actual filtering is a square sum over the color difference in every pixel between the filter and the image, and the output value is applied on the pixel at the upper left corner of the filter. A value close to zero represents a good match and to prevent false peaks, a maximum allowed difference value is used. If the maximum filter output value is above the maximum allowed difference, the filtered area is considered to be lacking an alignment mark symbol.
It would obviously be possible to use all of the four 90° rotated versions of the corner detection filter and combine their output to detect the positions of the alignment marks with a higher confidence. However, since test results clearly show that it is enough to use only one of the four versions, the use of three extra filters is considered a waste of computing time.

4.1.1.2 Normalizing

Once the locations of the alignment marks have been recognized using the method described in 4.1.1.1.3, the transformations that normalize the form size, skew and position can be calculated. We will start by calculating the angle of rotation around the center point of the form that is needed to deskew an image.

4.1.1.2.1 Rotation Angle Calculation

The center point \((c_x, c_y)\) is simply calculated as the center point between the upper left \((ul_x, ul_y)\) and the lower right \((lr_x, lr_y)\) alignment marks.

\[
\begin{align*}
    c_x &= \frac{1}{2} \left( lr_x + ul_x \right) \\
    c_y &= \frac{1}{2} \left( lr_y + ul_y \right)
\end{align*}
\]

The next step is to rotate the image around the center point so that the alignment marks in the image get aligned. With the alignment marks positioned as described in 4.1.1.2, we can either vertically align the two upper marks, or horizontally align the two rightmost marks. Theoretically, both choices should produce the same result.

However, empirical tests show that the best results are given by vertical alignment of the two upper marks. The cause of this can be that the skew introduced in the fax scanner is not always uniform. If one end of the paper is forced to move horizontally in the scanning process, the resulting horizontal alignment will be incorrect.
If we denote the center point position as \((c_x, c_y)\), the upper right and left mark positions as \((ur_x, ur_y)\) and \((ul_x, ul_y)\), and define \(v\) to be the counterclockwise rotation angle that is required to vertically align the two upper alignment marks, we get the equation:

\[
(ul_y - c_y) \cos(v) - (ul_x - c_x) \sin(v) = (ur_y - c_y) \cos(v) - (ur_x - c_x) \sin(v)
\]

\[
\Rightarrow \\
\sin(v) = \frac{ur_y - ul_y}{ur_x - ul_x}
\]

\[
\cos(v) = \frac{ur_x - ul_x}{ur_y - ul_y}
\]

\[
\Rightarrow \\
v = \arctan\left(\frac{ur_y - ul_y}{ur_x - ul_x}\right)
\]

*Figure 13 - Rotation angle calculation*

### 4.1.1.2.2 Rotation Transform

Once the rotation angle needed for deskewing the image has been calculated, the actual rotation transform can be performed. First, a new image of the same size as the old is created. For every pixel in the new image, the corresponding coordinate in the old image is calculated using the rotation around the center point. The actual pixel color can be retrieved in several ways. The calculated coordinate is usually not an exact pixel position but more probably somewhere in between four pixel positions. What we are facing here is the exact same problem that is discussed in the image up-sampling Chapter 5.1.2.2. If we choose the nearest neighbor approach presented in Chapter 5.1.2.2.1, we get an image containing lots of jaggedness and distortions. A better technique is to use the bilinear interpolation presented in Chapter 5.1.2.2.2. This technique interpolates the color values of the nearest four pixels, which results in a grayscale image of a higher image quality. Since the image of the form is used for automated optical recognition, the image quality is obviously important.

### 4.1.1.2.3 Trimming and Size Normalizing

After applying the rotation transform, the new positions of the alignment marks are calculated using the rotation angle around the center point. The next step is to trim the form by removing the borders that are outside the alignment marks. Since the two upper marks are used to align the form, we also use these two marks to trim the left, top and right side. The bottom side is trimmed using the vertical position of the lower right mark. In other words, we are extracting the sub-image between coordinates \((ul_x, ul_y)\) and \((ur_x, lr_y)\), where \(ul\) stands for the upper left mark, \(ur\) stands for the upper right mark and \(lr\) stands for the lower right mark.

Apart from the trimming, the size of the image bitmap is left unchanged. Instead of normalizing the size of the bitmap, certain positions in the image are from now on accessed using normalized coordinates relative to the width and height of
the image. This way, the size of the image is in some sense normalized while the actual image bitmap data is left unchanged.

### 4.1.1.3 Field Image Extraction

Once the image has been normalized, the fields can be accessed using the predefined normalized field coordinates, relative to the position of the alignment marks. Hence, the actual field extraction is simply a matter of copying a sub-image from the normalized form.

### 4.1.2 Mark Recognition

In the previous subchapter, a form field extraction method is presented. By applying it on a scanned fax form we are able to extract images of specified fields in the form. The next task is to be able to interpret these images.

In this chapter, we concentrate on interpreting checkbox marks. This functionality is a subsection of a technology called OMR – Optical Mark Recognition. The method presented in the next subchapter is a simple solution that was developed and used during the work on this thesis.

#### 4.1.2.1 A Simple Checkbox Mark Recognition System

A prerequisite of the method is that the field extraction method presented in Chapter 4.1.1 has been successfully applied on the form image. Otherwise, the system will try to interpret undefined areas of the form and the output will be erroneous.

The developed method is very simple. The average pixel value of the image is calculated and compared to a threshold value. The threshold value is predefined using the result from a number of checked/unchecked test cases. If the average pixel value is below (darker than) the threshold, the checkbox is recognized as checked and if the value is above (lighter than) the threshold, the checkbox is recognized as unchecked.

This may sound like a fail-safe solution but unfortunately it is not. The major problem is that when a mark in the checkbox is very small or thin (see Figure 14a), the average pixel value might be only marginally affected. One solution to the problem is to limit the size of the checkboxes. With small-sized checkboxes, marks get proportionally larger and have a higher impact on the average pixel value. On the other hand, small sized checkboxes put higher demands on the quality of the form field extraction method described earlier. Another solution could be to use empty/filled circles (see Figure 14b) instead of checkboxes. However, this type of checkbox system is not quite as intuitive as ordinary checkboxes and typically requires on-paper usage instructions.

![Figure 14 – Mark types](image)

(a) ![Checkbox](image) (b) ![Filled Circle](image)
4.1.2.2 Alternative Methods

More advanced methods for mark recognition are usually based on other features than simple pixel color averaging. For example, the methods for character recognition used in Chapter 4.1.3 can also be used to recognize marks. However, that also makes the system more sensible to the style of the marks, which is not solely a good property.

4.1.3 Single Character Recognition

This subchapter covers the work on single character recognition. The method for mark recognition that was presented previously is well suited for automatic recognition of checkbox marks on forms. It is quite possible to create a form where addresses are given using grids of checkboxes, as in Figure 15, but to create a more user-friendly form it is preferable to use a system that is also able to recognize characters, as in Figure 16.

![Figure 15 - Address entry using checkboxes](image1)

![Figure 16 - Address entry using numerical characters](image2)

The freeware character recognition development packages that are available on Internet are mainly built for recognition of machine-printed characters. The results from using these packages on handwritten characters are quite unsatisfactory and the professional development packages are simply too expensive for evaluation use. Therefore an idea of trying to create an own single handwritten character recognition system evolved. The requirements are that the recognition system is good enough to use for digit recognition in the evaluation version of the fax system, and general enough so that it can easily be exchanged for a professional solution if the fax system should be adapted for real world use in the future.
4.1.3.1 Initial Approach

The basic design of the recognition system core is a simplified version of the technique used in the OCRchie system [10]. A set of learning symbols is first given to the program. After a few modifications presented later, the image of each symbol is shrunk to a size of 5*5 pixels and stored. When a new symbol is to be read, it is first modified in the same way and the differences between the symbol and each of the learned symbols are then calculated using difference square sums. Finally, the image producing minimal difference simply qualifies as the recognized symbol. Below is the formula for calculating the difference between two images. The notation \( a(x, y) \) refer to the pixel value at position \((x, y)\) in image \(a\) and \(b(x, y)\) refer to the pixel value at position \((x, y)\) in image \(b\). Both images are of width \(w\) and height \(h\).

\[
e(a, b) = \sum_{y=1}^{h} \sum_{x=1}^{w} \left( a(x, y) - b(x, y) \right)^2
\]

Figure 17 - Difference square sum calculation

4.1.3.2 Symbol Size Normalization

The method described in the previous subchapter partially works without any extra image modifications prior to the shrinking, but it has an unacceptable error rate of about 50% even on carefully written symbols. One reason is that, in its basic construction, the method implicitly depends on the alignment and size of the symbols. A smaller and a larger symbol, at different positions, never produce a good match even if their shapes are basically the same (see Figure 18a). To solve this, a method for extracting only the rectangular area that is occupied by the symbol was developed. The minimal rectangle containing the entire symbol is extracted using the sum of pixel values on each column and row and a threshold value. For example, the horizontal starting point of the symbol is set to the point where the sum of all pixel values to the left of the point is less than 1% of the sum of all pixel values in the entire image. The threshold value serves as an edge dust remover, since a certain portion of dark pixels is allowed to fall outside the extracted area. The result is an image of the contained symbol, enlarged to a normalized size. As can be seen in Figure 18b, this method drastically improves the quality of the character recognition system.
4.1.3.3 Line Width Normalizing

Another problem with the initial approach is that since symbols with different line thickness match badly, the use of different pencils easily confuses the recognizer. The introduction of the symbol size normalization method in the previous subchapter also suffers from this problem, since the enlargement of small symbols also thickens the line width. A solution to the problem is to normalize the line thickness by, for example, using a thinning and re-thickening algorithm. The thinning algorithm is used to remove pixels so that every line of each symbol is only a single pixel wide and the re-thickening algorithm is used to thicken the symbols to a normalized line width.

4.1.3.3.1 Thinning Using Skeletonization and Pruning

There are several algorithms available for performing thinning. In the fax-server the thinning is done using a thinning algorithm presented in [11]. The algorithm is actually a skeletonization algorithm, based on the hit-and-miss transform. A number of structuring elements are translated and compared to the underlying pixels in the image. Where any of the structuring elements match a portion of the image, the center pixel of that portion is modified. The eight structuring elements used by the thinning algorithm in [11] are shown in Figure 19. 1’s in the structuring element stands for foreground (pixels that are part of a symbol) and 0’s in the structuring element stands for background. The gray areas are simply ignored. When one of these structuring elements matches a portion of the image, the center pixel is removed (thinned) and the algorithm is repeated until no more thinning can be performed.
The result from this algorithm is an image, where each symbol has been replaced with its own skeleton, as shown in Figure 20b below. The same figure also shows a problem with the thinning algorithm. Uneven edges of the symbol tend to cause small spurs on the skeleton. This is a feature of the skeletonization that is unwanted in our system. Therefore, a pruning algorithm is used to remove them. In [11], a hit-and-miss transform is presented as a solution to this problem, but instead a somewhat simpler algorithm was developed. The idea behind the algorithm is simply to remove every foreground pixel that is not a link between two or more pixel groups. To perform the connection check, the surroundings of a pixel are traversed and every foreground pixel that comes before a background pixel is counted. If the count is less than 2, the center pixel can be removed. If the count is 2 or higher, the center pixel is linking two or more pixel groups and is therefore not removed. Since infinite repetitions of the algorithm would remove every part of the symbol that is not a closed loop, there has to be a predefined number of maximum repetitions. Figure 20c below shows a symbol that has been thinned using skeletonization and pruned using the method described above.

4.1.3.3.2 Re-Thickening

There are several ways of performing re-thickening on an image. The method developed in this thesis simply expands every foreground pixel to a circle of predefined size. A disadvantage is that this may cause unwanted curvature in the edges of the symbols, but the method is considered good enough to suit the needs of the system. Figure 20d shows a symbol after it has been re-thickened using the described method.
4.1.3.4 Extended Feature Extraction

The use of size and line width normalization methods clearly improves the system, but it is still not as reliable as required for use in the evaluation version of the DARC fax system. Without further improvement, the system still has an error rate of more than 10% using numerical symbols, written by the same person using the same pen. Mainly, the symbols 9 and 4 are confused. The symbols 5 and 6 also show to be a problem and sometimes even more unlikely misinterpretations are made. Due to this, a last effort is made to try to increase the quality of the character recognition system. The idea is to extend the symbol feature extraction to involve some other features of the symbols, like perhaps the line curvature and slope of the symbols.

4.1.3.4.1 Horizontal and Vertical Line Extraction

A simple approach is to extract the horizontal and vertical lines in the symbols and compare them separately in the recognition progress. The result can be seen as an indirect slope and curvature comparison. The horizontal and vertical line extractions are well suited to be performed in between the thinning and the re-thickening of a symbol image, when the line width is one pixel and line slopes are easy to recognize. Usually, a 2-dimensional filter, such as the line detection filter, presented in [11], is used for finding lines. However, since we know that the image has just been thinned, the method can be simplified. After the thinning algorithm has been performed, a foreground pixel is set to be part of a horizontal line if a number of horizontally adjacent pixels are set to foreground and part of a vertical line if a number of vertically adjacent pixels are set to foreground. If, for example, only the closest pixels on each side are considered, three foreground pixels in a row make the center pixel part of a horizontal or vertical line. Three pixels in a row can only be found in lines with a slope of less than \( \arctan(1/2) \approx 27^\circ \). To restrict the allowed slope of the lines, more pixels must be considered. If the closest two pixels on each side is considered there has to be five pixels in a row to make the center pixel part of a horizontal or vertical line. Five pixels in a row can only be found in lines with a slope of less than \( \arctan(1/4) \approx 14^\circ \). After the horizontal and vertical line extraction has been performed, re-thickening is used on the original image as well as on the two extracted images to reproduce the connectivity and thickness of the lines. Figure 21 shows an original image (a) and its horizontal (b) and vertical (c) extracted images. The images have been re-thickened after the extractions.

![Figure 21](image)

*Figure 21 – Horizontal (b) and vertical (c) extraction*
4.1.3.5 Final System

The use of the horizontal and vertical line extraction methods shows to be yet another clear improvement. When trained and tested on different sets of numerical symbols, the system now produces an error rate of less than 10%. Relative to the results of the earlier versions of the system, these figures are actually quite good. However, an error rate of 10% is not acceptable in a system where reliability is a vital aspect.

The character recognition system has so far been an interesting sidetrack of the thesis but considering the amount of work and time needed to further improve the system, the development of the recognizer has to be terminated. The system is currently good enough for evaluation purposes, and might even be acceptable for professional use if digits are written with great care or only computer printed symbols on prepared fax forms are used.

4.2 Alternative Methods

This chapter presents some alternative solutions to the fax document routing problem. In Chapter 4.1, optical methods for automated fax form reading has been presented. In this chapter, a number of alternative methods are discussed. The suggested solutions are all commonly used fax routing techniques, which can, for example, be found in the FAXserve system presented in [12].

4.2.1 Line Based Routing

Line based routing is in some ways the simplest solution to the routing problem. It means that every receiver address has its own telephone line connected to the fax server. Hence, all fax documents received over a specific telephone line is forwarded to the same destination.

The main advantage of the line based routing strategy is its simplicity. Since all documents received over a specific telephone line are treated the same way, there is no need for additional routing information. This eliminates the need for extra user interaction, but also limits the options available to the sender.

The major drawback is of course the inflexibility of the system. While the strategy is excellent for a very limited small-scale system or a larger broadcasting system, it is extremely unsuited for use with many individually addressable receivers.

4.2.2 DTMF Interaction

DTMF – Dial Tone Multi Frequency, is a signal formed by the sum of two sinusoids, generated by pressing the keys on a DTMF enabled phone. The DTMF tones are usually used by the switchboard to recognize dialed numbers, but they can also be used to send key-press information to the other side of the line after a connection has been established.

When using this feature in a fax routing approach, the user first dials the number to the fax server and then waits for it to pick up. After a connection has been established, the system can either simply wait for the user to enter the address of
the receiver, or run a more complex sound-based menu system using prerecorded voice messages. The user could, for example, be asked to press certain key combinations for different functionality and get feedback based on the choices.

The DTMF control system is comparably easy to build and the only hardware requirement is that the modem connected to the fax server must be voice enabled. There are also several professional development tools available to simplify the construction of a DTMF based menu system.

Unfortunately, even this solution has its flaws. The problem lies in the way fax machines are used. Professional fax machines are usually equipped with a call queue system, where outgoing fax documents are put on hold until the line is free and a call connection can be established. When using the DTMF menu system, the sender is required to be present when the fax server picks up. Moreover, DTMF based menu systems are usually not very user-friendly due to the time-consuming audio interaction.

4.2.3 Direct Inward Dialing

DID - Direct Inward Dialing is a method where the main switchboard at the telephone company forwards the last digits of the phone number to a PBX - Private Branch Exchange, at the end of the line. This functionality is used at many companies, where the first digits in the phone number lead to a PBX at the company’s central office and the PBX then uses the extension to route calls within the company.

When using this method for fax document routing, the first digits in the phone number lead to a fax server equipped with a DID interface board and the last digits identify the receiver of the document. A DID system typically supports up to 4 extension digits, enabling 10,000 individual addresses.

The DID solution is the only alternative that requires highly specialized hardware. The DID interface board is quite expensive and must be installed in every fax server. This complicates the whole system installation and thereby also increases the cost for setting up a new system. Another drawback is the need for address administration. As the DID extension is restricted to a maximum of four digits, the number cannot be directly mapped to the 9-digit serial number of the CitySurfer DARC receiver. Instead a separate mapping database must be set up and administrated to keep track of which extension is associated with which receiver. This also requires development of special administration software solutions.
5 Fax Document Conversion

This chapter covers the work on fax document conversion. Before an incoming document is redirected to the right receiver it has to be converted into a form suited for display in the CitySurfer. This involves a number of problems that have to be solved but also leaves room for additional features. The main task is to minimize the size of the image while retaining high quality. Reducing the image data size decreases the risk for network congestions and also lowers the risk for data loss. It also reduces the amount of memory required in the CitySurfer.

5.1 Image Conversion

The following chapters present the results from the work on the image conversion methods.

5.1.1 Image Noise Reduction

This chapter covers image noise reduction techniques. When the fax machine scans a document, a lot of noise is introduced. Dust and speckles in the image make text harder to read and even the optical methods discussed in Chapter 4.1 tend to be sensitive to these types of disturbances. Due to this, it is interesting to look at different types of noise reduction. The following subchapters discuss the use of low pass filters and a median filter.

5.1.1.1 Low Pass Filter

A simple way of eliminating much of the high frequency noise is to apply a low pass filter on the image. When high frequencies get suppressed, sharp edges in the image get smeared out. As can be seen in Figure 22, the resulting image is less noisy but much blurrier.

![Figure 22 - A noisy image before (a) and after (b) low pass filtering](image)

The low pass filter that is used in this thesis is a simple mean filter, presented in [11]. The idea of the mean filter is to replace every pixel with the mean value of the surrounding pixels and itself. This is done using the 2-dimensional filter kernel showed in Figure 23. This kernel is the same as the one presented in [11].
A kernel is a weight value matrix that is applied on every pixel in the image. If we denote the value at position \((a,b)\) in a kernel of size 3*3 as \(K(a,b)\), the pixel color value at position \((x,y)\) in the old image as \(I_o(x, y)\) and a pixel at the same position in the new image as \(I_n(x,y)\), the filter calculation can be expressed using the formula in Figure 24.

\[
I_n(x, y) = \sum_{a=-1}^{1} \sum_{b=-1}^{1} K(a,b) \cdot I_o(x + a, y + b)
\]

Figure 24 - 3*3 kernel filter calculation

5.1.1.2 Low Pass Filter with Threshold Level

When filtering fax documents, the source image is a usually a bi-level (black and white) image and therefore it might be desirable to also retain a bi-level image as result. To achieve this, a threshold level is experimentally introduced on the output of the low pass filter. If the kernel calculation results in a value above or equal to the threshold, the pixel is set to high (white) and if the result is below the threshold, the pixel is set to low (black). If a threshold level of 5/9 of maximum color level is used on a bi-level image, we get the same effect as when using the median filter described in Chapter 5.1.1.3. The result from using the threshold level on the output of the low pass filter can be seen in Figure 25.

Figure 25 – A noisy image before (a) and after (b) low pass filtering using a threshold level

5.1.1.3 Median Filter

The third evaluated filter is the median filter presented in [11]. The median filter is used to remove dots of dust in the image while still preserving good detail. The algorithm is simple. The color values of a pixel and its eight neighbors are sorted and the center pixel color value is replaced by the middle value in the sorted list. When using this filter on a bi-level image, the algorithm can be simplified into counting all black pixels in the 3*3 area over the center pixel. If
there are less than 5 black pixels, the center pixel is set to white. Otherwise it is set to black. As mentioned in Chapter 5.1.1.2, the median filter effect on a bi-level image is the same as when using the mean filter and a threshold level of 5/9 – see Figure 25. Due to this, only the mean filter with an optional threshold level, as presented in Chapter 5.1.1.2, is available in the final fax server software.

5.1.2 Re-Sampling

One of the most important parts of the fax image conversion is the re-sampling. To better fit for display in the comparably small sized display of the CitySurfer, the fax documents have to be shrunk. This is done using a down-sampling method that is presented in Chapter 5.1.2.1. In some situations, like the normalization part of the Character Recognition engine presented in Chapter 4.1.1.2, images need to be enlarged using an up-sampling method that is described in Chapter 5.1.2.2. The interpolation method used for up-sampling is also used by the rotation transform in the two deskew algorithms presented in Chapter 4.1.1.2.2 and Chapter 5.1.3.

5.1.2.1 Down-Sampling

Down-sampling a digital image is the same as shrinking it. The image consists of a 2-dimensional array of pixel values and in the down-sampling process these pixel values are mapped onto an array of smaller size. The perhaps most intuitive way of doing this is to decimate the original image (see [13]), or in other words simply remove a number of rows and columns to retain the new image. As is easily understood, decimating is not a preferable solution since it threatens to remove lines and other features vital to the contents of the image. Theoretically, this behavior is caused by aliasing in the frequency domain and is a result from sampling with a too low frequency (see [13]). Figure 26a shows the sixth CCITT reference image (see Appendix A) down-sampled using decimation only. As can be seen in the figure, a lot of detail in the image is lost.

To get rid of the aliasing, we should rely on the sampling theorem (see [13]) and first apply a low pass filter on the image using a cut-off frequency that is at most half the frequency that we are intending to use for sampling. Figure 26b shows the result from using such a low pass filter prior to the actual down-sampling.

An ideal low pass filter corresponds to a $sinc$ function (see [13]) in the spatial domain (see Figure 27a) but in reality, that kind of filter is not feasible, since it is
infinite and way too precise for the needs of the fax server image conversion system. Instead, an approximation in the form of a simple box filter, presented in [14], is used. The box filter can be seen in Figure 27b.

![Figure 27 – An ideal low pass filter (a) and a box filter approximation (b)]

5.1.2.1.1 Box Filter

The box filter is actually an averaging filter (see [14]). As the name implies, the filter has the form of a box in the spatial domain. The filtering and the following decimation are preferably performed in a single step and the result can be seen as if each pixel in the new image is calculated as the average of the pixels it covers in the old image. The result is a low pass filtered and decimated version of the old image. The difference between the ideal low pass filter and the box filter that can be seen in Figure 27 introduces some new aliasing artifacts in the resulting image. However, as visual analysis of a number of resulting images show clearly satisfactory results, the filter is considered good enough to be used in the fax server software.

5.1.2.1.2 Tent Filter

To increase the quality of the down-sampling method, the tent filter shown in Figure 28 can be used instead of the box filter. The tent filter, presented in [14], is a down-sampling variant of the bilinear interpolation filter that is used in Chapter 5.1.2.2.2 and is a better approximation to the ideal low pass filter that is still relatively easy to implement. Since the results from using the box filter are considered good enough, the tent filter is not used in the fax server. It could however be interesting for future improvement.

![Figure 28 - Tent filter approximation]

5.1.2.2 Up-Sampling

Up-sampling a digital image is the same as enlarging it. The problem is that we want to expand the image to contain more detail than is available. The same problem arises when a digital image is rotated, as described in Chapter 4.1.1.2.2 and Chapter 5.1.3. The missing details of the image have to be extracted from the information we possess and this can be done in several ways. The images in Figure 29 show the results from up-samplings using the methods called Nearest Neighbor (b), Bilinear Interpolation (c) and Bicubic Interpolation (d). The result from the bicubic interpolation method is shown for comparison reasons only. It
is not used in the fax server since the increased quality is not considered worth the increased computing time and the time needed for implementation.

![Image](figure29.png)

*Figure 29 - Up-sampling using Nearest Neighbor (b), Bilinear Interpolation (c) and Bicubic Interpolation (d). The original is shown in (a).*

### 5.1.2.2.1 Nearest Neighbor

The nearest neighbor method (see [15]) simply means that every pixel in the new image is set to the same value as the nearest available pixel in the old image. The method is a common method that is very fast but produces images with relatively low quality. Jaggedness in the image is introduced and symbol proportions might be affected. The situation where this method can be preferable is when a bi-level (black and white) image is requested as the result.

### 5.1.2.2 Bilinear Interpolation

Bilinear interpolation (see [15]) is another common method that usually produces a better result than the nearest neighbor approach. The basic idea is to calculate the missing pixels as combinations of the surrounding pixels, with more weight on pixels that are closer. As can be seen in Figure 30, the new pixel value on position \((x, y)\) is actually calculated using the areas to the nearest four pixels as weight values for the corresponding pixel color values. The upper left pixel value is weighted with \((x-a)*(y-b)\), the upper right with \((a+1-x)*(y-b)\), the lower left with \((x-a)*(b+1-y)\) and the lower right with \((a+1-x)*(b+1-y)\). The result is an image with soft color transitions, as can be seen in Figure 29c. The Bilinear Interpolation method was considered to give the best balance between speed and quality and is the method used for enlarging and rotating images in the fax server.

![Image](figure30.png)

*Figure 30 - Bilinear Interpolation*

### 5.1.2.3 Combined Up- And Down-Sampling

Sometimes, there is a need for down-sampling an image in one direction while up-sampling it in another. Since this is an uncommon situation, the easiest
A Portable DARC Fax Service

approach is used. The image is simply first down-scaled in one direction, using the box filter method described in Chapter 5.1.2.1.1 and then up-sampled in the other direction, using the bilinear interpolation method described in Chapter 5.1.2.2.2. There are probably better solutions to this problem, but since the combined up- and down-sampling will most probably never be used in the fax server software, there is no obvious need for further improvement of the method.

5.1.3 Generic Document Deskewing

This chapter presents a generic method for document deskewing that was developed and evaluated during the work on the thesis. As mentioned in Chapter 4.1.1, documents that are scanned in a fax machine often get skewed. The deskewing method presented in Chapter 4.1.1.1 and 4.1.1.2 is actually superior in terms of speed compared to the one that will be presented here. The problem is that the method is dependant on the existence of alignment marks in the image and thus only works on specially designed fax form pages.

Before the development of the method begun, the idea was that deskewing all pages of an incoming fax document would not only improve the image quality and make the images fit better in the display of the CitySurfer but also produce images that could result in higher compression rates. The reason for this assumption was the fact that the bitmaps in the CitySurfer are compressed using run length encoding (see Chapter 5.2.3). Since a run of pixels of the same color compresses better than a run of pixels of different color, a deskewed image would typically compress better than a skewed image, due to its larger amount of horizontal lines. Unfortunately, the test results that were backing up the theory showed to be erroneous and the real compression ratio improvement is actually almost negligible. Despite that, the generic deskewing algorithm performs so well that it is still available in the fax server software and is used to deskew documents to better fit in the display of the CitySurfer.

5.1.3.1 Theory

The basic idea behind the deskewing algorithm is that since most pages contain strong horizontal characteristics, like horizontal text or images with horizontal lines and edges, these horizontal characteristics could be used to deskew the image. The goal is to find the rotation angle that maximizes the horizontal characteristics of the image. The interesting question is how these characteristics are measured. The idea is to calculate the square sum of the sum of pixel values on each row and let a high value stand for a high horizontality. This approach is initially not based on any known method but as the development of the method continued, the approach showed to be very similar to the deskewing method presented in the OCRchie system [10].

In the expression below, the horizontality measurement, \( h \), is calculated as a function over the angle \( v \) and \( p_r(x,y) \) stands for the darkness on position \( (x,y) \) in the image \( p \) rotated by the angle \( v \). Since it is the darkness of the pixels that are measured, dark colors are defined as high while light colors are defined as low.
\[ h(v) = \sum_{y} \left( \sum_{x} p_{v}(x, y) \right)^2 \]

*Figure 31 – Horizontal characteristics calculation*

The square in the expression results in a higher horizontality value for images where black pixels are distributed onto a small number of horizontal rows or, in other words, where as many dark lines or rows of text are as horizontal as possible.

Now that a way of calculating the horizontal characteristics is defined, we need to be able to find the rotation angle that maximizes the expression. The horizontal characteristics calculation is time consuming so the number of iterations should preferably be held to a minimum. There is also a need for restricting the possible angle of rotation to a few degrees, since the algorithm should not be allowed to run amok when facing a document with low horizontality. With these basic rules in mind a simple algorithm was developed.

### 5.1.3.2 The Algorithm

First of all, a rough assumption must be made. The image is simply assumed to produce a horizontal characteristics value function that has a symmetric maximum somewhere in between \( v = \pm a \). To find this maximum, initialize a variable \( v \) to 0 and \( a \) to an appropriate interval value and perform the following steps a predefined number of times:

- Calculate a horizontal characteristics value, \( h_1 \), over the original image rotated by an angle \( v - a/2 \)
- Calculate a horizontal characteristics value, \( h_2 \), over the original image rotated by an angle \( v + a/2 \)
- If \( h_1 > h_2 \) set \( v = v - a/2 \)
- Else set \( v = v + a/2 \)
- Set \( a = a/2 \)
- Repeat

If the assumptions are met, an infinite number of repetitions will set \( v \) to the exact rotation angle needed to get the maximum horizontality value. Since the interval is halved after every iteration loop, \( n \) iterations will give a precision of \( \pm a/2^n \).

The values in Table 1 show the results from testing the method on the CCITT reference images (see Appendix A). The algorithm is initialized using \( a = 0.15 \) radians \( \approx 8.6^\circ \) and \( n = 5 \), giving a maximum rotation angle of \( 8.6^\circ \) and a precision of about \( \pm 0.27^\circ \), which should be enough for most purposes. As can be seen in the table, the results are quite satisfying. The numbers in the column headers are the angles at which each image was first rotated and the numbers in the columns are the angles returned by the algorithm when trying to deskew the
images. As can be seen in the table, the second image differs a little from the rest. The cause of this is that the image shows a sketch that is not perfectly horizontal (see CCITT2 in Appendix A). The similarities in the errors of the rest of the numbers are caused by the very nature of the algorithm. If the algorithm is allowed to iterate infinitely, these numbers will converge and then eventually slightly differ from each other.

<table>
<thead>
<tr>
<th></th>
<th>-5.7°</th>
<th>-2.9°</th>
<th>0°</th>
<th>2.9°</th>
<th>5.7°</th>
</tr>
</thead>
<tbody>
<tr>
<td>CCITT1</td>
<td>5.625°</td>
<td>2.946°</td>
<td>-0.268°</td>
<td>-2.946°</td>
<td>-5.625°</td>
</tr>
<tr>
<td>CCITT2</td>
<td>5.089°</td>
<td>3.482°</td>
<td>-0.804°</td>
<td>-2.411°</td>
<td>-5.089°</td>
</tr>
<tr>
<td>CCITT3</td>
<td>5.625°</td>
<td>2.946°</td>
<td>0.268°</td>
<td>-2.946°</td>
<td>-5.625°</td>
</tr>
<tr>
<td>CCITT4</td>
<td>5.625°</td>
<td>2.946°</td>
<td>-0.268°</td>
<td>-2.946°</td>
<td>-5.625°</td>
</tr>
<tr>
<td>CCITT5</td>
<td>5.625°</td>
<td>2.946°</td>
<td>-0.268°</td>
<td>-2.946°</td>
<td>-5.625°</td>
</tr>
<tr>
<td>CCITT6</td>
<td>5.625°</td>
<td>2.946°</td>
<td>0.268°</td>
<td>-2.946°</td>
<td>-5.625°</td>
</tr>
<tr>
<td>CCITT7</td>
<td>5.625°</td>
<td>2.946°</td>
<td>0.268°</td>
<td>-2.946°</td>
<td>-5.625°</td>
</tr>
<tr>
<td>CCITT8</td>
<td>5.625°</td>
<td>2.946°</td>
<td>0.268°</td>
<td>-2.946°</td>
<td>-5.625°</td>
</tr>
</tbody>
</table>

Table 1 - Deskewing algorithm results

5.1.4 Grayscale Conversion

The CitySurfer is capable of displaying images in a four level grayscale, but fax documents are usually bi-level. This adds an extra possibility to the conversion methods. To prevent truncation errors, some of the image conversion methods, e.g. rotation and resampling, are working with a full grayscale precision of 8-bits/pixel. This led to the decision to use full grayscale precision in all image conversion stages and not truncate the precision until the final conversion step.

Separating the color level truncation from the rest of the conversion steps also introduces new possibilities in customizing the final color level output. Using the extra color levels significantly increases the perceptual quality of the images, makes texts easier to read and images look smoother. As stated in Chapter 5.2.4, the loss in compression ratio, caused by the introduction of an extra information bit for every pixel, is considered justified by the improved image quality.

The conversion from 256 to 4 level grayscale was initially done using the two most significant bits in the 256 level grayscale byte. The result of this is that the values 0-63 map to black, 64-127 map to dark gray, 128-191 map to light gray and 192-255 map to white, as can be seen in graph b in Figure 32. However, visual inspections of the results show that since light pixels in the CitySurfer are affected by the typical green-yellowish LCD background color, it is by no means certain that such a direct mapping is the best choice. Hence, the mapping levels were instead experimentally chosen using a sequence of grayscale images with different mapping levels. The final choice is to map the values 0-63 to black, 64-191 to dark gray, 192-254 to light gray and 255 to white, as can be seen in graph c in Figure 32.
5.2 Image Compression

This chapter covers image compression.

5.2.1 Limitations

As stated in Chapter 1.4, it would be preferable if the fax service could be designed to work with the current CitySurfer firmware. This puts a heavy restriction on the flexibility of the image compression solutions. Using the current CitySurfer firmware, a fax document must be presented in the form of a HTML document with text and images in SBM – Sectra Bitmap format. In the following chapters, the SBM format is presented along with the results from trying to improve its usage using a directory compression algorithm. There is also a discussion about what could be earned, in terms of image compression ratio, if a firmware replacement was to be approved.

5.2.2 Transmission and File System Compression

As mentioned in Chapter 3.3.3, the paging message transmission and the internal image storage in the CitySurfer are compressed using the zlib compression software.

The zlib software uses the LZ77 algorithm to compress the representation of repeated data patterns and Huffman coding for compressing common data (see [18]). Since the zlib compression is always applied on image files that are sent over DARC or stored in the CitySurfer, it is important that the zlib compression is also considered when comparing different image compression techniques.

5.2.3 SBM Compression

This chapter presents the file format and compression technique used in the standard quad-level (2-bit) version of the SBM files. The information presented here is based on the file format specifications in [17]. The SBM format also defines a bi-level version that has not been implemented in the CitySurfer. The bi-level version is briefly discussed in Chapter 5.2.4.

Another possibility is to interleave pixels of different tones of gray to produce dithering in the image. Dithering is used to simulate grayscales that is not available as a direct color. However, as the use of dithering would complicate the run length encoding compression (see Chapter 5.2.3) and thereby produce larger image files, the method was discarded before it was even tested.
The SBM format uses RLE compression to reduce file sizes. This means that a run of the same value is represented by an indicator of run length and value. From this follows that horizontally large areas of the same color are well compressed, while areas with pixels of interleaved colors are not compressed at all.

The SBM file format consists of a 5 bytes header, containing type and size information, followed by a number of pixel color data bytes. The following figure shows the header structure of the file format. The width and height of the image is given in little-endian byte order (least significant byte first).

<table>
<thead>
<tr>
<th>Byte 1</th>
<th>Byte 2</th>
<th>Byte 3</th>
<th>Byte 4</th>
<th>Byte 5</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Image Width</td>
<td>Image Height</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

*Figure 33 - SBM header, 4 level grayscale version*

The pixels are seen as a one-dimensional array of color values, starting in the upper left corner, going from left to right, downwards one row at a time. There are two types of data bytes. The bitmap coded byte type contains pixel data for three individual pixels. Its bit structure is shown in the figure below.

<table>
<thead>
<tr>
<th>Bit 8</th>
<th>Bit 7</th>
<th>Bit 6</th>
<th>Bit 5</th>
<th>Bit 4</th>
<th>Bit 3</th>
<th>Bit 2</th>
<th>Bit 1</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td></td>
<td>Pixel 1 Color</td>
<td>Pixel 2 Color</td>
<td>Pixel 3 Color</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

*Figure 34 - Bitmap coded byte type*

The RLE coded byte type contains pixel data for a run of up to 31 pixels of the same color. Its bit structure is shown in the figure below.

<table>
<thead>
<tr>
<th>Bit 8</th>
<th>Bit 7</th>
<th>Bit 6</th>
<th>Bit 5</th>
<th>Bit 4</th>
<th>Bit 3</th>
<th>Bit 2</th>
<th>Bit 1</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Run Length</td>
<td>Pixel Color</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

*Figure 35 - RLE coded byte type*

### 5.2.4 Bi-level SBM Format

The SBM format presented in [17] also specifies a bi-level (1-bit) image format that has not been implemented in the CitySurfer. The bi-level version has the value 2 in the first byte of the header. Its bitmap coded byte type contains pixel color for 7 individual pixels and the RLE coded byte specifies a run of up to 63 pixels of the same color (black or white). One interesting question is what could be gained from implementing and using this format for displaying bi-level fax images.

Theoretically, an image containing only interleaved pixel colors would in the 1-bit format consume $3/7$ of the space needed for the 2-bit format. An image containing only one color would in the 1-bit format consume $1/2$ of the space needed for the 2-bit format. In the two first columns of Table 2, the CCITT reference images have been compressed using 1-bit and 2-bit format respectively. The results support the conclusion that the 1-bit format is, roughly, resulting in files half the size of those created using the 2-bit format. However, as mentioned in Chapter 3.3.3, both the transmission and the storage in the CitySurfer are further compressed using the zlib compression algorithms. When
this compression is applied on images of the two format versions, the figures drastically change. As seen in the last two columns of Table 2, the use of 1-bit SBM format only gives a difference of about 6% in file size after the \textit{zlib} compression has been applied.

The loss of image quality and the inflexibility of the 1-bit format are simply considered a too high price for only a minor change in final file size. The result from this evaluation is a recommendation not to implement or use the 1-bit version of the SBM format at all.

\begin{table}
\centering
\begin{tabular}{|l|c|c|c|c|}
\hline
 & 1-Bit SBM & 2-Bit SBM & 1-Bit SBM & 2-Bit SBM \\
 & & & \textit{zlib} & \textit{zlib} \\
\hline
CCITT1 & 18.5\% & 32.5\% & 5.2\% & 5.5\% \\
CCITT2 & 16.3\% & 28.9\% & 4.6\% & 4.9\% \\
CCITT3 & 22.7\% & 38.6\% & 8.2\% & 8.9\% \\
CCITT4 & 34.1\% & 53.1\% & 17.8\% & 18.8\% \\
CCITT5 & 23.6\% & 39.5\% & 9.7\% & 10.4\% \\
CCITT6 & 19.9\% & 32.9\% & 5.5\% & 5.9\% \\
CCITT7 & 30.9\% & 48.4\% & 19.3\% & 20.7\% \\
CCITT8 & 18.5\% & 31.6\% & 7.1\% & 7.7\% \\
Average & 23.1\% & 38.2\% & 9.7\% & 10.3\% \\
\hline
\end{tabular}
\caption{Comparison between 2-bit and 1-bit SBM format, before and after \textit{zlib} compression has been applied}
\end{table}

\subsection*{5.2.5 Dictionary Compression}
As mentioned in Chapter 3.3.3, the paging client in the CitySurfer is restricted to the use of HTML image tags and SBM images. When working on the formatting issue, an idea arose of how to use the available functionality to improve the compression ratio and develop a method for \textit{Dictionary Compression}. This type of compression means that patterns appearing more than once in an image are moved into a “dictionary”. Instead of coding the same pattern more than once, the image contains several references to the same pattern in the dictionary.

When using the SBM files and the HTML subset of the CitySurfer, we are limited to display rectangular images in an aligned grid. Therefore we are also restricted to finding matching rectangular areas instead of freely extracted matching patterns. In the developed algorithm, the image is first divided into a grid of equally sized squares. Smaller squares mean a higher possibility of finding matching squares, but also increases the overhead needed for file headers and image references.

Two squares are considered equal if the square sum of the pixel color differences is below a predefined value. In Figure 36, the fifth CCITT reference image has been resized to 1:4 and encoded using the presented algorithm with a square size of 8*8 pixels. To the left is the original image and to the right is an image where all squares that will be replaced by other squares have been inverted. In other words, only the non-inverted squares will be stored and used to reproduce the original image. As can be seen, these squares make up only a minor portion - in this case close to 30% - of the total image area. So far, this dictionary
compression algorithm seems to work flawlessly. The interesting question is what effect it has on the final file size.

Considering the effective reduction of squares in the example above, the numbers of the final file size is quite a big disappointment. The original image is compressed into a file size of 13,758 bytes. The dictionary image file sizes sums up to 13,083 bytes, which is merely a 1% decrease in file size. These results are mainly caused by the increased data overhead. First of all there has to be an extra header of 5 bytes for every image in the dictionary. In the example above, where there are 1,087 squares in the dictionary, the headers sum up to 5,435 bytes, which is more than 40% of the total data space used. Then there is the problem with the run length encoding. Every time the square segmentation breaks a possible run of the same color we lose compression possibilities.

However, the main problem of this method has not been presented yet. The images in the dictionary have to be referenced and combined in some way to produce the final image. The only possible way to do this, using the current CitySurfer firmware version, is to use HTML files containing image tags:

```
<IMG SRC="001.SBM">
```

In the example presented earlier in this chapter, there are 3,942 squares, resulting in a HTML file of about 78kB. Due to its repetitive nature, the HTML file can be compressed to about 1% of its original size, but the restriction to a maximum decompressed message size of 32kB, mentioned in Chapter 3.3.2, unfortunately still makes this compression method unusable.

### 5.2.6 Alternative Compression Techniques

This chapter presents what could be gained in terms of image compression if a reprogramming of the CitySurfer should be allowed. As this is not in line with
what was initially requested, these alternative compression techniques will be just briefly evaluated.

In Table 3, the compression ratio of the 2-bit SBM format further compressed using the zlib software [18], is compared to a number of market-leading compression standards for bi-level images, such as G4, JBIG and variants of JBIG2. The figures rely on the information on the compression ratio on the CCITT reference fax images, given by [19]. G4 and JBIG are used in some of the newer fax machines and JBIG2 is a newly developed and still emerging standard for superior compression of bi-level images. Please note that the numbers for the SBM format indicates the compression ratio after the zlib compression has been applied. It may seem an unfair comparison, but since the other compression techniques already produce high compressions, the zlib compression will not be able to do much difference in terms of file sizes.

<table>
<thead>
<tr>
<th>SBM zlib</th>
<th>G4</th>
<th>JBIG</th>
<th>JBIG2 MQ</th>
<th>JBIG2 CSM/G4</th>
<th>JBIG2 CSM/MQ</th>
</tr>
</thead>
<tbody>
<tr>
<td>CCITT1</td>
<td>5.5%</td>
<td>3.3%</td>
<td>2.5%</td>
<td>2.5%</td>
<td>1.4%</td>
</tr>
<tr>
<td>CCITT2</td>
<td>4.9%</td>
<td>2.1%</td>
<td>1.6%</td>
<td>1.5%</td>
<td>2.1%</td>
</tr>
<tr>
<td>CCITT3</td>
<td>8.9%</td>
<td>5.1%</td>
<td>4.0%</td>
<td>3.8%</td>
<td>2.8%</td>
</tr>
<tr>
<td>CCITT4</td>
<td>18.8%</td>
<td>12.7%</td>
<td>9.5%</td>
<td>9.1%</td>
<td>3.2%</td>
</tr>
<tr>
<td>CCITT5</td>
<td>10.4%</td>
<td>5.8%</td>
<td>4.6%</td>
<td>4.4%</td>
<td>2.5%</td>
</tr>
<tr>
<td>CCITT6</td>
<td>5.9%</td>
<td>3.1%</td>
<td>2.3%</td>
<td>2.3%</td>
<td>3.0%</td>
</tr>
<tr>
<td>CCITT7</td>
<td>20.7%</td>
<td>13.1%</td>
<td>10.3%</td>
<td>10.1%</td>
<td>8.3%</td>
</tr>
<tr>
<td>CCITT8</td>
<td>7.7%</td>
<td>3.6%</td>
<td>2.8%</td>
<td>2.5%</td>
<td>3.6%</td>
</tr>
<tr>
<td>Average</td>
<td>10.3%</td>
<td>6.1%</td>
<td>4.7%</td>
<td>4.5%</td>
<td>3.4%</td>
</tr>
</tbody>
</table>

Table 3 - Compression ratio comparison

The presented figures indicate that the SBM format in combination with the zlib compression, used for transmission and storage in the CitySurfer, is actually quite efficient. The much more complex compression standard JBIG2 in its most aggressive version does not produce much better than about three times smaller files. The benefits from integrating a new image decoder in the CitySurfer are therefore probably not worth the effort.

5.3 OCR Conversion

This subchapter presents the results from a brief evaluation of an OCR - Optical Character Recognition, approach. In Chapter 4.1.3, a system for recognition of handwritten characters is presented. What is discussed in this chapter is a similar, but extended, method that is able to recognize machine printed symbols, words and sentences. The idea is to be able to extract text from an incoming document and to send only the text representation to the receiver. The gain is decreased file sizes and increased readability on the receiver side.

5.3.1 OCR System Considerations

One possibility would of course be to further develop the single character recognition system presented in Chapter 4.1.3 to allow for full text recognition, but since the quality of the system is already questionable, the wisest choice is probably to integrate an external OCR package instead. Based on this decision,
the problem is reduced to a matter of finding a Windows-based OCR package, preferably available as freeware. It can, for example, be sufficient to use a stand-alone application that takes a scanned document image as input and is capable of storing the result in a text file without user interaction.

The task of finding an OCR system matching all the given criteria is extremely difficult. Many open source projects, like Clara OCR, are only available for Unix/Linux. Other projects, such as OCRchie and JOCR/GOCR, are simply too unreliable. High quality products, such as SimpleOCR (formerly WOCAR), typically don’t have freely available APIs or source codes and don’t seem to be able to run automated.

If a sharp version of the fax server system should be implemented in the future, a professional OCR system would probably be used. Therefore, it was simply decided to exclude the OCR functionality from the evaluation system. However, the system can easily be expanded with professional OCR software if desired in the future.
6 Fax Server System Design

This chapter will present recommendations for how a sharp version of the fax server system could be designed. The recommendations are based on the results from the evaluations of the test system.

6.1 Fax Server Software Design

This chapter contains recommendations on the fax server software design. The recommendations are based on the system specifications initially requested by Sectra Wireless Technologies AB, which means that the system consists of a PC-based solution, running in a Windows 2000 environment. A standard fax enabled modem is used for receiving fax transmissions and a system for optically readable forms is used for routing incoming documents.

6.1.1 Software Overview

To ensure that the fax server is always ready to pick up on incoming calls, the software constantly has to monitor the fax modem. Therefore it is intuitively a good solution to implement the system as a service and let the fax modem communication run as its main thread. This approach requires the image processing and paging server communication to be moved into a separate unit. If the image processing part is very time-consuming it could also be useful to use a queue system for the images, as shown in Figure 38.

![Figure 38 - Fax server block diagram](image)

6.1.2 Fax Receiver

As mentioned in Chapter 3.4.1, the evaluation version of the system is using an external freeware application called StupidFax to receive fax documents. In a sharp version of the system, the StupidFax software should preferably be exchanged for a professional solution that is able to communicate directly with the image-processing unit of the system. StupidFax is developed using an API available as a part of Async Professional from Turbo Power. Hence, using the same API to implement a fax receiver service would guarantee the functionality
of the system. The same package also contains tools for creating a DTMF based routing system as presented in Chapter 4.2.2 if that solution is of interest.

6.1.3 Image Processor
Based on the model presented in Chapter 6.1.1, the image processor should be implemented as a separate unit, executing in parallel to the main thread of the service.

6.1.3.1 Fax Form Parsing
When optical recognition is used to solve the routing problem, every incoming document has to be run through the fax form parser. The parser could either consist of the evaluation system developed during the work on this thesis (see Chapter 4.1), or be based on an external package. The evaluation system has been implemented as a collection of C++ classes that are directly compatible with the fax image representation used in the fax image conversion part of the system (see next subchapter). The positions of the form fields have been hardcoded in the program, resulting in an extremely inflexible but quick solution. In a sharp version of the system, the form definition variables should preferably be handled by some kind of maintenance application instead (see Chapter 6.2).

6.1.3.2 Fax Image Conversion
The fax image conversion is preferably done using the tools presented in Chapter 5. These tools have been implemented as a collection of C++ classes that are based on a simple linear pixel buffer representation of the fax images.

6.1.3.3 DARC Transmission
When connecting to the Paging Server in the paging system, the Fax Server will identify itself as a Paging Terminal. The fax document data is then sent using Sectra Paging Protocol [8]. This is done using existing code from other projects.

6.1.3.4 Logging
One important feature is the logging functionality. It is obviously important to be able to go through the log files to identify erroneous transmissions, especially since the use of optical methods makes the system somewhat unreliable. The logging is preferably performed as a last step of the image-processing unit shown in Figure 38.

6.1.4 Additional Functionality
Another interesting functionality, that has not been considered yet, is the possibility to let the server call back and transmit a fax receipt, holding information about the parsed receiver address, file sizes and transmission timestamps, to the sender of the fax document. A fax receipt can of course only be sent if the fax modem is available, so this functionality will have to involve the fax receiver as well as the image-processing part of the system. The idea of a fax receipt solution has not been further evaluated, so there is room for further studies on this topic, as is mentioned in Chapter 7.2.
6.2 Fax Server Maintenance Application

To increase the usability of the system, a maintenance application for the fax server should be created. The server maintenance application should be used to start and stop the service, administer the fax modem, read logs and enter forwarding rules. It would also be preferable to have methods for positioning the form fields and get direct visual feedback on the result.

6.3 Fax Viewer System

As mentioned in Chapter 3.3.2, a paging message must be smaller than 32kb after decompression. Therefore, a multiple page document must be sent one page at a time. By using a document id number, it is possible to keep track on the pages that belongs to a specific document. Figure 39 shows an example of an incoming fax presented as a number of paging messages in the CitySurfer.

![Figure 39 - CitySurfer paging message window example](image)

When a page is selected and opened in the paging client, the desirable functionality would be to be able to pan across the image using the micro joystick on the CitySurfer. However, since the browser only supports vertical scrolling, another approach has to be found. The suggested solution is to divide every page into three slightly overlapping vertical sections that can be individually viewed, as can be seen in Figure 40. When a user chooses to view a page of a fax document in the CitySurfer, a thumbnail image is first displayed, in which one of the three sections of the page can be selected. When a section is selected, a 90° clockwise-rotated image of the selected section is displayed. As can be seen in Figure 40, this usually means that the CitySurfer has to be rotated 90° to allow the image to be viewed properly. This is however the only way to make a document scrollable in horizontal direction.
This solution can be realized using only one small HTML file, three tiled thumbnail images and three larger, rotated images. The HTML file is the actual message file, containing the thumbnail images, which are in turn links to the corresponding larger rotated images.

To prevent parts of the image from falling outside the screen, the height of each tile has to be less or equal to the width of the display in the CitySurfer. From this follows that since the display in the CitySurfer is 240 pixels wide, the tile can be only 240 pixels high. With three tiles we get a total image height of 720 pixels, which is roughly 1/3 of the vertical resolution of a standard resolution fax image.

Based on the results from a small number of image compression tests, this resolution also seems to be a good choice for preventing the uncompressed message size from exceeding the 32kB limit (see Chapter 3.3.2). However, since the resolution typically has a high impact on the image quality, a closer follow up on the resolution issue is probably required if the solution is to be used in a sharp version of the system in the future.

---

**Figure 40 - Enabling horizontal document scrolling in the CitySurfer**

A Portable DARC Fax Service
7 Conclusions and Further Studies

Working on this thesis has been an interesting and rewarding task. Despite the practical nature of constructing a software system, a number of more theoretical issues have been identified and handled. This chapter sums up the results from the previous chapters and represents the end of this thesis.

7.1 Conclusions

A number of possible solutions to the fax routing system have been presented, but since they all suffer from various flaws, unfortunately none of them can currently be considered perfect. Line-based routing is too inflexible, DTMF menu systems are not compatible with the way fax machines are used and Direct Inward Dialing is not compatible with the 9 digits serial number of the receivers. The optical recognition method, initially the solution requested by Sectra Wireless Technologies AB, would seem like a good choice, if it were not for the issue with erroneous recognitions.

It can be argued whether it was a wise decision or not to choose to develop the character recognition system presented in Chapter 4.1.3. The somewhat unsatisfactory results from the developed system could easily have been predicted, considering the limited amount of theoretical preparation and available time. However, as the resulting system is definitely good enough for evaluation of the fax service, the development of the character recognition system can hopefully be seen as a theoretically interesting sidetrack of the thesis.

Initially, it was highly doubted that the compression ratio of the SBM image format could stand up against the compression techniques used by market leading image formats. Therefore, it came as a big surprise to see how well the SBM format actually performed, in combination with the zlib-compression, compared to other image compression formats. Due to these results, there is no need for developing alternative image format support in the CitySurfer.

The fax viewer was simply a big disappointment. The limitations in the current CitySurfer firmware didn’t leave much room for development of a usable viewer system. On the other hand, this part of the system was among the last evaluated, so there is reason to believe that the evaluation might not have been as thorough as needed for making definite decisions based on the results.

7.2 Further Studies

The next step in the development of the fax routing system is definitely to try and improve the quality of the optical recognition routing method. One way to do this could be to continue the development of the own developed character recognition system, but the best choice would probably be to simply exchange it for a professional solution, if such a system would be considered worth the money.

In the area of the fax document conversion system, the next thing to do is probably to do a follow-up on the OCR conversion solution discussed in Chapter 5.3. This will involve the purchase and integration of a professional OCR.
conversion package and will hopefully result in a useful feature of the fax system.

In Chapter 6.1.4, fax-receipt functionality is mentioned. This solution involves sender-id extraction and would enable the fax server to return an optional receipt of the fax transmission to the sender of a fax document. This solution requires some rearranging of the suggested system layout and is an interesting topic for further studies.

The perhaps most important topic for further evaluation is the usability of the current fax viewer system presented in Chapter 6.3. The limitations of the viewer system are currently affecting the quality of the entire system and it seems like the only way to improve the viewer would be by allowing a reprogramming of the CitySurfer firmware. One idea, that has not been previously discussed, is to evaluate the possibility of creating a fax conversion system with support for both the old firmware and a new firmware version with extended viewer capabilities.
8 References


[2] DARC Introduction, Sectra, SWT-02.0233-1.0

[3] Sectra DRT-4000 Technical Description, Sectra, v5.0


[17] Browser Module Specifications, Sectra, CS-00.61-0.1


A CCITT Reference Images

CCITT1

CCITT2

CCITT3

CCITT4

CCITT5

CCITT6

CCITT7

CCITT8
På svenska

Detta dokument hålls tillgängligt på Internet – eller dess framtida ersättare – under en längre tid från publiceringsdatum under förutsättning att inga extra-ordinära omständigheter uppstår.

Tillgång till dokumentet innebär tillstånd för var och en att läsa, ladda ner, skriva ut enstaka kopior för enskilt bruk och att använda det oförändrat för ickekommersiell forskning och för undervisning. Överföring av upphovsrätten vid en senare tidpunkt kan inte upphäva detta tillstånd. All annan användning av dokumentet kräver upphovsmannens medgivande. För att garantera äktheten, säkerheten och tillgängligheten finns det lösningar av teknisk och administrativ art.

Upphovsmannens ideella rätt innefattar rätt att bli nämnd som upphovsman i den omfattning som god sed kräver vid användning av dokumentet på ovan beskrivna sätt samt skydd mot att dokumentet ändras eller presenteras i sådan form eller i sådant sammanhang som är kränkande för upphovsmannens litterära eller konstnärliga anseende eller egenart.

För ytterligare information om Linköping University Electronic Press se förlagets hemsida http://www.ep.liu.se/

In English

The publishers will keep this document online on the Internet - or its possible replacement - for a considerable time from the date of publication barring exceptional circumstances.

The online availability of the document implies a permanent permission for anyone to read, to download, to print out single copies for your own use and to use it unchanged for any non-commercial research and educational purpose. Subsequent transfers of copyright cannot revoke this permission. All other uses of the document are conditional on the consent of the copyright owner. The publisher has taken technical and administrative measures to assure authenticity, security and accessibility.

According to intellectual property law the author has the right to be mentioned when his/her work is accessed as described above and to be protected against infringement.

For additional information about the Linköping University Electronic Press and its procedures for publication and for assurance of document integrity, please refer to its WWW home page: http://www.ep.liu.se/

© Björn Husberg