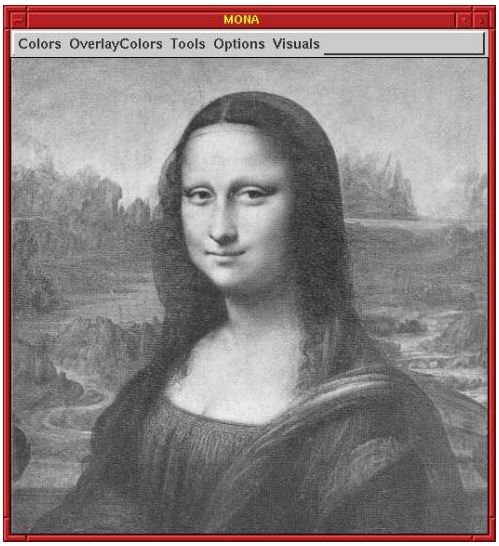
Image Processing Laboratory
Department of Informatics
University of Oslo

Report No. 91



XITE

X-based Image Processing Tools and Environment

System Administrator's Manual

For version 3.48

Svein Bøe

September 2013

Bildebehandlingslaboratoriet
Institutt for informatikk
Universitetet i Oslo

Image Processing Laboratory
Department of Informatics
University of Oslo

Tittel/Title:

XITE System Administrator's Manual

Forfatter(e)/Author(s):

Svein Bøe

Rapport nr./Report no.: **91** ISBN: **82-7476-061-1** Dato/Date: **September 2013**

Resymé/Abstract:

XITE consists of display programs with image widget and graphical user interface as well as more than 200 command line programs and 600 subroutines for image processing, all documented on-line.

The command line programs and subroutine library are written in C and run under UNIX and Windows.

The display programs run under UNIX. They work with images of arbitrary size and pixel type on 8-bit PseudoColor and 24-bit DirectColor and TrueColor X11 displays. Images can be zoomed and panned, and colortables can be selected from a menu. The main display program, xshow, gives access to most of the other command line programs via a menu interface which the user can customize and extend to include local programs. Input images for the menu entries can be selected with the mouse, and output images appear on the display.

This report describes how to install XITE and how to prepare a user account for the software.

Norske emneord/Indexing terms - Norwegian:

Bildebehandling
Bildebegrep
Bildeprogram
Vindussystemet X
Farger
UNIX
C
Windows

Engelske emneord/Indexing terms - English:

Image Processing
Image Concept
Display Program
X Window System
Colors
UNIX
C
Windows

Adresse:
Bildebehandlingslaboratoriet
Institutt for informatikk
Universitetet i Oslo
Boks 1080 Blindern
0316 Oslo

epost: blab@ifi.uio.no
tlf: 22 85 24 10

Address:
Image Processing Laboratory
Department of Informatics
University of Oslo
P. O. Box 1080 Blindern
N – 0316 Oslo
NORWAY

email: blab@ifi.uio.no
phone: +47 22 85 24 10

XITE

X-based Image processing Tools and Environment

System Administrator's Manual

For version 3.48

Image Processing Laboratory
Department of Informatics
University of Oslo

Svein Bøe
September 2013

Contents

1 Introduction

XITE (pronounced *excite*) is an acronym for "X-based Image processing Tools and Environment". The non-display programs run under UNIX (System V as well as BSD versions) and Windows. The display programs are based on the X Window System, Version 11, operating in a UNIX environment. XITE is — at least — being used on the following platforms

- Windows
- Linux

XITE consists of C source code, documentation, some images and colortables and installation scripts. If installing from a tar-ball, the XITE administrator may choose where to locate object libraries, executables, manual pages and other documentation. If instead installing the XITE rpm, all locations are predefined.

1.1 Required disk space

Estimated disk space requirements

Source code:	≈ 10 Mb
Manual pages:	≈ 7 Mb
Hypertext doc:	≈ 3 Mb
Other doc:	≈ 2–25 Mb
Scripts, images etc:	≈ 4 Mb
Libraries and executables:	≈ 8 Mb

1.2 Root/administrator privileges

If you install XITE from the tar-ball, you may need `root/administrator` privileges for the following

- Creation of XITE home directory.
- Creation of directories for XITE C header files, libraries, executables, man pages, hypertext reference manual and other documentation.

If you install the rpm, you will also need root privileges.

1.3 What you need besides XITE

If you want to be able to convert images between the XITE file format BIFF (Blab Image File Format) and other popular formats such as TIFF, `PNG`, `pnm` (`pbm`/`pgm`/`ppm`) and MATLAB's `MAT`-files, you will need the `libtiff` TIFF library[1], `libpng`[2] library, `Netpbm`[3] and MATLAB[4].

To build the TIFF-to-BIFF converters you need an ANSI C compiler, such as `gcc`.

[1] `http://www.libtiff.org`
[2] `http://www.libpng.org`
[3] `http://netpbm.sourceforge.net`
[4] `http://www.mathworks.com`

The XITE display programs need the X Window System, Version 11, a PseudoColor 8-plane (preferrably) or 24-plane DirectColor display. It should also work with a TrueColor 24-plane display, although the colortables can not be manipulated on such a display. Modern workstations often offer all these color modes in a single display.

1.4 About this report

This report describes how to

- install the XITE rpm if this is your preferred installation method
- build the object libraries and executables
- build hypertext and man page documentation
- hook a WWW browser to the *Help* button in the main display program
- prepare user accounts for access to XITE

Refer your XITE users to the User's Manual [2] for information on how to use XITE. The Programmer's Manual [1] describes how to write XITE dependent functions and programs, including the use of the X-based toolkit ximage. ximage is a part of XITE.

XITE has also been described in [3, 4, 5].

2 Installing XITE

2.1 On UNIX platforms

Go through the following three steps in order to install XITE.

1. For an rpm-based system, such as Red Hat, install `xite.rpm`. This completes the installation.
2. For non-rpm-based system,
   - Create an XITE home directory and set the environment variable

     `XITE_HOME`

     to reflect your choice. If you need more information about this, please refer to appendix A.1.
   - Move the script `install_xite` and the file `xite.tar.gz` (or `xite.tar.Z` or `xite.tar`) into the XITE home directory.
   - From the XITE home directory, issue the command (where `§` represents the shell prompt)

     ```
     § sh install_xite
     ```

   If you need more information about the install script and how to check for installation errors, please refer to appendix A.2.

2.1.1 Installing XITE for multiple architectures

If you decide to make a complete XITE installation on a different, non-rpm platform, then rerun the script `install_xite` as explained in section 2.

If you simply want to install binary executables, object libraries and possibly include files, manual pages and hypertext documentation for a platform in the same networked file system, while not duplicating the source files and scripts, there is no need to unpack XITE again. Instead, use the command

```
§ sh install_xite -d
```

You may want to check out the `-p` and `-loc` options. If you need more information on this, please refer to appendix A.3.

2.2 On Windows

There are two possible ways to install XITE on a Windows host. The easiest way is to start from the prebuilt XITE distribution for Windows.

If your Windows system has access to a UNIX file system where XITE is installed (e.g via Samba), you can instead start from this UNIX installation and leave the libraries and executables on the UNIX file system.

2.2.1 Using the prebuilt distribution for Windows

Standalone executables Go through the following steps

1. Move the file `xite.zip` to the C drive, e.g. `C:\xite`, or anywhere you choose.

2. Unzip the file `xite.zip`. This will create a subdirectory `xite`.

3. If the Microsoft Visual C++ dynamic link runtime library corresponding to your choice of compiler (e.g. `msvcr90.dll`) is not already installed on your system, get the file from the XITE download location or from Microsoft at no cost and copy it to the bin directory which was created when `xite.zip` was unzipped.

4. Set the `XITE_HOME` to refer to the top xite directory, e.g. `C:\xite`, and set `XITE_MAN` to refer to the XITE man-page directory, typically `%XITE_HOME%\man`.

The XITE to MATLAB bridge The XITE to MATLAB bridge is a collection of MATLAB mex files which call the XITE programs and return the results as arrays in MATLAB.

For MATLAB to be able to load and run the mex files, MATLAB must be able to find the libraries/dlls which the mex files depend on. The XITE library DLLs (`libxite.dll` etc.) must be on the Windows Path or in the same directory as the XITE mex DLLs.

If the XITE DLLs are located in `%XITE_HOME%\lib`, set the Windows PATH either in the control panel or in a command window with

```
set PATH=%PATH%;%XITE_HOME%\lib
```

If you have already started MATLAB, the `PATH` variable can be set from inside MATLAB by

```
§ setenv('PATH',[getenv('PATH') ';' getenv('XITE_HOME') '\lib'])
```

In MATLAB, add the locations of the mex executables to the MATLAB path. The mex executables are spread among a number of subdirectories below `%XITE_HOME%\matlab\xite` if you chose to use the prebuilt distribution.

You may also need to install C++ runtime dll as outlined in the previous paragraph concerning the standalone executables.

2.2.2 Starting from a UNIX distribution

Building standalone executables On a UNIX host, run the command

```
§ sh install_xite -d -p Win32
```

This will create makefiles for Windows and the necessary folders for libraries and executables. Check the files `Makefile.Win32` in the directories `etc` and `src` to make sure the paths are valid and in Windows notation.

If Microsoft Visual C++ 2008, compiler version 9.0, is installed on your system, open a Command Prompt window, go to the XITE source directory and issue the commands

```
§ set XITE_HOME=full_path_of_XITE_folder
§ "C:\Program Files\Microsoft Visual Studio 9.0\VC\vcvarsall.bat"
§ nmake /f Makefile.Win32
```

Building the XITE to MATLAB bridge The standalone programs for conversion between XITEs `.biff`-files and MATLABs `.mat`-files (`biff2matlab` and `matlab2biff`) need no special configuration. They should have been built by the `nmake` command above.

To build the mex files which can call XITE functions from MATLAB, the mex compiler must be configured. Run the MATLAB mex-script with the `-setup` option to produce a file configured for the chosen compiler

```
§ mex -setup
```

Make a note of where the file mexopts.bat is placed and copy it to `%XITE_HOME%\src\matlab\mexopts.bat`.

Finally, compile all the mex files with the commands

```
§ cd %XITE_HOME%\src\matlab
§ nmake /f %XITE_HOME%\etc\Makefile.Win32 mexobj mexlib xite_mex_dso xitemex
mexhelp
§ cd xite
§ nmake /f %XITE_HOME%\etc\Makefile.Win32 mathelp
§ for /d %d in (*) do (cd %d & nmake /f %XITE_HOME%\etc\Makefile.Win32 &
cd ..
```

3 Customizing the installation

After having installed XITE successfully, there are a few remaining steps which you need to go through manually, in order to make XITE operate at its best.

3.1 Setup of user accounts

3.1.1 On UNIX platforms

The installation procedure also produced setup files for XITE users. If you installed the rpm, every user logging in to the computer will automatically have the correct setup for XITE.

Without the XITE rpm, the easiest way for a UNIX user to get access to XITE is to invoke XITE setup files from the initialization files of the login shell. If the login shell is csh-compatible, include the following line in the user's `.cshrc` file

```
source $XITE_HOME/etc/xite_cshrc
```

(where `$XITE_HOME` is replaced by the actual XITE home directory). If the login shell is sh-compatible (e.g. bash), include the following line in the user's `.profile` file

```
. $XITE_HOME/etc/xite_profile
```

As the XITE administrator you should check that the files `xite_cshrc` and `xite_profile` reflect the choices made during installation.

3.1.2 On Windows

The `PATH` environment variable must be modified so that XITE programs can be started from the Windows command window. Add the name of the folder in which the XITE executable programs reside, to this environment variable. The folder name is probably `bin` or `bin\Win32` below the XITE home directory.

The environment variable `XITE_MAN` should be set to refer to the folder in which the program manual files reside. On a default installation this will be `$XITE_HOME/man`[5]. The environment variable is required for the `-man` and `-whatis` program options to work.

3.2 Building a manual page database

A manual page database is required for the XITE program option `-whatis` to work.

3.2.1 On UNIX platforms

On a UNIX system, the whatis program and man with certain options also need the manual page database.

The XITE manual pages have been installed in a directory determined by the rpm or which you chose during the unpacking procedure. On a UNIX system, you should probably go to this

[5] or `%XITE_HOME%\man` in Windows notation

directory and build a whatis or windex database for the man pages. This will enable keyword search with man to work. On my local systems I use one of the two commands

```
§ makewhatis $XITE_HOME/man
§ catman -M $XITE_HOME/man
```

(Replace $XITE_HOME by /usr/share/xite for an rpm-based installation). This may be different on your system.

3.2.2 On Windows

A windex database and preformatted man pages are supplied with the prebuilt Windows distribution.

3.3 X app-defaults resource files

This section applies only to platforms with the X Window System installed.

The XITE display programs will not work properly without the app-defaults files. If you installed with the rpm, they will be located in the standard directory /usr/share/X11/app-defaults/xite.

If you installed without the rpm, they will be found in the directory

```
$XITE_HOME/etc/app-defaults
```

Each display program sets the environment variable XAPPLRESDIR[6] so that the app-defaults files can be found by the X Window System.

If you already have a previous version of XITE, it may be that the resource files for the previous version were installed in the standard location[6]. You should remove those files.

A warning will be printed if the display programs can't find these files.

See documentation on the X Window System for more information about the above environment variables.

3.4 WWW browser as xshow help program on UNIX platforms

xshow tries to load an external helper application when the *Help* button in the Control window is pushed. The default behaviour is to start the WWW browser firefox, pointing to the XITE Reference Manual (which was built during the installation procedure). If firefox is not available, xshow will usually try to launch another program. The exact behaviour is determined by the X application resources xiteHelper and xiteHelperOption or the environment variables XITE_HELPER and XITE_HELPER_OPTION.

If you want to customize the behaviour, refer to XITE User's Manual [2] or the manual page for xshow.

[6] possibly also XUSERFILESEARCHPATH

3.5 Customization of the xshow menu tree

The main display program xshow can launch most of the other programs supplied with XITE. This is done via a menu hierarchy which is defined in an ascii menu file. One of the entries in the main menu is *Site-specific programs...*. If you want all your users to get access to locally developed programs which operate on images in the XITE file format BIFF, these programs may appear in a submenu tree below this entry.

For information on how to add local programs to the menu hierarchy, refer to the manual page for xshow.

3.6 Compilation of XITE dependent software

A script, cxite, is supplied for compilation and linking of new programs which use XITE routines. It was installed in the XITE executable directory you chose during installation. On UNIX platforms, the script was automatically edited by the install script. It should be able to supply the necessary options to find XITE header files, the XITE library as well as other required header files and libraries.

The command

```
§ cxite -help
```

will give more information about this script.

cxite is not available under Windows.

4 Miscellaneous

4.1 Bug reports

If (or when) you encounter bugs, whether during installation or execution, please send bug reports to the XITE developers. You'll find a bug report template in the file

```
$XITE_HOME/etc/.install/bugReport
```

or in `/usr/share/doc/xite-3.48/bugReport` for an rpm-based installation.

Mail the bug report to *xite-bugs@ifi.uio.no*.

4.2 Registration

If you install and use XITE, please let us know by filling in the registration form in the file

```
$XITE_HOME/etc/.install/REGISTER
```

(or `usr/share/doc/xite-3.48/REGISTER`).

Mail it to *xite-register@ifi.uio.no*. This will help us get a general idea of what people find useful, what platforms XITE is most widely used on and whether the users are in universities, research institutions etc.

The use of XITE is free, but we would be happy to know that you find it useful.

4.3 Copyright

XITE comes with a mild copyright notice. You'll find a copy of this in the file

```
$XITE_HOME/etc/.install/COPYRIGHT
```

(or `/usr/share/doc/xite-3.48/COPYRIGHT`).

4.4 Mailing list

An international mailing list exists for discussions about XITE. If you would like to subscribe to this list, send a mail to *xite-request@ifi.uio.no*. The list name is

xite@ifi.uio.no

4.5 WWW home page for XITE

With your favourite World Wide Web browser, such as firefox, you may find more information about XITE in its home page with URL

```
http://www.mn.uio.no/ifi/english/research/groups/dsb/resources/software/xite
```

4.6 How to contribute to XITE

If any of your users writes software which she would like to contribute to XITE, refer her to the Programmer's Manual [1] supplied with the XITE distribution. Please email *blab@ifi.uio.no* if further instructions are needed.

If anyone at your site makes changes or fixes bugs, please let us know. Refer to section 4.1 for details on sending a bugreport.

A Details on the installation procedure

If you ran into problems during the UNIX platform installation procedure which was briefly outlined in section 2, you may find help here.

A.1 XITE home directory

For a tarball-based installation, all the C source code and installation scripts will by default be placed below the XITE home directory. For an rpm-based installation, the source code is packaged inside the source rpm, as a tarball.

The object library, executables, manual pages and other documentation do not need to reside below the XITE home directory. This can be chosen during the installation procedure.

Set the environment variable `XITE_HOME` equal to the chosen XITE home directory. On a UNIX system, if you are using the csh or tcsh shell, this can be done with the command

```
§ setenv XITE_HOME /usr/local/xite
```

(if you chose `/usr/local/xite` as the XITE home directory). If you are using an sh-compatible shell, e.g. bash, use the command

```
§ XITE_HOME=/usr/local/xite; export XITE_HOME
```

Throughout this document the string `$XITE_HOME` represents the chosen XITE home directory.

Before unpacking, building and installing XITE, the home directory must be created. If you have chosen `/usr/local/xite` as the XITE home directory, issue the commands

```
§ cd /usr/local
§ mkdir xite
```

You will probably need `root` privileges to be allowed to create the directory. You could, of course, alternatively have chosen to install XITE below your own home directory, in which case you would not need any special privileges at this stage.

A.2 Unpacking, building and installing

A.2.1 What you need before unpacking

Before you start to build and install XITE, make sure that you have everything you need. You should have the following files

- `install_xite`
- One of `xite.tar.gz`, `xite.tar.Z`, `xite.tar`

For an rpm-based system, you only need `xite.rpm`, and possibly xite-src.xpm.

If you lack any of these files, you will find them with a WWW browser pointed to XITE's home page, at

```
http://www.mn.uio.no/ifi/english/research/groups/dsb/resources/software/xite/
```

Move the above files into `$XITE_HOME` and change to that directory,

```
§ cd $XITE_HOME
```

Do **not** unpack the `tar` file manually.

A.2.2 Unpacking the tar-ball

The script `install_xite` will unpack, build and install XITE. To get a list of the various options supported by this script, give the command

```
§ sh install_xite -h
```

Note in particular the options `-d`, `-f` and `-o`. You may be using these options if you choose to install XITE for multiple computer architectures.

The easiest way to unpack, build and install XITE is to simply issue the command

```
§ sh install_xite
```

and give the information requested by the script as it runs. Be prepared to supply the following

- Directory for storage of XITE C header files
- Directory for storage of XITE object libraries
- Directory for storage of XITE executables
- Directory for storage of XITE manual pages
- Directory for storage of XITE hypertext documentation

The default values suggested by the installation script for the above are all below the XITE home directory.

For the sake of the display programs in XITE, the following information is also needed

- Location of header files for the X Window System
- Location of object libraries for the X Window System

Depending on whether you have MATLAB, the `libtiff` TIFF object library, the `libpng` object library or the `Netpbm` conversion utilities installed on your system, you may also want to supply the location of

- `libtiff` TIFF header files
- `libtiff` TIFF object library
- `png` header files
- `png` object libraries

- pnm header files
- pnm object libraries
- MATLAB header files
- MATLAB object libraries
- MATLAB mex files (for storage)

At some point during the installation procedure you will be asked whether you want to continue building the library, executables and documentation. This means that unpacking of the XITE distribution has been completed. What remains is the actual compilation, linking and documentation production. This is handled by a separate script, makexite, which is located in `$XITE_HOME/etc`. If the installation so far has gone by without problems, just type `y` in response to this question, and makexite will be started automatically.

A.2.3 Checking for errors

All of the compilation errors and warnings found while the makexite script is run, will be stored in a file with filename starting with the string `xite_log`, and with filename extension given by the computer architecture you are installing XITE on. You should check this file when the installation is finished, looking especially for error messages. A quick way of doing this is with a command such as

```
§ grep -i error xite_log.IRIX | grep -v ignored
```

(if you are installing XITE on a computer running the IRIX operating system). This will tell you whether any errors were found. Search through the file to see what the errors were, before sending a bug report to the XITE developers (see section 4.1 on how to send a bug report).

A.2.4 Rebuilding of parts which failed

If the building of XITE libraries or executables failed, you may try to rebuild the system using a different compiler, different compiler options or change other information which was acquired during the unpacking phase. Rebuilding of XITE can be done without unpacking the distribution once again. Go through the following steps

- From the log file, try to locate what part of the build process failed. In which source directory did it occur? Did the compilation of library functions fail, or compilation of executables? Or perhaps there was an error during the linking phase?
- Do you think that it would help making changes to the `Makefile` for the particular architecture you are installing on? You will find the corresponding makefile in `$XITE_HOME/etc`, with an appropriate filename extension. Perhaps you need to change the `CFLAGS` variable (options given to the compiler), or perhaps you want to use a different compiler (the `CC` variable).
- Rebuild the parts of the system which failed, by using the script

  ```
  $XITE_HOME/etc/makexite
  ```

  Give the option `-h` to get help on how to use this script.

A.3 Building XITE on a different platform

If you ran into problems when following the instructions in section 2.1.1, read this section.

If the new platform is unknown to XITE (install_xite does not accept the name of your desired platform), go through the following steps

1. Edit the script `$XITE_HOME/etc/.install/testPlatform` to allow this extra platform.
2. Among the platforms mentioned in the testPlatform script, decide which one is closer to the new platform. Check to see whether there is a corresponding Makefile or Makefile template (extension `.tmpl`) in `$XITE_HOME/etc`. If you can't find one, generate it with the command

   ```
   § $XITE_HOME/etc/.install/makeMakefile -p <platform name>
   ```

   Make a copy with filename extension equal to the new platform name.
3. Incorporate any necessary changes in the Makefile, particularly to `CFLAGS`, possibly also to `LFLAGS` and `CC`.
4. Check the files `includes.h` and `xite_*.h` in the XITE header file directory to see that the appropriate header files will be included with the particular platform. You may need to add extra #ifdef directives to treat your platform appropriately.
5. Use the command

   ```
   § sh install_xite -d
   ```

   or

   ```
   § sh install_xite -d -p
   ```
6. Send an email to *blab@ifi.uio.no* describing this new platform and any changes or additions you made to XITE in order to make it run.

References

[1] Svein Bøe. XITE – X-based Image Processing Tools and Environment – Programmer's Manual for version 3.4. Report 92, Image Processing Laboratory, Department of Informatics, University of Oslo, P. O. Box 1080 Blindern, 0316 Oslo, Norway, June 1998, `http://www.mn.uio.no/ifi/english/research/groups/dsb/resources/software/xite/ProgrammersManual` 2, 10

[2] Svein Bøe, Tor Lønnestad, and Otto Milvang. XITE – X-based Image Processing Tools and Environment – User's Manual for version 3.47. Report 56, Image Processing Laboratory, Department of Informatics, University of Oslo, P. O. Box 1080 Blindern, 0316 Oslo, Norway, June 2004, `http://www.mn.uio.no/ifi/english/research/groups/dsb/resources/software/xite/UsersManual/`. 2, 7

[3] H.I. Christensen and J.L.Crowley, editors. *XITE – X-based Image Processing Tools and Environment*, pages 63–88. World Scientific, Singapore, 1994. ISBN 981-02-1510-X. 2

[4] Otto Milvang and Tor Lønnestad. An image widget for image processing. *The X-Journal*, 1(2), November/December 1991. 2

[5] Otto Milvang and Tor Lønnestad. An object oriented image display system. In *Proceedings, 11th International Conference on Pattern Recognition*, pages D218–221, The Hague, The Netherlands, August/September 1992. 2

REPORTS FROM THE IMAGE PROCESSING LABORATORY

78 Maurycy Szmurlo :
"A comparative study of statistically classifiable features
used within the area of Optical Character Recognition"
Cand.Scient. (Master) thesis May 1995.

79 Ingvil Hovig :
"Verktøy og metoder for komprimering av MR bilder"
Dr.Scient. (Ph.D) thesis August 1995.

80 Øyvind Akerhaugen :
"Automatisk plassering av navn på kart ved hjelp av simulated annealing"
Cand.Scient. (Master) thesis February 1996.

81 Marius Midtvik :
"Reversibel komprimering av MR bilder basert på statisk kildemodellering"
Cand.Scient. (Master) thesis May 1996.

82 Tor Øyvind Didriksen :
"Linjefinning og klassifikasjon med Random Hough-transform
— en eksperimentell studie"
Cand.Scient. (Master) thesis June 1996.

83 Edward Allen Smith :
"Image Processing Techniques on DNA Fingerprint Images
and its Application to Genetic Similarity Analysis"
Cand.Scient. (Master) thesis August 1996.

84 Parviz Heydari :
"Line Following in Digitized Map"
Cand.Scient. (Master) thesis November 1996.

85 Ramin Gordjianfar :
"Vectorization of the Cartographic Data"
Cand.Scient. (Master) thesis November 1996.

86 Luren Yang and Torfinn Taxt :
"Robust Methods for Sonar Bottom Detection"
February 1997.

87 Christian Wladimir Hansson :
"Strukturgrammatikk — en høyeredimensjonal grammatikk
for syntaktisk mønstergjenkjenning av 3D objekter i bilder"
Cand.Scient. (Master) thesis December 1996.

88 Sverre H. Huseby :
"Video on the World Wide Web —
Accessing Video from WWW Browsers"
Cand.Scient. (Master) thesis February 1997.

89 Håvard Lauritzen :
"Raster til vektor konvertering ved simulert størkning"
Cand.Scient. (Master) thesis May 1997.

90 Irene Rødsten :
"Texture Segmnetation using Moment based Features
obtained by Locally Adaptive Thresholding"
Cand.Scient. (Master) thesis May 1997.