TORNAI RÓBERT: MMA SOFTWARE DEVELOPING AND BITMAP TRANSFORMATIONS

MMA Software Developing and Bitmap Transformations

PhD dissertation

Tornai Róbert

Debreceni Egyetem
Debrecen, 2005.

Ezen értekezést a Debreceni Egyetem TTK Matematikai és Számítástudományi Doktori Iskola Informatika programja keretében készítettem a Debreceni Egyetem TTK doktori (PhD) fokozatának elnyerése céljából.

Debrecen, 2005. szeptember 1.

A jelölt aláírása

Tanúsítom, hogy Tornai Róbert doktorjelölt 1999-2005 között a fent megnevezett Doktori Iskola Informatika programjának keretében irányításommal végezte munkáját. Az értekezésben foglalt eredményekhez a jelölt önálló alkotó tevékenységével meghatározóan hozzájárult.

Az értekezés elfogadását javaslom.

Debrecen, 2005. szeptember 1.

A témavezető aláírása

Contents

List of Figures

Introduction

In this PhD dissertation I attempt to gather the results of my latest developments together into a rounded work. Lately I have developed a medical software in a team work as program designer and programmer. My colleagues were Péter Görömbey as project manager, and Gábor Rácz and Antal Smudla as programmers. Some of the work described here in conjunction with the MMA technology and the related softwares is common (Chapter 1 and Chapter 2), while the last chapter (Chapter 3) reflects completely own work.

The first chapter covers the MMA technology that was worked out together, the whole team took part in it. Péter Görömbey was the project manager, furthermore he was responsible for the mathematical part of the project (cluster classification and curve fitting). Gábor Rácz coded the common class hierarchy. Antal Smudla developed the PC interface and made the graphical objects. Besides taking part in the designing work, I did almost entirely the porting of the softwares (including the development of new control objects).

The second chapter deals with the cross platform developing. It lists the porting issues and their solutions. I played an important role in choosing the developing language and tools for the platforms and solving the problems arisen during the programming work.

The third chapter of this work contains my results in bitmap transformations to various surfaces and shapes. I have started this research with Dr. Schwarcz Tibor and later continued alone. The common articles reflect 50%-50% work. This part introduces solutions based on already known techniques that were not used for bitmap transformations so far. Some of the transformations could be applied in an easier way, but in computer graphics not always the easier way is the fastest. Many times a different approach or method, which may seem more complicated at a glance, is better, because the partly results can be reused more effectively and yields less computation all together.

Chapter 1

The MMA Technology

MMA stands for the Multiplex Microbead Assay term. The MMA technology is an analytical method. It is used in for medical and experimental purposes. The aim of our team was to build up a complex method that is supported by automated computer programs. This chapter gives the problem's description and skeleton of the solution how to handle Multiplex Microbead Assay measurements. It will give an overview of the most important equipment producer corporations, the main steps of a measurement, the structure of a List Mode (LM) file, the architecture of the developed programs and the current state of the project.

1.1 Introduction

The basic problem is to evaluate the results of flow-cytometry measurements. Our team has developed two softwares in corporation with Soft Flow Kft. (see [10]). The softwares support the field of microbead technology inside the flow-cytometry and can be used to evaluate measurements related mainly to immunology. Flow-cytometry is a very young technology, the first cytometer launched the market in 1968. In fact it is a diagnostic and analytical method.

1.2 Principles of the MMA Technology

The advantage of the MMA technology is that several reactives can be used in a measurement, provided that several stains are used with homogeneously mixed beads that are covered with different reactives (see [1], [2], [3]). This way the necessary amount of sample (blood, plasma or tincture) will be significantly smaller. The beads are identified by the emission of the contained fluorescent stain. If only one stain is used with more numerous concentrations, then maximum 10-16 beads can be distinguished (with little overlay) from the data of one channel. Two channels' data is needed when two stains are used, so the maximum number of distinguishable beads rise to 256. Theoretically more stains can be used as well, but there is a biological limit to create such a complicated system. (Even the usage of three stains is not common.) The number of distinguishable beads can be extended by using different sized beads, namely the size of different beads correlate with side scatter of the laser beam. The software firstly recognize the same sized beads, which is done with the help of the side scatter channel (SSc). The second step is to distinguish the events by analytes using one or two dimensional area division method. The segregated patches are called clusters.

1.2.1 Flow-cytometers

To perform a flow-cytometry measurement, a flow-cytometer machine is needed. This machine can be a general-purpose (used for research), an independent or a special hardware, which is the main part of a haematologic machine (see [7], [8]).

The most well known flow-cytometer manufacturers are: Beckton Dickinson, Luminex (Figure 1.2.1 shows a popular model, see [4]), Coulter and Bio-Rad.

1.2.2 Aim of the Developed Softwares

The aim of the developed softwares is to help doctors and researchers in using multiple beads during measurements (see [5], [6]). Although the beads were ready for a while for use, there were no fashioned way for using them. Doctors and researchers have made up better or worse own systems, and applied them by many paperwork and computations by hand. We wanted to give a well-fashioned schema and support it by automating the process by the developed programs as far as it is possible.

There are a lot of Macintosh systems installed already in the laboratories, while new medical systems usually come with IBM compatible computers. This is why we kept the primary development on the PC version, and the Macintosh version always followed it. There were no doubt that we have to support the Mac OS series 9, since Macintosh is widespread in medical solutions for a long

Figure 1.2.1: Picture of a Luminex 100 system

time. Meanwhile the Mac OS X has arrived, so we adjusted the Mac version to the new environment, too.

ASF Creator supports the creation of an Assay Specification File that comes with a kit. In turn, the SAT Flow software supports a whole measurement process, where one or more kit can be used for the samples.

The softwares do not support haematologic machines, since these machines were developed for special haematology tasks and their results can be interpreted directly without using external software or hardware. The software supports flow-cytometers which are mainly used by researchers for various type of measurements. Hospitals do use these kind of machines as well, however the current software version of SAT Flow does not have license nor have certificate to use it in such environments. The reason is that hospital usage was not a purpose. The most important consideration was to fulfill the demands of the research usage and as it has been proven, the final product perfectly fit these demands.

1.3 Steps of a Measurement

The guiding principle is as follows:

- First of all, the assay to be examined, which can be a group of cells or a microbead mixture, mixed with the appropriate reagents.

- After this step, the obtained mixture is mixed with an organic fluorescent stain that is connecting to the reagent.
- The mixture inserted into the machine.
- The diluted mixture goes into the cuvet (exposition box) through a capillary.
- The high speed mixture is lit by laser in this box (see Figure 1.3.1). The laser light is reflecting in various directions or absorbing in it. In the second case the laser excites the stain molecules, so they begin to emit light.

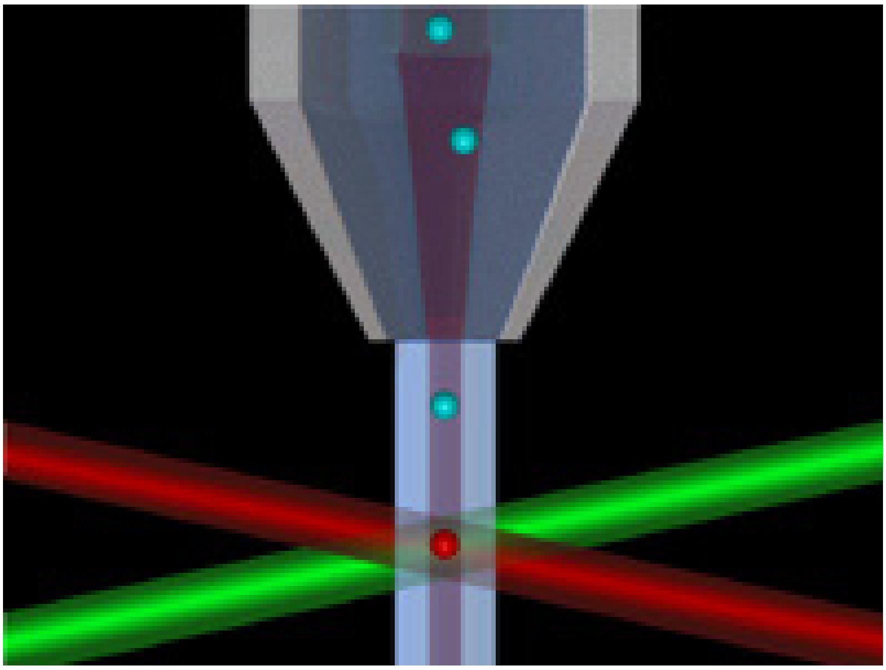


Figure 1.3.1: Beads are lit by laser beams

- The 'event' phrase will be used for the assay that has just passed by the laser beam.

- The intensity of the reflected light is perceived by some well placed sensors.
- The signs are forwarded to the cytometer's processing unit. The signs are digitized here. Older machines use 8, 10, 15, 16 bits, the newest models are capable to use 32 bits. Usually a logarithmic conversion is possible to achieve the best results.
- The digitized channel values get to the collector machine through the data cable of the cytometer. This is usually a PC, where a special collector software interprets the data. A file is created for each measurement. This file is called List Mode file (abbreviated form: LM file).

1.4 List Mode Files

Flow-cytometer instruments produce List Mode files that usually follow the Flow Cytometry Standard (abbreviated form: FCS). The FCS structure contains:

- Header (sequence of ASCII characters): contains ID-value pairs separated by one or two special characters (called separators). Separators are producer defined. Some of them follow the standards, others are uniquely defined.
- Constant values: e.g. powering factor, digital conversion resolution (bit depth), name of the channels, date and time of the measurement, etc.
- Number of events: number of the perceived beads.

The following parts are bytes making up data sequences that can be interpreted by the previous informations. These data sequences store the converted, digitized channel values for the events in time sequence.

Nowadays CSV-based (Comma Separated Values) List Mode files are getting widespread among Luminex instruments. They include the intensity values assigned to the events one after other. The new version of our softwares are capable to recognize and process these files, too.

1.5 The ASF Creator Software

The software does not even start without authorization. It starts after a login procedure (see Figure 1.5.1).

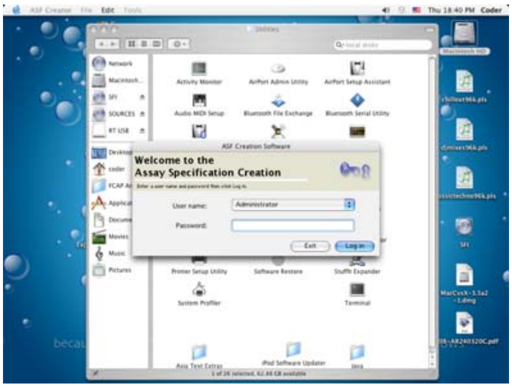


Figure 1.5.1: Login screen of the ASF Creator

After the program started, everything is logged (see Figure 1.5.2). The log is archived to a file, but the log screen can be hided.

A new assay kit generation can be induced from the File menu. The new assay kit can defined by giving all data or by modifying an existing kit's data (see Figure 1.5.3). A new assay kit can support one of the qualitative or quantitative models. It can be set on one of the following wizard pages.

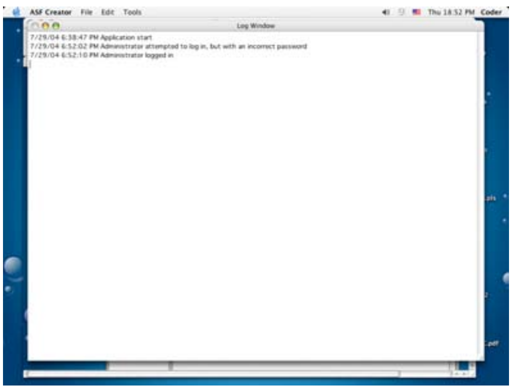


Figure 1.5.2: Log of events

The producer of the kit assigns a name and an ID to the assay that will be the identifier of it. LOT is a unique number that identifies a batch. A batch is an amount of preparation produced together.

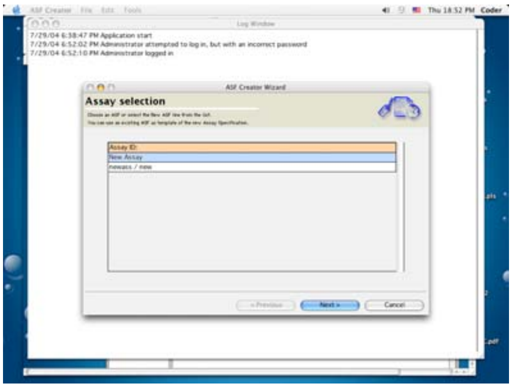


Figure 1.5.3: First step of creating a new assay

The Name - ID - LOT triple uniquely identifies a kit. The number of clusters determines how many clusters will be in the AS file. Expiration date is the creation date and the default expiration interval (3 months) that can be changed in the Preferences in the File menu. Creation date shows when the kit is produced (see Figure 1.5.4).

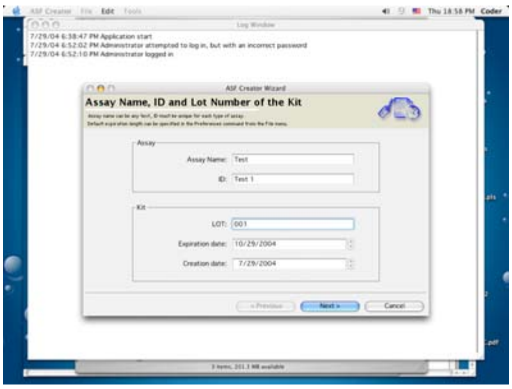


Figure 1.5.4: Naming and identifying a new assay

Assay type defines the evaluating mode. The Number of Clusters determines how many clusters will the software look for in the LM file in the followings. The clustering parameters will define the clusters. The Number of Scatter (SSC) Peak can be only 1 in this version of the program (see Figure 1.5.5).

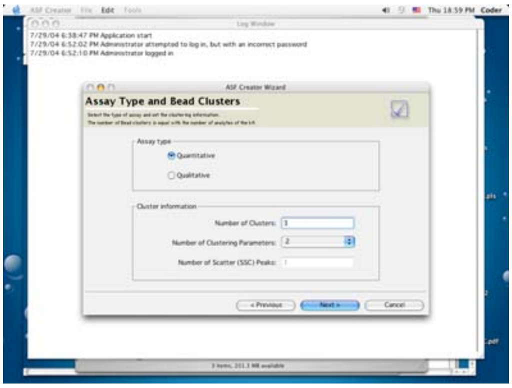


Figure 1.5.5: Assay type selection

Only those instruments' LM files will be recognized by the SAT Flow software that are listed here with the proper parameter order (see Figure 1.5.6). The combo box for the Instrument name has the elements of the Instrument List of the known instruments (see Figure 1.5.36).

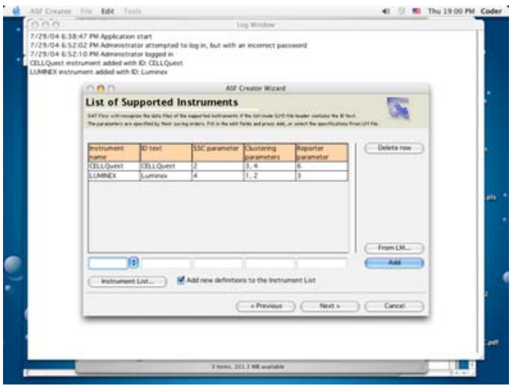


Figure 1.5.6: List of the supported instruments by the assay

Instruments can be defined by LM files made with the instrument to be added (see Figure 1.5.7).

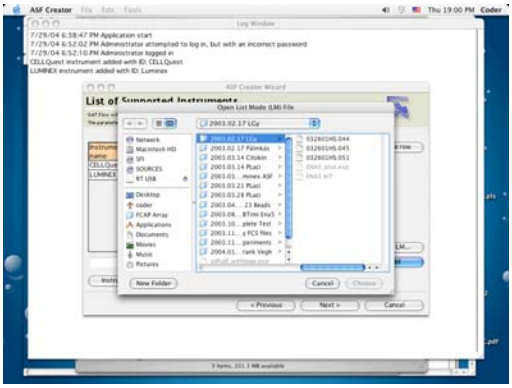


Figure 1.5.7: List Mode file handling for easy identification

The user has to give a name to the instrument and select a text from the file that identifies it uniquely. Furthermore, the SSC, clustering and reporter parameters have to be defined by combo boxes that are filled up with predefined values (see Figure 1.5.8).

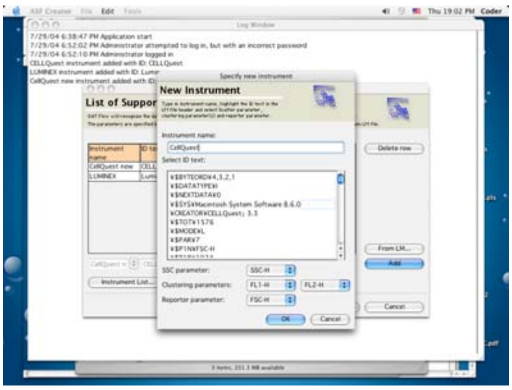


Figure 1.5.8: Specifying a new instrument

The ID text has to be unique. If it is not so, the program warns the user (see Figure 1.5.9). The newly added instrument appears in the list immediately

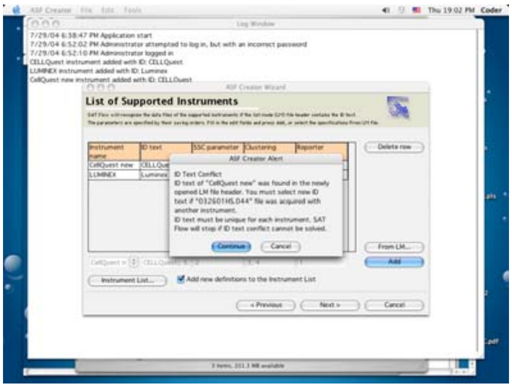


Figure 1.5.9: Error message of adding a new instrument

(see Figure 1.5.10). It is a good idea to add the instrument to the list which has created the LM (List Mode) data file.

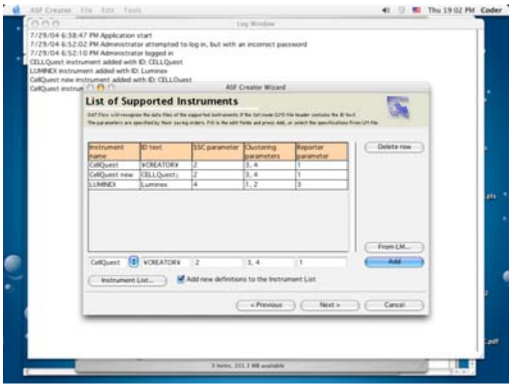


Figure 1.5.10: List after adding a new instrument

The next step is to specify the LM file holding the data of the assay kit. The LM file has to be created with one of the known instruments. The parameters take the default value from the instrument list (see Figure 1.5.11). The user can override these default values by modifying the combo boxes' selections. By pressing the next button, the software checks the LM file. When the automatic

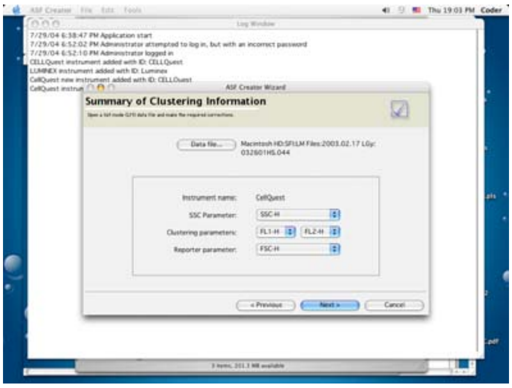


Figure 1.5.11: Data file specifying and supervising of the settings

processing fails that means there are not enough clusters, a message box warns the user to review the clustering parameters (see Figure 1.5.12).

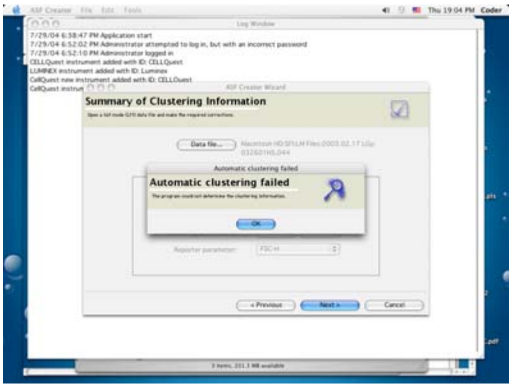


Figure 1.5.12: Error message warns the user of a failure in the clustering

If the clustering is done without problem, the user has to pair the analytes and the beads. The table can be edited with combo boxes. The combo boxes have the elements of the known beads and analytes from the bead and analyte lists (see Figure 1.5.13).

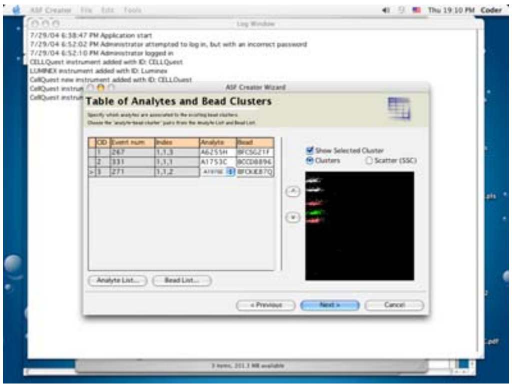


Figure 1.5.13: The cells of the table are edited by a combo box

The finished table is shown on Figure 1.5.14. If Show Selected Clusters is enabled, the individual clusters will have respective colors. The figure on the right can visualize either the clusters (see Figure 1.5.13) either the side scatter data (see Figure 1.5.14).

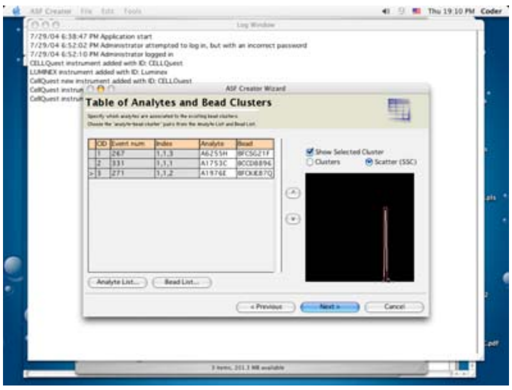


Figure 1.5.14: The table filled up with analyte and bead pairs

From this step we get different wizard pages based on the selected model.

1.5.1 Quantitative model

Until this step, the wizard pages of the Quantitative and Qualitative models are the same.

In Quantitative model each analyte has to be assigned to a fitting equation with the help of a list box (see Figure 1.5.15). For example cubic, four parameter logistic and exponential sigmoid functions can be selected. The elements of the list box are fixed and can not be modified by the user.

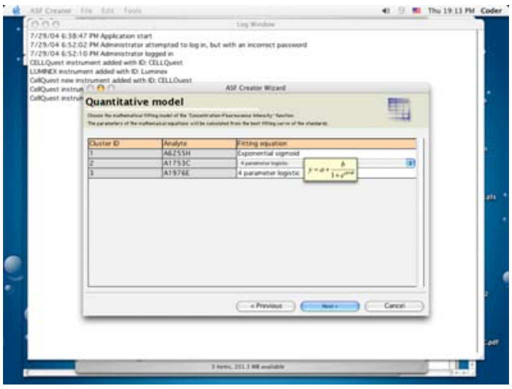


Figure 1.5.15: Fitting equations for analytes (tooltip shows the exact formula)

Standards will yield the base points of a curve that converts the intensity values into concentrations. The fitting functions of the quantitative model determine their parameters based on the measured intensity values and the given concentrations of the standards. They fit a curve on the yielded points. Putting in the test samples intensity values in this function we get the corresponding concentrations. A kit can contain arbitrary number of standards, but since the standard samples are very expansive, a reasonable number of them is expected (see Figure 1.5.16).

Each standard will have a column. The column of the concentrations is editable with a list box. The valid concentrations are given in a list that can be edited (see Section 1.5.3) from here by pressing the List of units button (see Figure 1.5.37). The columns of the standards can be edited by an edit box that allows entering numbers.

The number of replicate of the standards can be set individually or together by setting the entered number for the default number.

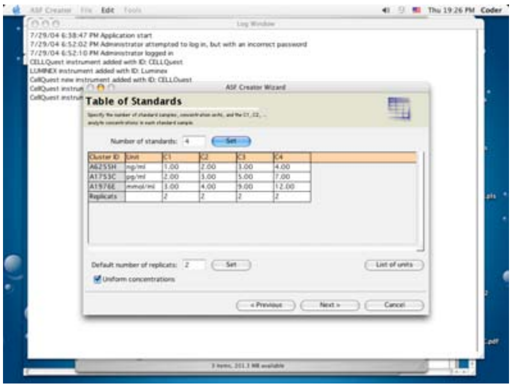


Figure 1.5.16: Filled up table of standards

The quantitative control samples can be used for verification purposes. They have three concentrations: low, medium and high. The Low C, Medium C and High C columns store the nominal concentration values for the analytes. The Low dx, Medium dx and High dx columns store the allowed deviance from the nominal concentration values in percentages (see Figure 1.5.17).

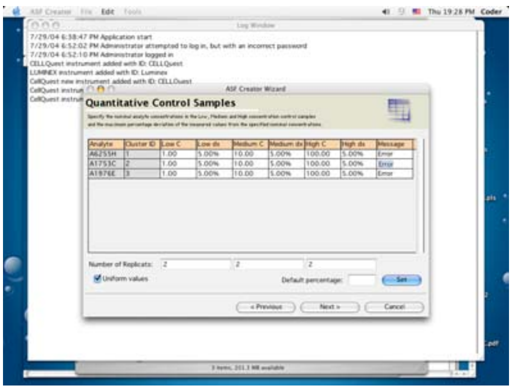


Figure 1.5.17: Messages for the quantitative control samples

The message column has the warning texts that appears in the reports of the SAT Flow program when the concentrations are outside of the proper interval.

When the concentration of an analyte satisfies certain conditions, then the user is prompted a message. This can be the text of the diagnosis for example. These messages can be defined on the next wizard page (see Figure 1.5.18).

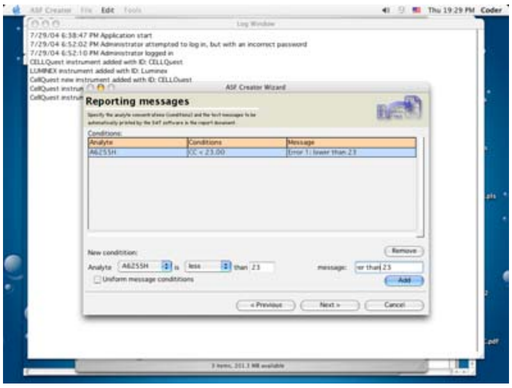


Figure 1.5.18: Error messages for the analytes

On the last option page of the wizard it can be set how many instrument standard beads will be in the kit. Their Relative Fluorescent Intensity (RFI) can be given separately. These standards can have arbitrary number of replicates, only the cost limits this number (see Figure 1.5.19).

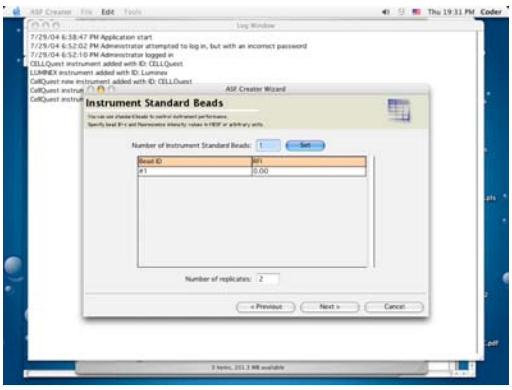


Figure 1.5.19: Setting the instrument standard beads

The last information page summarizes the choices made so far (see Figure 1.5.20). The user can print and save the kit to file here. Furthermore, there is a possibility to set up the printer or get a preview of the printed kit documentation.

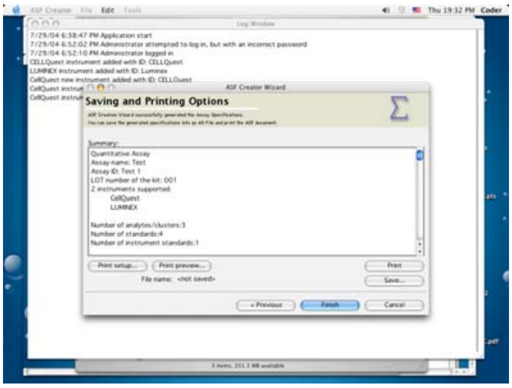


Figure 1.5.20: Last page of the wizard with saving and printing of the assay

The print preview shows how the official printout will look like (see Figure 1.5.21). It is possible to zoom the preview to predefined sizes. The fonts of the printout can be changed from the File menu.

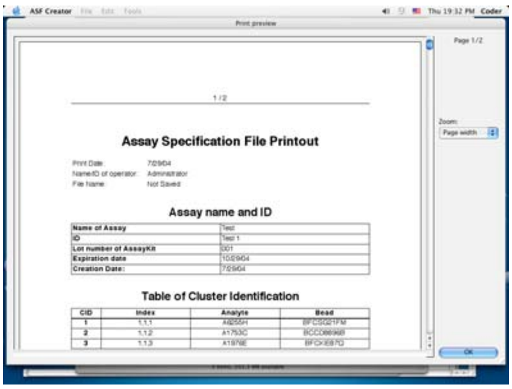


Figure 1.5.21: Print preview of the created assay

For the page setup the standard dialog boxes can be used. The page setup includes the paper size, orientation and scaling factor setting, too (see Figure 1.5.22).

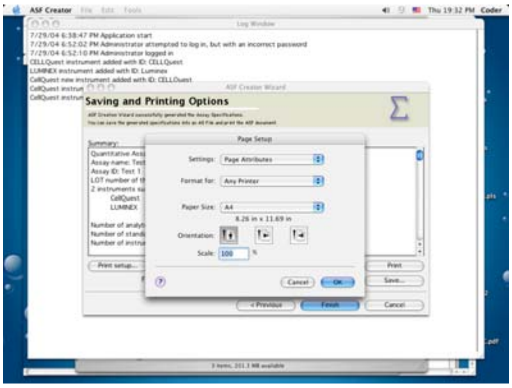


Figure 1.5.22: Page setup for printing

Firstly the folder has to be specified with standard browse panel, where the kit will be saved after the user pressed the Save... button. The kit name automatically expanded with the .kit extension (see Figure 1.5.23).

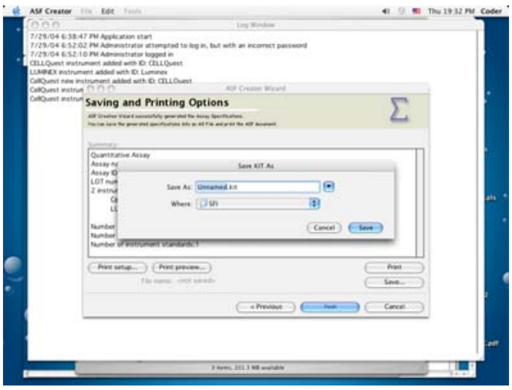


Figure 1.5.23: Saving the kit

When the kit is saved, the Save... button is greyed. A kit can be saved only once (see Figure 1.5.24).

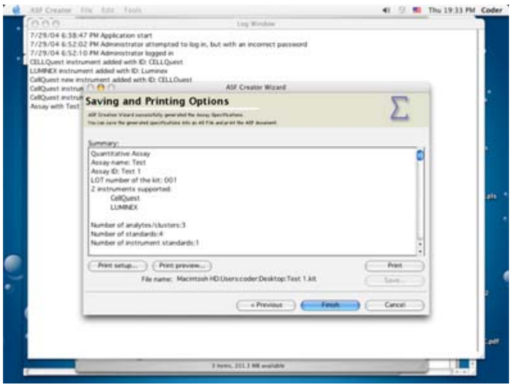


Figure 1.5.24: Last page of the wizard, the kit is already saved (button is greyed)

Then the kit can be printed or the user can decide to completes the kit by pressing the finish button. If the kit is not saved yet, the user is prompted a warning message that there is no original printout of the kit. Printing is not obligatory, but strongly recommended (see Figure 1.5.25).

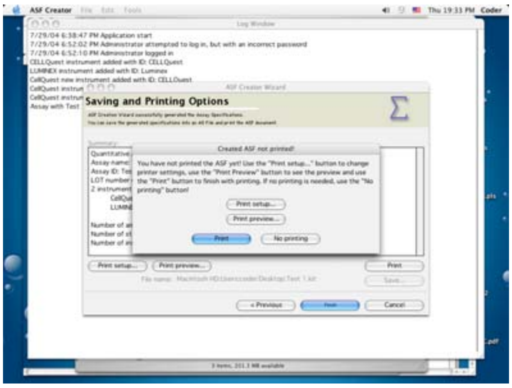


Figure 1.5.25: Warning that no original copy is printed of the kit

1.5.2 Qualitative model

After the wizard page for giving the analyte and bead pairs, the qualitative model has a page for specifying the data necessary for the model (see Figure 1.5.26). The qualitative controls have similar role to the standards of the quantitative model. They determine the parameters of a function that is necessary to compute the Cut-Off Indices (COI).

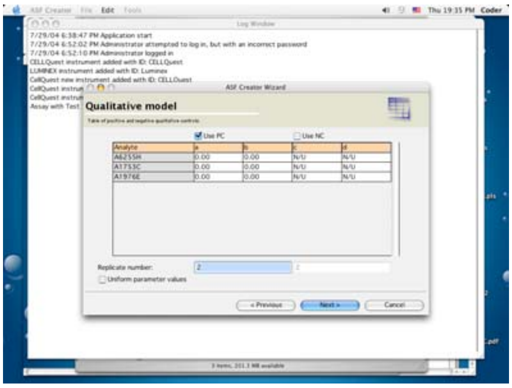


Figure 1.5.26: Data for the qualitative model

The user can define the positive and negative control samples separately. The columns for parameters a and b belong to positive controls. The same way columns for parameters c and d belong to negative controls. It is common that only one kind of the control samples is defined in a kit. The replicate number of the control samples are given respectively. It is possible to use uniform values for speeding up the process.

The next two steps are identical to quantitative case. The user has to give the reporting messages for the analytes and define the instrument standard beads similar to the quantitative branch. The most important difference is that these messages are based on Cut-Off Indices instead of concentration values. The final saving and printing process are the same, too.

After the kit is successfully saved, the log indicates that the kit is created and stored in the ASF creator's database (see Figure 1.5.27).

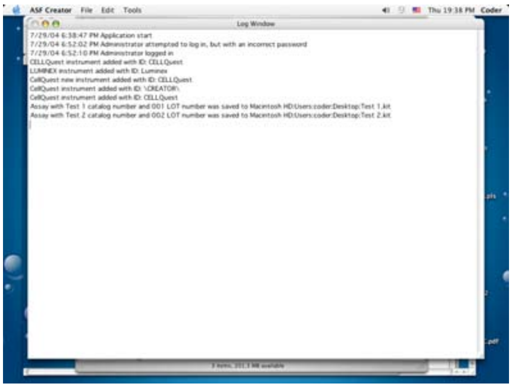


Figure 1.5.27: Log indicates that the second kit is saved

1.5.3 Functions of the ASF Creator Menus

We have extended the standard (File and Edit) menus and we have created a new Tools menu for our settings. The most frequently used options have keyboard shortcuts, too. The File menu was extended by many useful options (see Figure 1.5.28). The most important is the New ASF… option. This starts a new kit generation.

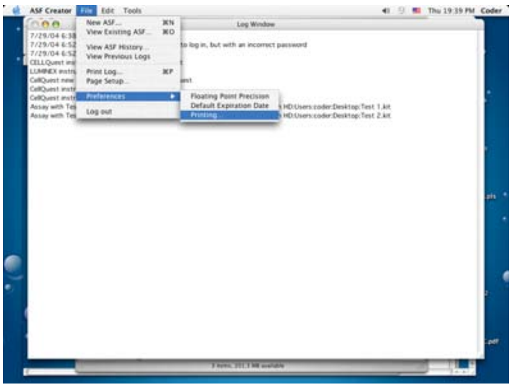


Figure 1.5.28: The File menu

The Logout option is essential, because for security reasons it is logged when the user started the program, what he did during his work period and when he closed the software. The history of kit creations can be reviewed by choosing the View Existing ASF... option (see Figure 1.5.29).

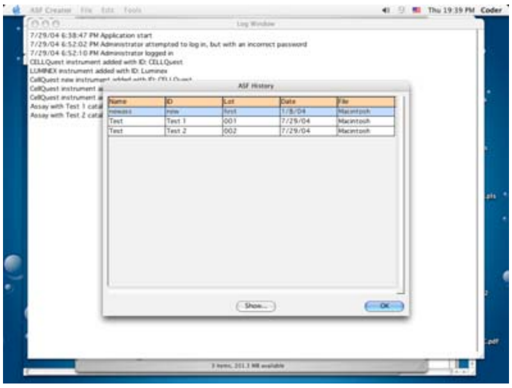


Figure 1.5.29: History of AS files

The most important data of the kits are displayed (name of the kit, ID, LOT number, creation date and the location of the file).

The Preferences option has three submenus. One for setting the precision of the printed results (see Figure 1.5.30).

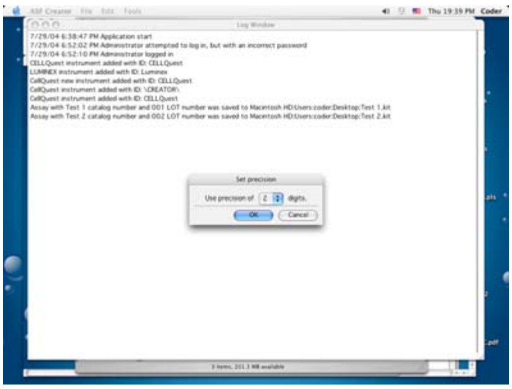


Figure 1.5.30: Precision setting

The expiration date will be the current date and the expiration interval defined here (see Figure 1.5.31). This value can be override in the wizard.

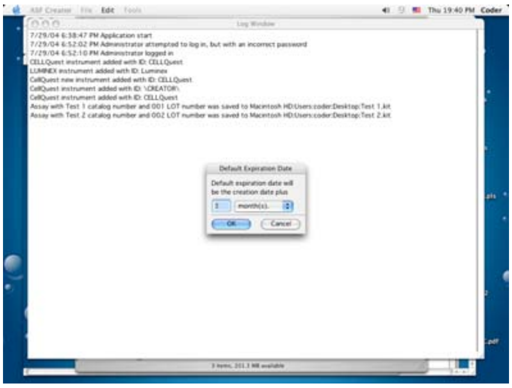


Figure 1.5.31: Setting the default expiration date

The third option helps to control the lookout of the printed results. The font type and size of the various captions (symbols, normal font, headings, etc.) can be set independently (see Figure 1.5.32). Of course, the default values can be reset easily with a button and effects of the modifications can be reviewed instantly.

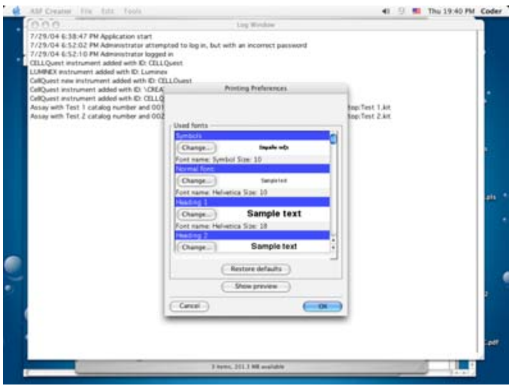


Figure 1.5.32: Setting the printing preferences

The Tools menu enables the user to manage the similar structured (bead, analyte, instrument, unit, assay) lists, the users of the program and their passwords (see Figure 1.5.33). In order to manage users and passwords, the logged in user has to be the administrator.

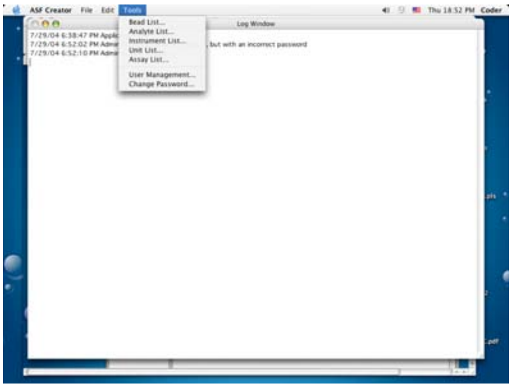


Figure 1.5.33: The Tools menu

The bead list and analyte list can be accessed from the Tools menu or from the Table of Analytes and Bead Clusters wizard page. The appropriate ID and description is shown in the tables (see Figure 1.5.34).

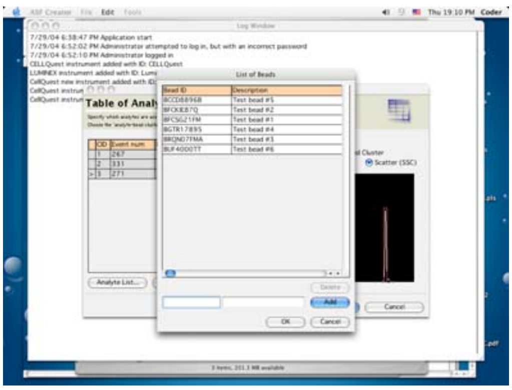


Figure 1.5.34: The list of beads

With the Add and Delete buttons the list can be extended or shortened. The elements of these lists make up the combo boxes for editing the cells of the tables in the wizard. The List of Analytes is shown on Figure 1.5.35.

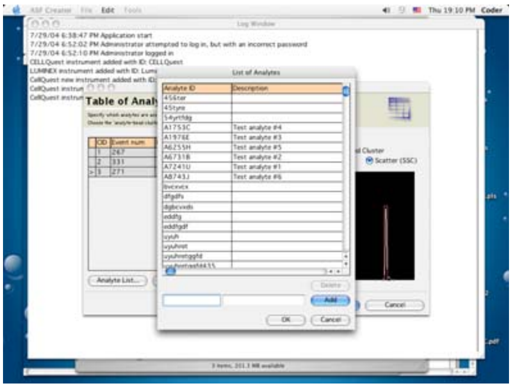


Figure 1.5.35: The list of analytes

The instrument list can be accessed from the Tools menu or from the List of Supported Instruments wizard page. Instruments are described by a name, an ID text, the SSC parameter, the clustering and the reporter parameters as shown in Figure 1.5.36.

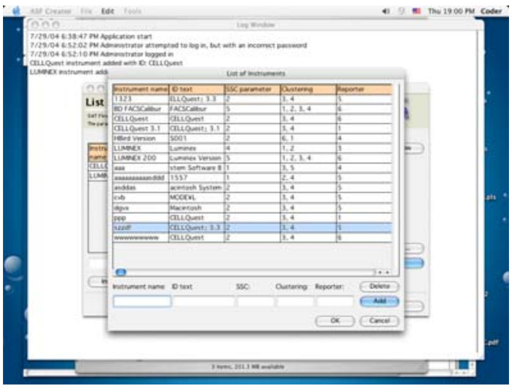


Figure 1.5.36: Dialog box for managing the list of instruments

There are predetermined concentration units that the user can be extend by new ones. The new concentration unit can be given by the unit and its description in the corresponding edit boxes. The not used concentration units can be deleted by selecting them and pressing the Delete button (see Figure 1.5.37).

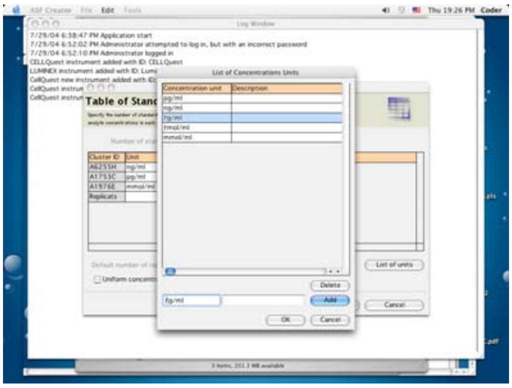


Figure 1.5.37: Dialog box for managing the used concentration units

The List of Assays menu item shows the created assays (see Figure 1.5.38). If there are too many, the unwanted ones can be deleted. Each user can change

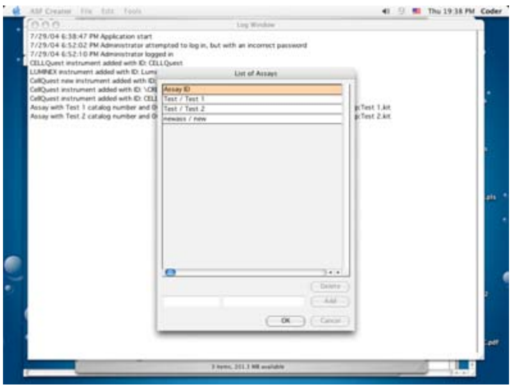


Figure 1.5.38: List of the created assays

his password with the help of the dialog box shown in Figure 1.5.39. It works the usual way, the user enters the old password, then fills in a new one which he has to retype. The two copy has to be correspond.

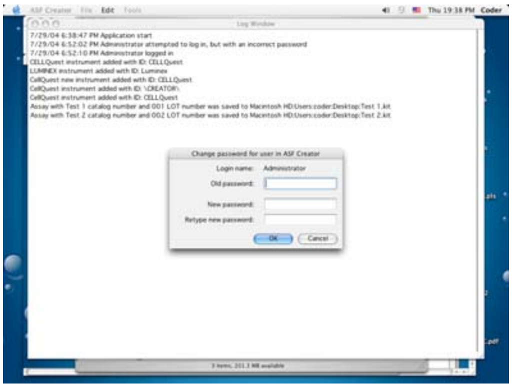


Figure 1.5.39: Change password dialog

Only the administrator can manage the users of the software (see Figure 1.5.40). The administrator can add new users to the software or delete old ones. Furthermore, he can change anyones password in case he should forgot it. Only one user, called 'Administrator' can not be deleted.

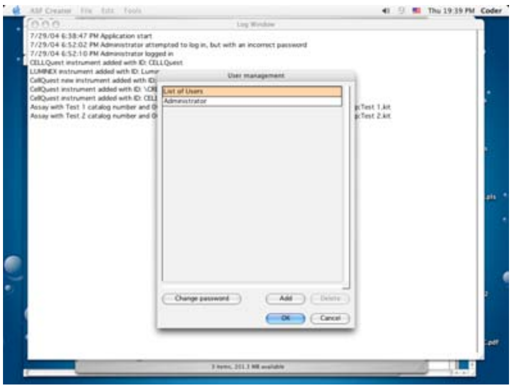


Figure 1.5.40: User management dialog

1.6 AS Files

The ASF Creator software yields an AS file that supports a kit. It contains:

- Type of the evaluation (qualitative, quantitative, IS)
- A name that describes the function
- Producer defined catalog ID
- Series identifier (LOT number)
- Unique identifier number
- Names of the supported cytometers
- Character series that identifies the cytometer in a List Mode file's header
- Stain (clustering) identifier parameters
- Size (SSc) identifier parameter
- Analytical (reporter) parameter
- Reactive (analyte) list in the order defined by the clustering procedure
- Bead list in the order defined by the clustering procedure
- Conditional messages' types, bounds and the texts of the messages

In quantitative case:

- The type of the function to be fitted
- Concentrations of the standard samples by analytes and the unit of the concentration

In qualitative case:

- Cut-Off Indices (COI) of the positive and negative control samples by analytes and the tolerance

In Instrument Standard case:

- Lack of the clustering parameters, stain list, control samples and conditional messages
- Relative Fluorescent Intensity (RFI) value for each bead

1.7 The SAT Flow Software

Steps of an experiment:

- The researcher decides the type of the experiment. He specifies the reactives and decides whether qualitative or quantitative measurement is needed. Instrument standards can be used to verify the integrity of the cytometer.
- The researcher selects the kit that satisfies his needs and places the order. He has to import the AS file of the kit into the SAT program. After importing the kit is ready for use in experiments.
- Assays of a kit can be used in different ways (see Figure 1.7.1).

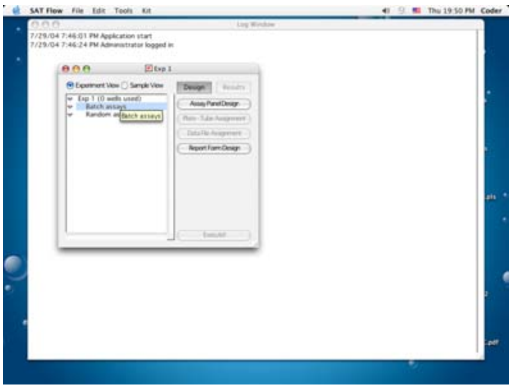


Figure 1.7.1: The experiment dialog box

  - Batch assay: A given test has to be applied for samples obtained from many different sources. The end user can place test samples in a Batch assay (see Figure 1.7.2).
  - Random assay: Used for patients (see Figure 1.7.3). Assays for the distinct tests have to be assigned to the appropriate patients. Then, many distinct tests can be applied to a given sample obtained from one source.

- A researcher can add and modify test samples to an assay (see Figure 1.7.4). The amount of the other samples is defined in the AS file. These samples that can not be modified are:

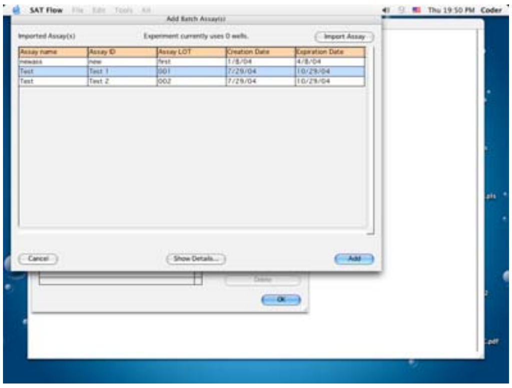


Figure 1.7.2: The Add Batch Assay(s) panel

– Standards - base points of the curve used for convert the intensity values into concentrations at quantitative evaluation. An intensity value associated with this sample in the LM file having known concentration gives the second coordinate of the base point.

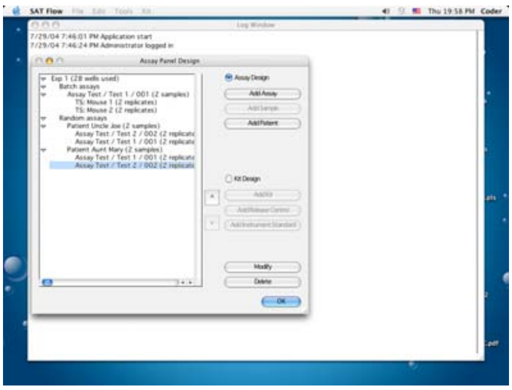


Figure 1.7.3: Patients are added to the Random assays

– Quantitative Control - used for verification, shall be in a range, else the evaluation process gives an error message.
– Qualitative Control - the qualitative controls have similar role to the

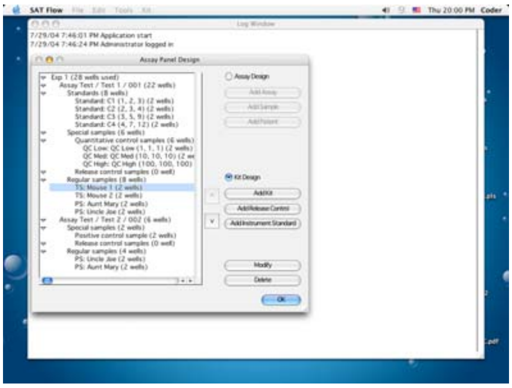


Figure 1.7.4: Regular samples can be modified

standards of the quantitative model. They determine the parameters of a function that is necessary to compute the Cut-Off Indices.

  – Instrument Standard - unstained calibrating beads. The usage of the instrument standards can be turned off.

- The samples can be arranged on a Microbead Array Plate, or on a Rack (see Figure 1.7.5). A plate has 96 or 384 wells in a matrix arrangement.

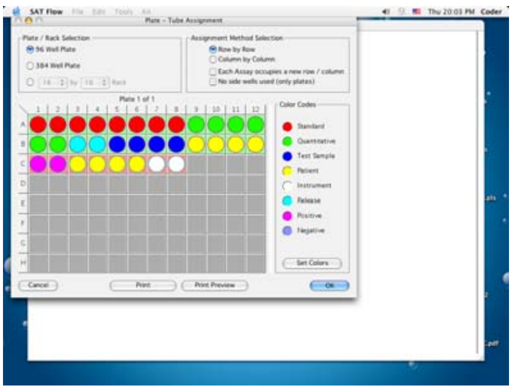


Figure 1.7.5: Plate - Tube Assignment

Rows are identified by letters, and columns are identified by numbers.

There are several possibilities to automatically arrange the samples. For better results, replicates can be asked for any sample. Each replicate has an own well in the plate. Then, a map can be printed for the technicians who will accomplish the measurements (see Figure 1.7.6). This map shows which tincture must inserted in which well. The LM files shall be created with the cytometer's own, special data storing system.

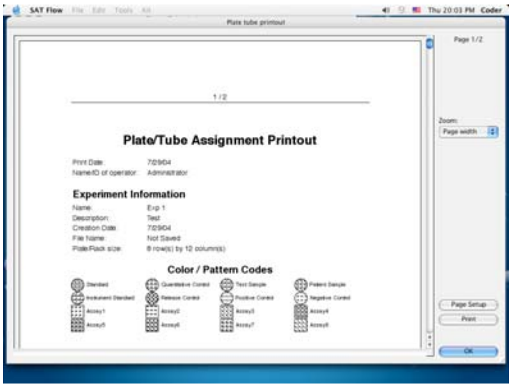


Figure 1.7.6: Plate - Tube Assignment Printout preview

- Figure 1.7.7 shows that Data File Assignment option can be used for asso-

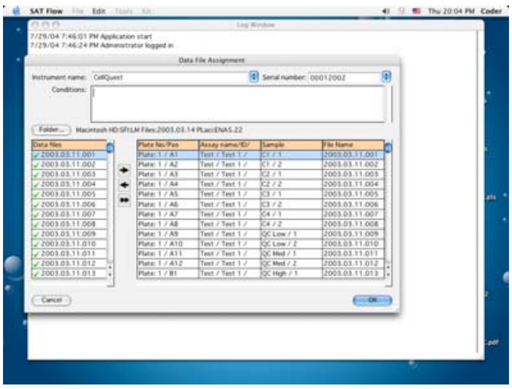


Figure 1.7.7: Assigned data files to samples

ciating assay replicates with List Mode files when LM files are copied into one folder. If the LM files have ordered names, then automatic assignment can be used. This assignment can be modified manually if necessary.

- The report's form and appearance farthest at this point must be decided. At this point can be fixed the object of interest and the level of detail. Besides, the arrangement of the measurement results can be set up. Reports that are generated with the software are typically 5 to 60 pages long. Two kind of report can be generated, a shorter 'Summary Report' and a detailed 'Full Report' (see Figure 1.7.8). The same options are used for both kind of reports.

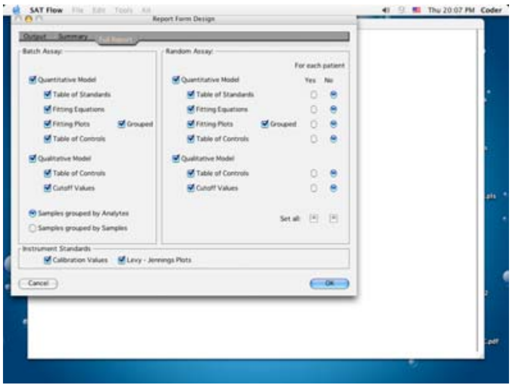


Figure 1.7.8: Report Form Design - Full Report panel

- After everything is set up, the evaluating procedure can be started (see Figure 1.7.9), so the List Mode files get processed. Clustering is done due to the device parameters fixed in the AS file, this means the selection of the events of one reactive. Intensity values of clusters are computed. The qualitative and quantitative methods use different evaluating procedure. In quantitative case, there is a curve fitting for the standards with the curve fixed in the AS file. With the help of the obtained curve parameters the software defines the concentration values of all other samples and classifies the control samples. In qualitative case, a Cut-Off Index is computed from the positive and negative control samples' intensity. The test samples' intensity values get divided with this Cut-Off Index. One control sample is mandatory, and the missing Cut-Off Indices will be zero values.

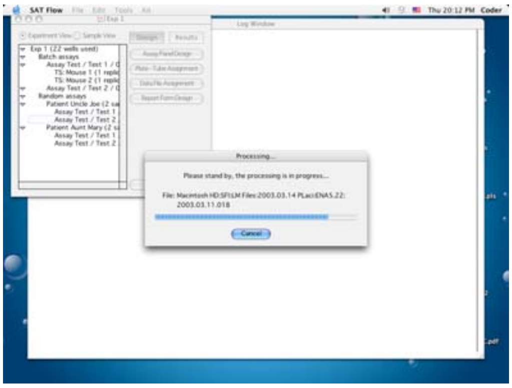


Figure 1.7.9: Evaluating process of the experiment

- Possible errors during the evaluation process:
  - The instrument cannot be identified with the help of the given character string at the pattern matching phase (see Figure 1.7.10).

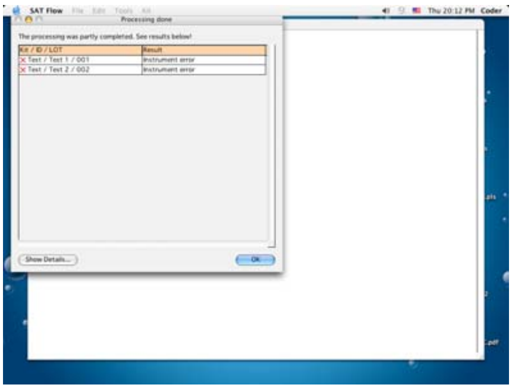


Figure 1.7.10: Feedback of the results

  - There is different number of clusters than should be.

- There is no proper curve to fit with the given function for the intensity values of the clusters at quantitative standards.

If the assays are free of these mathematical errors then the experiment can be evaluated biologically. The results are represented in a tabular form (see Figure 1.7.11). The most important results are the derived concentrations, the derived Cut-Off Indices and the deviation from the expected values at the control samples. With the help of the reporting messages the results of the test samples can easily be interpreted. All these tools together yield that a biologist researcher can plan and evaluate his experiment quickly and easily in a comfortable environment and gets the results in a form that can be read and interpreted easily. One of the greatest advantage of the software is that it automates the printed documentation, too.

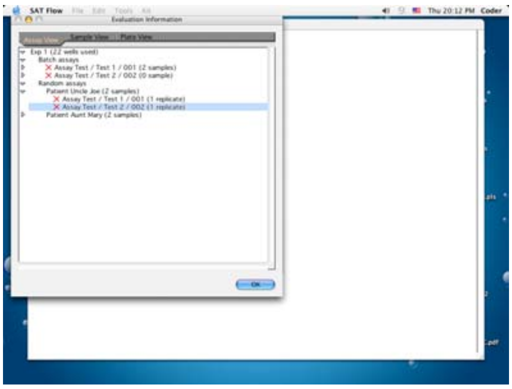


Figure 1.7.11: Details: Evaluation Information - Assay View panel

1.7.1 Functions of the SAT Flow Menus

Similar to the ASF Creator software, we have extended the standard File menu (see Figure 1.7.12) and we have created a Tools and a Kit menu.

The Preferences submenu has a new menu item, namely the Fitting Accuracy… entry.

The keyboard shortcuts are enabled for the standard functions of the menus. Those settings are grouped into the menus, which are independent from the process of evaluating an experiment. Some of the functions can be accessed from the experiment's windows, too.

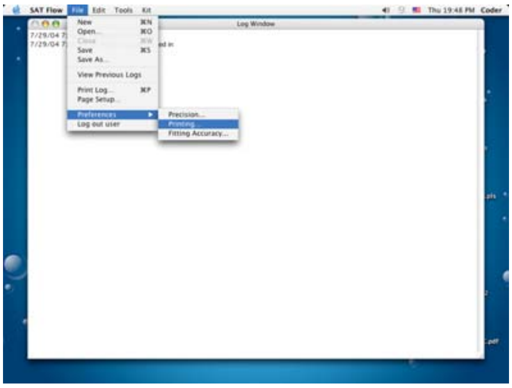


Figure 1.7.12: The File menu

The Precision... menu is similar to the ASF Creator's Floating Point Precision menu item (see Figure 1.7.13). It affects only the formatting of the printed floating point numbers. The computing is always done with full precision. Two digits after the decimal point is the default.

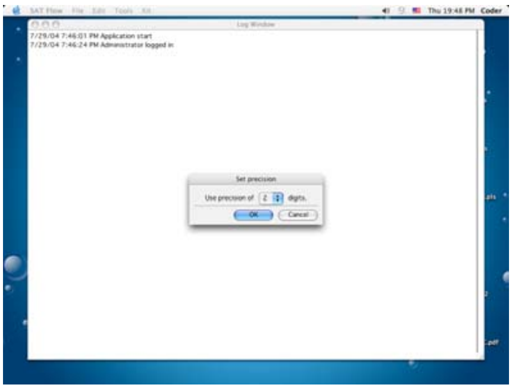


Figure 1.7.13: Setting the precision of the results

Fitting Accuracy influences the curve fitting process. The default value is

98%, but it can be modified by a slider or an edit box (see Figure 1.7.14). For satisfactory results it can not be less than 90%.

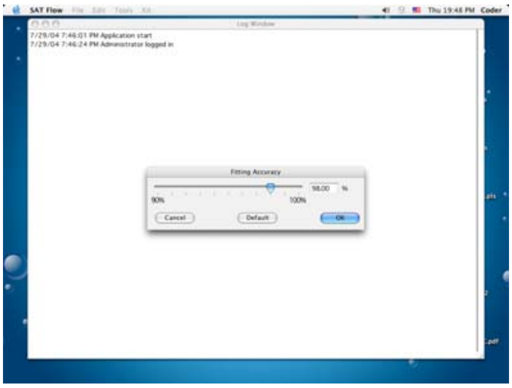


Figure 1.7.14: Setting the fitting accurancy

The Tools menu has four options. The User management... and Change passwords... menu items are identical to the ASF Creator software's options (see Figure 1.7.15).

The Color assignment... and Instrument calibration history... menu items are specific to the evaluation of the experiments.

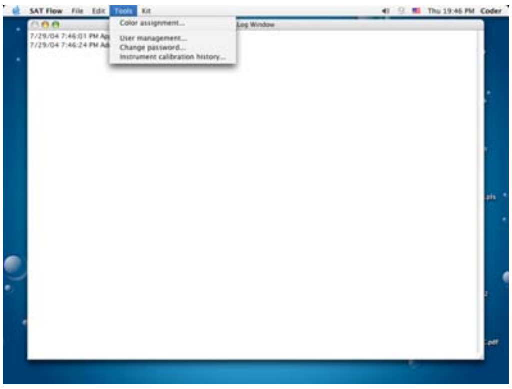


Figure 1.7.15: The Tools menu

The Color assignment... defines the colors for the various controls, samples and assays (see Figure 1.7.16). The samples are designated by circles, the assays are marked with rectangles.

This color code helps easily distinguish the samples and assays on the plate. For monochrome printers there is an individual pattern for each color.

The default colors can be reset at any time.

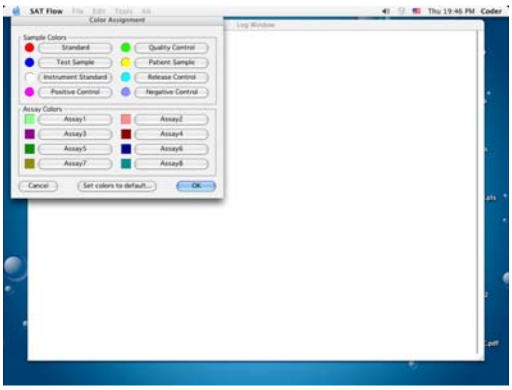


Figure 1.7.16: Color assignment for plate-tube assignment

Instrument Calibration History (see Figure 1.7.17) keeps track of the in-

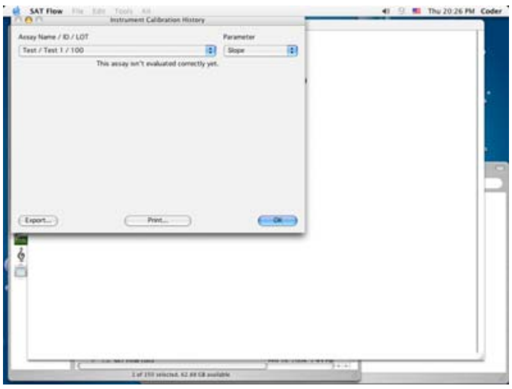


Figure 1.7.17: Instrument Calibration History

strument calibration samples. These samples have to be evaluated regularly to ensure that the instrument works properly.

The Kit menu serves the kit import and management (see Figure 1.7.18).

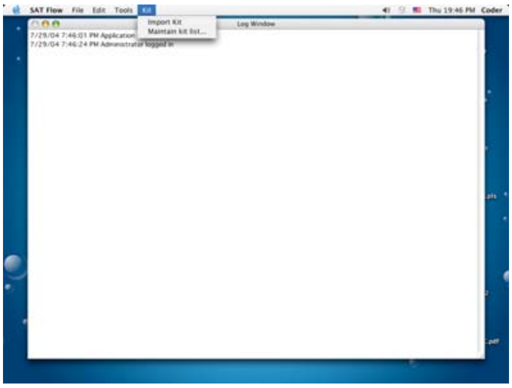


Figure 1.7.18: The Kit menu

After the kits are imported into the SAT Flow software for use they can be managed by the Kit list maintenance dialog box (see Figure 1.7.19).

The user can check the details of a kit, import a new one or delete the unnecessary kits.

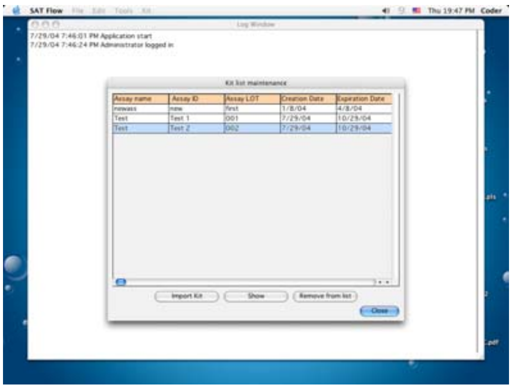


Figure 1.7.19: The Kit list maintenance dialog box

1.8 Current State of the Project

The two programs were united and enhanced by several new features. It is in use at Beckton Dickinson, they enclose it to their flow-cytometer machines.

Chapter 2

Cross Platform Developing

Since there are a lot of Apple Macintosh machines in the laboratories from the time when these kind of computers were the only choice, we had to support this platform, too. Nowadays IBM PCs are getting widespread even in this field. We have felt the wind of change, so the PC became the primary target.

Unfortunately, the Japanese partner changed the plans all over the time. This led us to the strategy to develop the PC version in advance and when a concept survived by time, then we have adopted the feature in the Mac version. There were tasks that could made parallel with the software planning and the early development stages. For example: getting to know the new platform and its philosophy, possibilities and disadvantages.

Originally, the project would have taken one year. The frequent changes led to longer developing time. In general, one can say that creating the executables for a new platform takes one man's work in a four men project.

2.1 Choosing the Programming Language and the Developing Tools

This step is one of the most important task. The selected developing tools determine the whole process of implementing the software plans. First of all, we have tried to choose the appropriate object oriented programming language. When the language was fixed, we had to solve the task of the version control and the archiving of the source codes and other resources.

2.1.1 Programming Language

Many aspects arise at choosing the right programming language. This is particularly true when the software has to support many platforms.

Although there are several programming languages, some of them, like Visual Basic, is not supported on most of the platforms. Microsoft has ceased to continue developing the Visual Studio product for Macintosh after version 4. On the other hand, the strongly object oriented Java has not satisfied our security and performance needs. Finally, C++ was the natural choice, because the shared classes needed only a few modifications (permanent rewrite of code pieces and conditional codes directed by macros) to get compiled on both platforms. Besides, the interface programming is very similar, but at this point we had some smaller and bigger problems. Furthermore, we had a lot of codes in C inherited from a former project, where LM files were processed. We could have these C codes very conveniently reused. Thanks to the highly optimized compilers and the power of the C++, the resulted executables are fast.

2.1.2 Developing Tools

After inspecting the existing developing tools, we have decided to use Microsoft's Visual Studio 6.0 on PC. Visual Studio has a solid basis and had a long enough evolving phase, it has relatively few errors and disadvantages. On the other hand, it has several advantage: program development has the support of MFC (Microsoft Foundation Classes) standard. The programs generated with the help of the MFC is not only programmed quicker, but furthermore they have the standard look of the Windows programs as well.

For the Macintosh platform there was no real choice. The standard compiler that comes with the operating system does not has support for huge projects' needs, e.g. resource handling including string tables for multiple languages, dialog boxes and icons. The Codewarrior integrated development environment (IDE) seemed to be the best solution. It has a class hierarchy that is similar to the MFC and has a complex resource handler. As Visual Studio yielded the standard look of a Windows program, Codewarrior yields the standard look of a Macintosh program.

The first version of our programs were made for version 8.5, and it has worked under OS series 9 the same way. Meanwhile the new version, the OS X has arrived. It has been rewritten from the sketch. It is based on a special Linux version, the Darwin. Not only the Pascal heritage was gone (this heritage with the compatibility issues are becoming the biggest problem nowadays), but the subsystems were recoded. If we give up supporting the older operating systems, then life gets much simpler. Unfortunately, there are a lot of older machines running older OS in the laboratories. So, we had to create an interface for the old graphical system and a little modified one for the new visual interface, called Aqua, that was totally rewritten.

2.1.3 Version Control and Archiving

When the programming language and the development environments were chosen, the version control and the archiving had to be solved. PC was the primary target, so we have used the Source Safe subsystem of the Visual Studio to keep track of the consecutively changing codes. Besides, this system can archive the project. We have archived everything related to the project each day, or even more when it was necessary. The archived files were saved onto CD regularly. In the case of a failure the whole project can be restored from the archived files.

Because the Macintosh version has followed the PC, version control was not necessary. We have decided to create zip archive of the project each day that was written to a CD, too.

2.2 Differences Between the Platforms

Historically, Apple has designed the first window based graphical system in the end of the '70s. Later, Microsoft and IBM started to develop a similar graphical user interface (GUI). The common feature is the usage of windows as basic elements.

A Macintosh program follows strict standards that are laid down in the Apple's Human Interface Guide (HIG) (see [9]). It covers many fields from psychology to the size of interface elements and screen sizes and resolutions. For example, in the first times the monitor was an integral part of the system and a specific graphic card controlled it, and only one resolution could have been used. This ensured that a window had the same physical extension on all Macintosh computers. In the present days this strict rules are getting less importance at software design.

Windows programs adapt the new techniques more easily. There is a noticeable difference in the user group of the different platforms. Macintosh users are more professionals in their fields or they do not care the technology itself and they are conservative in a way. Possible this is the reason for the highly standardized Macintosh platform.

2.2.1 Messaging Mechanisms

The messages are collected in message queues. The messages are handled by the appropriate functions of the system and the application. The two platforms have slightly distinct messaging mechanisms.

2.2.2 Differences Between Window Systems' Elements

An essential part of a Macintosh system is a mouse having only one button from the beginning. A Macintosh's mouse has only one button even nowadays. This led us to use only the left button of a PC's mouse, too. Only some less important feature can be reached with the right button that has a substitution of pressing the Mac button on the keyboard together with the button of the mouse.

2.3 Security Issues

Our programs are used for medical and research purposes, so security issues play a great role at the planning and programming phase. More person can use the ASF Creator and the SAT softwares on computers that have no or weak authentication systems, therefore our softwares shall handle these topics. The files saved to disks are encrypted, and only one original printed copy can exists of the AS or SAT files that are time-stamped and signed manually by the responsible person.

2.3.1 Login System

Before a user could do anything, he shall authorize himself. On PC this means that the application stays hidden and logs the attempt. On success, the application becomes visible and the user can start his job. Every significant step is logged and saved to an encrypted file. On failure, the application logs the failed attempt and quits without becoming visible for the user.

On Macintosh the toolbar changes when new application is started, so the menus have to disabled for the time of authentication. On success, these menus become available. On failure, the program exits without giving chance to modify anything. The logs are saved, too.

There is an administrator user having a hardwired password, this user can create new users or delete old ones. The administrator user can not be deleted. On request, the administrator can change a forgotten password for an arbitrary user. Our login dialogs have the standard edit boxes for passwords, they put an asterisk for each typed character. This makes easier to type in longer passwords by giving visual feedback for the accepted password characters.

2.3.2 Saving, Loading Subsystem

Saving and loading is standardized and work in the same way on different platforms. We have prepared the classes in a way, that the objects originated from the classes are able to save and load themselves. The strings are saved encoded by UNICODE for supporting more languages, and the numerical types follow the PC standards. The files are encrypted in order to protect the stored information. The files can not be edited easily with an ordinary text editor.

On loading the information from file, the appropriate functions check the integrity of the information. On failure the user is prompted an error message and the load procedure ends.

The numerical formats are handled specially on Macintosh, macros ensure the fast conversion between the platforms.

2.3.3 Printing

At printing one topic is very important: only one original printed copy shall exist signed by the responsible person. All other copies of the original are marked with caption 'Copy' on each page.

This means that printing an AS file or results of a SAT experiment saves the fact of printing in the file. After the original copy is printed, the following copies are all marked.

2.4 Interface Planning

During the planning phase we have kept in mind the standards and the most common solutions. Although, a few non-standard element had to be used. For example, we have developed an in-place editable grid for both platforms where the cells can be edited with an edit box or with a combo box, where the options are given.

2.4.1 The Wizard Technique

In the ASF Creator software there was a great freedom in the order of the tasks firstly. The options were grouped onto pages, and that pages were organized into a tab control, where the tab ears let the user change the option pages. Later, it became clear that it is advisable to set up a linear order to make sure that the user noticed all changes related to the last action. However, the tab control could be used in the SAT software very well (e.g.: Report Form Design or Evaluation Information dialog boxes).

With careful design, we managed to order the option pages in such a sequence, that a setting does not affect the previous pages' settings. This ordering made it possible to arrange the setting pages in a wizard. The main advantage of the wizard technique is the easy navigation through the pages with the next and previous buttons.

Besides, the user can be guided by not letting him going further if essential datum is missing. In such a case, clicking the Next button results a notification message box pointing out the problem that not allows the user leave the current page.

When the user steps back, he can overlook the settings he made so far. If he changes a setting, the change will only affect the current and the forthcoming wizard pages. It will not affect the previous pages' settings. The wizard will let the user to go to the next page only if he fills in the current page completely. The forthcoming pages will reflect the changes of the new setting. This way the user can supervise all induced changes through the wizard pages.

2.4.2 Level of Information

Our programs try to help the user by providing him with as many information as possible. The user is always faced only with the next necessary question that he can easily answer with the help of the continuous proper feedback of the project. Furthermore, the wizard of the ASF Creator has the description of the pages in the heading. Besides, where it is applicable, we used message boxes to indicate the exact problem.

The independent features can be set up from the menu directly too, not only from the page that needs the proper information. These features include managing the similar structured lists of beads, analytes and so on.

Some settings can be reached from the menu only, for example the used precision of displaying numerical data or the user maintenance.

2.4.3 Window Types

Windows systems have three basically different type windows: modal window, modeless window and float window. Macintosh systems have only one kind of window, but this window has many features. By setting these features in a proper way, all window types can be emulated.

2.4.3.1 Modal Windows

A modal window takes over the input events. This window handles all interaction with the user while the dialog box is active. This is what makes the dialog box modal; that is, the user cannot interact with other windows until the dialog box is closed. A modal dialog box requires the user to supply information or cancel the dialog box before allowing the application to continue. The wizard of the ASF Creator or the child windows of an experiment are modal windows. Modal windows usually do not have modeless child windows, they have float child windows instead.

2.4.3.2 Modeless Windows

A modeless dialog box allows the user to supply information and return to a previous task without closing the dialog box. The experiment windows of the SAT Flow are modeless dialog boxes. So, the user can handle more experiments parallel. Since an experiment is a huge task, a user is not supposed to work on more than one experiment, but we did not want to restrict the user in any way. Having an older experiment in memory can be useful. While the user has any open modal window of an experiment, he can not switch between the experiment windows.

2.4.3.3 Float Windows

This kind of windows are attached to a parent window and can be repositioned freely. Typically, the curves are drawn in float windows. They stuck to a modal window. When the parent window closes, all its float windows close.

2.4.4 Designing the Menus and the Icons

In ASF Creator, the File and Edit menus have the standard elements. The File menu has furthermore some element related to history and AS files. The Preferences option has some fine tuning settings (Floating Point Precision, Default Expiration Date and Printing...). The Tools menu helps maintain lists, change password or manage users.

SAT Flow follows the ASF Creator's customs. The File and Edit menus have the standard elements, but the File menu is extended with the Preferences and

View Previous Logs options. The Tools menu helps define the color assignment and handle user management, passwords and the instrument calibration history.

These considerations resulted a simple and easy to use menu systems. The floating toolbar and the icons are used only on PC. The icons yield a short way to access the functions of the program.

2.5 Localization

Localization means not only the change of the texts that appear on the screen and in printing, but it involves the order of a date's elements, the separators, the decimal point, etc.

2.5.1 Support for More Languages

There was a need to support more than one language, especially the Japanese. This means that the ASCII7 is not enough, a multi byte character set is needed. UNICODE seemed to be the solution, but the two platforms had distinct native support for this problem. Since the string tables are handled by the integrated development environments, this caused no problem. The string table that was set up in Visual Studio could be easily transformed into the CodeWarrior's string table. Only control sequences showed a little deviation.

2.5.1.1 MBCS

MFC specifically supports double-byte character sets (DBCS) from MultiByte Character Sets. Unicode is available only under Windows NT. We did not want a restriction in the type of Windows version, so this is why we decided to use MBCS that is available under all Windows version.

2.5.1.2 UNICODE

Unicode is a worldwide character-encoding standard that uses 16-bit character values to represent all the characters used in modern computing, including technical symbols and special characters used in publishing. Macintosh machines rely on standards much more than Windows based PCs. So, the CodeWarrior and its class hierarchy supports only UNICODE through the LString class.

2.5.2 Displaying Dates and Local Informations

Fortunately, there are standard ANSI C functions that format a date in the proper order of the elements, yield the name of the months in the local language, etc. We have extensively used these formatting functions.

2.6 Porting Problems and Solutions

Porting problems cover many questions that arise during the migration between the different platforms. From the byte order and representation standard of the float numbers through string handling to the classes of the development tools many difficulties has to be answered.

2.6.1 Byte Order

IBM PCs and Macintosh computers have different byte orders for floating point data types. This causes problem only if we want to store such a value in a file and then read back it from the file on other platform than it was originally saved.

With a simple macro this problem can be handled efficiently.

2.6.2 String Handling

Since even in nowadays the written text is the base of the remote human communication, strings are extensively used in computer programs. ANSI C functions helps the tasks related with strings. Visual Studio has a class named CString to collect these functions together, while Codewarrior serves the same purpose with LString.

2.6.2.1 CString

It is a well-written class providing the full functionality of the ANSI C string handling functions even with MBCS. The strings are null terminated and can have arbitrary length.

2.6.2.2 LString

Even the 9 series of the Macintosh operating system had a serious Pascal heritage, because compatibility issues back to the first Apple machine. The new

Mac OS 10 series, that is based on a Linux distribution named Darwin, hit the market when the development started already for months. The OS 10 and its developing tools are entirely based on the C language, but on the account of the compatibility with 9 series we chose to use CodeWarrior further. (Our market consists of research institutes and hospitals where there are a lot of older control computer that is not powered enough to run the new OS versions. This is why we kept the support for pre 10 series OS versions.)

In fact, the abstract class LString has a child class LString255 capable to store 255 characters where the first byte means the length of the string. We had to use this LString255 class, because the standard controls of the Macintosh interface use this kind of objects to communicate with. With the exception of some string of the dialog boxes, 255 characters are enough. The long strings have to be cut into more pieces.

2.6.3 Message Boxes

Message boxes are the basic form of the communication with user in decision-making situations. There are standard message boxes with different number of buttons in MFC on PCs.

- The message box contains one pushbutton: OK.
- The message box contains two pushbuttons: OK and Cancel.
- The message box contains two pushbuttons: Yes and No.
- The message box contains three pushbuttons: Yes, No, and Cancel.
- The message box contains two pushbuttons: Retry and Cancel.
- The message box contains three pushbuttons: Abort, Retry, and Ignore.

A message box can have different type of modality.

- Application modality: the user must respond to the message box before continuing work in the current window.
- System modality: all applications are suspended until the user responds to the message box. System-modal message boxes are used to notify the user of serious, potentially damaging errors that require immediate attention and should be used sparingly.

- Task modality: similar to Application modality, but not useful within a Microsoft Foundation class application. This flag is reserved for a calling application or library that does not have a window handle available.

Furthermore, various icons can be displayed next to the text (an exclamation-point, an icon consisting of an 'i' in a circle, a question-mark or a stop-sign).

I have implemented an adaptive message box class for Macintosh that resizes to the included text and has the proper buttons. Enumerated return value is used for notifying the caller function about the choice of the user. The dialog boxes are always modal to the application and they have no icon (our softwares use no icons in messages at all).

2.6.4 Tooltips

On Windows, programs frequently use tooltips to explain buttons, areas or to display hidden information in boxes where scrollbar would have been used (e.g. long elements of a narrow tree-control). On the contrary, Macintosh does not even has such a control.

We have found tooltips handy. By implementing a tooltip class for the Macintosh, we extended the possibilities. Our tooltip capable not only to show texts, but to display formulas or pictures when needed.

2.6.5 Printing

The printing strategy is very different on the two platforms. We have created a class that supports starting and ending a printing process. The class implements many methods from the primitives to print out texts or graphics to the complex functions as print out heading, footer or complex boxes arranged in an optimal manner.

The functions' implementations have directives for the compilers. Conditional compilation ensures the compact code for each platform. In a word one can say that the printing functions are part of a wrapper class.

2.6.6 Registry and Preferences

The programs usually have to store preferences that makes work easier. An option has to be set once, and next time the user starts the program, the customized interface comes up. (The preferences include decimal precision at displaying results, printing settings, color assignments, etc.)

Windows uses a central database called Registry. The programs are permitted to create entry points and numerous type data fields. On the other hand, Macintosh has a folder named Preferences where programs create a file with their names and store their preferences values in these files.

Similar to the printing system, we have created a wrapping class that manages the Registry entries on PC and writes to or reads from the Preferences file on Macintosh.

2.6.7 Compatibility of the Common Classes

The core or common classes caused only a few problem, because the C++ language platform independent and can be ported easily. Mostly one or two rows got into conditional compilation.

A few times the code had to be rewritten completely. For example, in some rare situations the PC version yielded very small numbers as result while the Macintosh version executed a division with zero divisor. This came from the different number representation standards. Introducing an epsilon value for comparing values together or with zero solved the problem entirely.

Chapter 3

Bitmap Transformations

The aim of my research is to create algorithms which are able to transform bitmaps in real-time (see [11], [12], [13], [14], [15]). The algorithms developed so far are able to accomplish the transformation of an $m \times n$ bitmap to either a convex polygon or a spherical surface or a cone patch or a cylinder patch or a free-form patch in real-time. These are simple projective mapping to the surfaces (see [16], [17], [18], [19], [20], [21]). In the literature there are several procedures that can be applied at the minor problems (see [22], [23], [24], [25]). In the sphere case, the projective co-ordinates of plane sequences are used instead of using sinus functions, which are very comfortable and fast. For the free-form case I have developed several techniques. The free-forms can be defined by well-parametrized B-spline curves, which I have examined in a reviewed article.

3.1 Notation and the Used Tools

Figure 3.1.1 shows the original bitmap that will be transformed to various shapes and surfaces. Let us indicate the vertices of this $m \times n$ bitmap with A, B, C and D. Let O_1 be the point of infinity of the straight line $\overline{AB}$ as shown in Figure 3.1.2. Let O_2 be the point of infinity of the straight line $\overline{AD}$. Let us consider an $m \times n$ bitmap as an $(m+1) \times (n+1)$ grid. Each pixel will be defined by four grid points. After the current transformation the area defined by the appropriate four transformed grid points have to be filled up with the color of the pixel. Applying this method to each pixel yields the transformed bitmap.

In the following sections I will handle Euclidean planes that are extended

Figure 3.1.1: The original bitmap (400 × 300)

with the points of infinity. I start from the finite part of the plane and I transform it to the finite part of the plane. Meanwhile the tools of the projective geometry will be used. A point of an Euclidean plane can be denoted by $P(x, y)$. We can introduce a new notation of P that will be the following: $P(\frac{x_1}{x_3}, \frac{x_2}{x_3})$, or $P[x_1, x_2, x_3]$. This way the points of infinity can be handled. When $x_3 = 0$, then the three co-ordinates describe a point of infinity. In this case (x_1, x_2) shows the direction of the point of infinity.

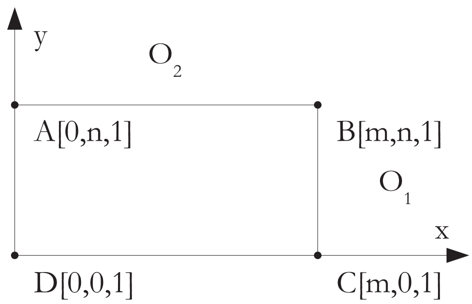


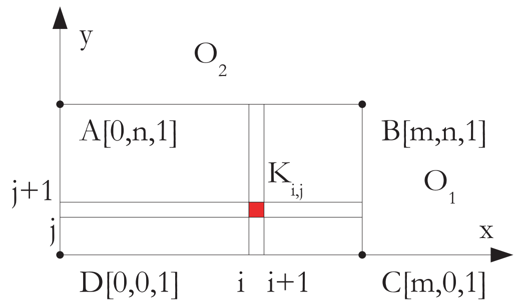
Figure 3.1.2: Notation of the vertices and points

3.2 Transformations Using Projective Invariants

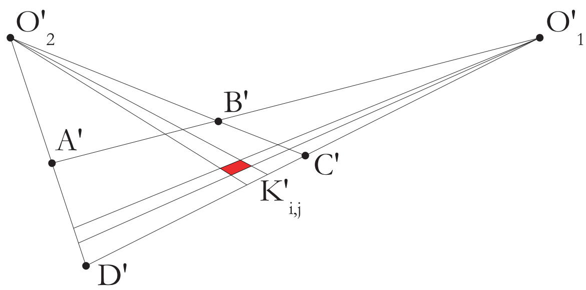
In this section I will describe projective mappings based on the invariance of the cross-ratio.

3.2.1 Transforming to a Convex Polygon

We have to find a quadrilateral for each $K_{i,j}$ pixel of the original bitmap (see Figure 3.2.1 (a)). When we know the position of the four vertices of the quadrilateral then we have to make a scan line conversion (see Figure 3.2.1 (b)).



(a) Original position



(b) Transformed position

Figure 3.2.1: Transformation of a texel

In Figure 3.2.2 the original bitmap can be seen transformed to a convex polygon.

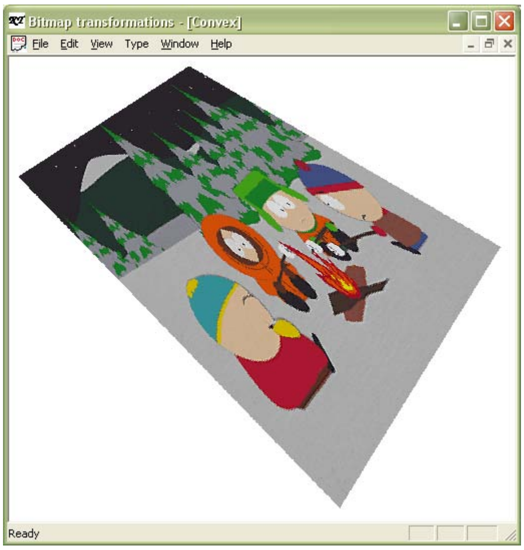


Figure 3.2.2: The original bitmap transformed to a convex polygon

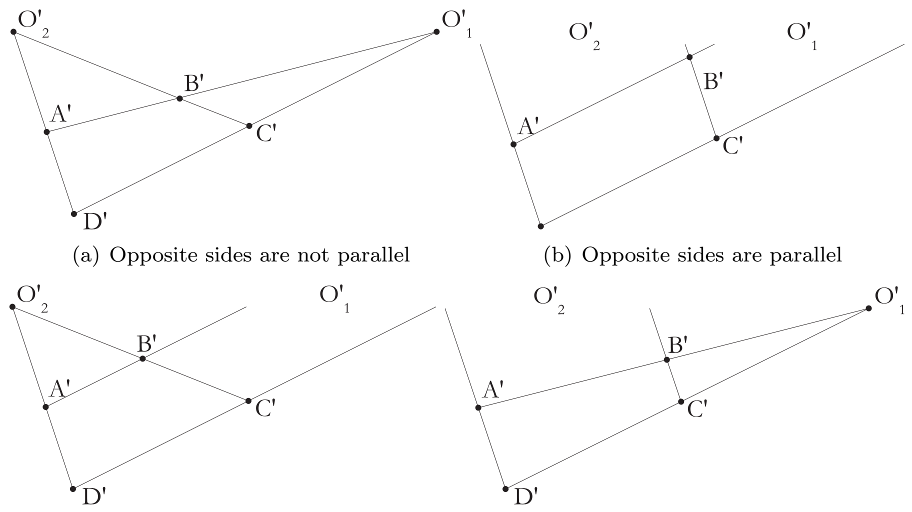
After the transformation of the bitmap to a convex polygon, we obtain one of the four cases shown in Figure 3.2.3.

The four vertices are points of intersections of lines which are defined by four pairs of points. The first pair of these points fit on the $\overline{A'B'}$ section, the second pair of these points fit on the $\overline{D'C'}$ section, the third pair of these points fit on the $\overline{A'D'}$ section and the forth pair of points fit on the $\overline{B'C'}$ section.

Let us indicate the actual grid-point of a border section with P and the transformed grid-point with P'. Since we can divide into n pieces the sections independently, we have to examine only two cases:

1. Opposite sides are parallel to each other.
   Let P a grid-point of the $\overline{AB}$ section. In this case we have to divide the $\overline{A'B'}$ and the $\overline{D'C'}$ section into n pieces according to the following:

$$\begin{aligned} (ABPO_1) &= (A'B'P'O'_1) \\ (ABP) &= (A'B'P') \\ \frac{\overline{PA}}{\overline{PB}} &= \frac{\overline{P'A'}}{\overline{P'B'}} \end{aligned}$$



(a) Opposite sides are not parallel
(b) Opposite sides are parallel
(c) $\overline{A'B'}$ and $\overline{D'C'}$ sides are parallel
(d) $\overline{A'D'}$ and $\overline{B'C'}$ sides are parallel

Figure 3.2.3: Possible cases after transformation

We get this result because O_1 and O_1' are points of infinity. This means that the length of the sections, defined by the transformed grid-points, will be the same.

2. Opposite sides are not parallel to each other.
Let P a grid-point of the $\overline{AB}$ section. The location of P' has to be defined according to the invariance of the cross-ratio. We get the following result from $(ABPO_1) = (ABP) = \frac{\overline{PA}}{\overline{PB}} = (A'B'P'O_1') = \frac{\overline{P'A'}}{\overline{P'B'}} : \frac{\overline{O_1'A'}}{\overline{O_1'B'}}$ and $\overline{P'B'} = \overline{A'B'} - \overline{P'A'}$ equalities: $\frac{\overline{PA}}{\overline{PB}} \cdot \frac{\overline{O_1'A'}}{\overline{O_1'B'}} = \frac{\overline{P'A'}}{\overline{A'B'} - \overline{P'A'}}$. From the last result we get $\overline{P'A'} = \frac{\overline{PA}}{\overline{PB}} \cdot \frac{\overline{O_1'A'}}{\overline{O_1'B'}} \cdot (\overline{A'B'} - \overline{P'A'})$ equality from which we want to express $\overline{P'A'}$. $\overline{P'A'} \cdot (1 + \frac{\overline{PA}}{\overline{PB}} \cdot \frac{\overline{O_1'A'}}{\overline{O_1'B'}}) = \overline{A'B'} \cdot \frac{\overline{PA}}{\overline{PB}} \cdot \frac{\overline{O_1'A'}}{\overline{O_1'B'}}$, so $\overline{P'A'} = \frac{\overline{A'B'} \cdot \frac{\overline{PA}}{\overline{PB}} \cdot \frac{\overline{O_1'A'}}{\overline{O_1'B'}}}{(1 + \frac{\overline{PA}}{\overline{PB}} \cdot \frac{\overline{O_1'A'}}{\overline{O_1'B'}})}$.

In this way we can define a P' point to each P point on the $\overline{A'B'}$ section. In the same way we can define the projected slide of the points of the $\overline{DC}$ section.

If we apply the intersecting function, which in fact does a vectorial multiplication between two points having projective co-ordinates, with the appropriate points of $\overline{A'B'}$ and $\overline{D'C'}$ sections, we get equalities of a line sequence. Intersecting these equalities with those we defined from $\overline{A'D'}$ and $\overline{B'C'}$ the same way, we get the projected slides of the points of the $m \times n$ grid.

3.2.2 Transformation onto Developable Superficies

Only those superficies can be handled that have known bijective developing function. First, we have to transform the four points defining the location of the transformed bitmap to the plane (on the basis of the lay-out of the surface), where we perform the definition of the points of the bitmap-grid with the help of the above-expounded method. After this step, we transform the points of the grid back to the surface. (The function of the lay-out of the surface is bijective, so that we can do back-transformation without any difficulty.) We have to fill up the quadrilaterals bordered by four points calculated this way with the proper colors of the bitmap. Thus these quadrilaterals give us the slide of the bitmap projected for example onto either cone of rotation (see Figure 3.2.4 (a)) or circular cylinder (see Figure 3.2.4 (b)).

(a) Cone of rotation (b) Circular cylinder

Figure 3.2.4: Transformation onto surfaces that can be laid out

3.2.3 Projective Transformation to a Polygon Having Orthogonal Symmetric Axes Pair

With the help of the result of the previous chapter, we can go on to transforming to a free-form patch. We can divide the bitmap into smaller partitions. The projection to these smaller partitions can be done according to the result of Section 3.2.1. This method works best when the number of the bitmaps' rows and columns are powers of two. This restriction can be satisfied easily by scaling the original bitmap. The transformed vertices of this partitions have to be defined so that for every two partitions having a common side, the following condition is true: the common side and the opposite sides to this common side must have the same point of intersection. As it can be seen in Figure 3.2.5, after the first one or two steps we cannot modify anything. In Section 3.3 alternative methods will be introduced for transforming to free-form patches.

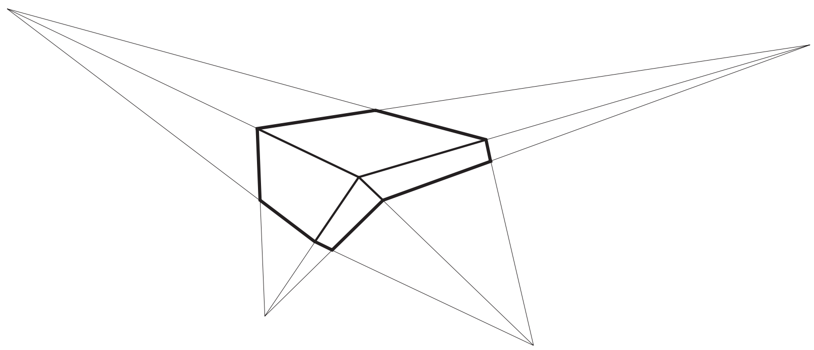
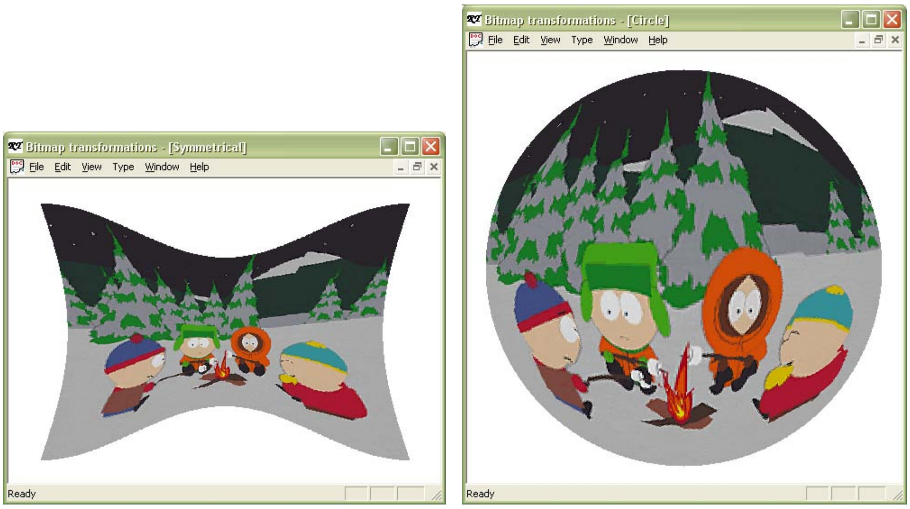


Figure 3.2.5: Partitions of the transformed bitmap

On the other hand, we can easily create special patches that are symmetrical longwise and crosswise as well and satisfy the above condition. During the partition, we can change the position of the vertices of the larger partition. After dividing this partition into four parts equally, the previous vertices and the point of intersection of the diagonals will be fixed. The position of the other vertices can be changed. If we apply this method to the four smaller partitions of the original bitmap the same way but symmetrically, then we defined the projection of the original bitmap to a symmetrical patch (see Figure 3.2.6 (a)). So this way we solved the problem of transforming for example to a circle (see Figure 3.2.6 (b)) or ellipse.



(a) General symmetric polygon (b) Circle

Figure 3.2.6: Transformation to symmetric polygons

3.2.4 Transforming to a Sphere

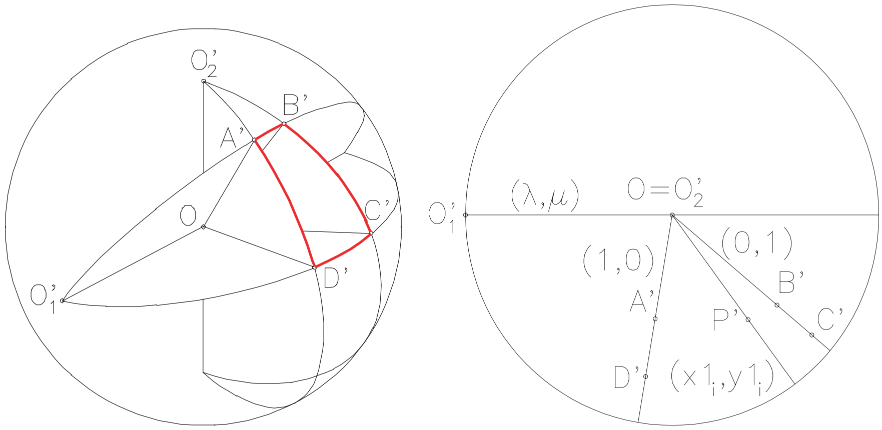
The situation of the transformed bitmap is defined by four points which have to be placed on the same half sphere. Thus, segments of main circles of the sphere will represent the parallel sides of the original rectangle. Indeed, it is the projective mapping of the plane to a half sphere. Let us indicate the transformed points with A', B', C', D', O_1' and O_2'. r will be the radius of the sphere. Let $\underline{a'}$ be the vector that points to A' from O which is the center of the sphere and the co-ordinate system, too (see Figure 3.7(b)). If we apply this notation to B', C' and D' points as well, we can obtain O_1' and O_2' for given A', B', C' and D' the following way:

$$\underline{o_1'} = (\underline{a'} \times \underline{b'}) \times (\underline{d'} \times \underline{c'}) \qquad \underline{o_2'} = (\underline{a'} \times \underline{d'}) \times (\underline{b'} \times \underline{c'})$$

$$\underline{OO_1'} = r \cdot \frac{\underline{o_1'}}{|\underline{o_1'}|} \qquad \underline{OO_2'} = r \cdot \frac{\underline{o_2'}}{|\underline{o_2'}|}$$

The spherical quadrilateral, on which the projection is accomplished, is defined by those segments of the spherical main circle which is the intersection of the plane determined by the center of the sphere and the proper vertical pairs and the sphere. Let us apply the notation shown in Figure 3.2.7 (a).

Let us consider that intersection of the sphere which is received so that it is perpendicular to the line of intersection of the $A'D'O$ and $B'C'O$ planes and let the O point, which is the center of the sphere, fit into this intersection. This intersection is shown in Figure 3.2.7 (b).



(a) The transformed vertices and points (b) The new co-ordinate system

Figure 3.2.7: Notations in the sphere case

In order to determine the transformed grid-points, we need two plane-sequences: one between the $A'D'O$ and $B'C'O$ planes and one between the $A'B'O$ and $D'C'O$ planes. The transformed grid-points will be the intersection points of the sphere and the two plane-sequences. This means a projective mapping between two first degree projective basic forms: between a plane-sequence and a point-sequence.

We can calculate a plane that will be element of the first plane-sequence and contains P' point for every P grid-point of the $\overline{AB}$ section. The same way we can calculate a plane that will be element of the second plane-sequence and contains P' point for every P grid-point of the $\overline{AD}$ section.

Let the basic planes of a projective co-ordinate system be the $A'D'O$ and

$B'C'O$ planes. $A'D'O$ will have $(0,1)$ projective co-ordinate values and $B'C'O$ will have $(1,0)$ projective co-ordinate values in the new co-ordinate system. The line of intersection of these basic planes will be the holder of the plane sequence with the help of which we can define the locations of the points transformed to the surface of the sphere. Let us express the $O_1'O_2'O$ plane having (λ,μ) co-ordinate values in the new co-ordinate system in terms of the basic planes. Let μ be 2. (We can do this because pairs having the same ratio define the same plane.) According to the equalities of the planes, λ can also be determined. We will get the proper λ value as well, if we use the normal vectors of the planes instead of the equalities of the planes.

$$\begin{aligned}\underline{n_{O_1'O_2'O}} &= \underline{o_1'} \times \underline{o_2'}\\ \underline{n_{A'D'O}} &= \underline{a'} \times \underline{d'}\\ \underline{n_{B'C'O}} &= \underline{b'} \times \underline{c'}\end{aligned}$$

Since $\underline{n_{A'D'O}}$ and $\underline{n_{B'C'O}}$ differ from each other at least in one co-ordinate value and μ is 2, the following three equations will contain at least two independent equations. There are two unknown values: k and λ.

$$\begin{aligned}k \cdot (\underline{n_{O_1'O_2'O}})_{(x)} &= \lambda \cdot (\underline{n_{A'D'O}})_{(x)} + \mu \cdot (\underline{n_{B'C'O}})_{(x)}\\ k \cdot (\underline{n_{O_1'O_2'O}})_{(y)} &= \lambda \cdot (\underline{n_{A'D'O}})_{(y)} + \mu \cdot (\underline{n_{B'C'O}})_{(y)}\\ k \cdot (\underline{n_{O_1'O_2'O}})_{(z)} &= \lambda \cdot (\underline{n_{A'D'O}})_{(z)} + \mu \cdot (\underline{n_{B'C'O}})_{(z)}\end{aligned}$$

From these equations k and λ can be determined easily. We are only interested in the value of λ.

Let an element of the plane sequence defined by $A'B'O$ and $D'C'O$ planes be indicated by $(x1_i, y1_i)$. We can produce this $(x1_i, y1_i)$ sequence if we take all P_i grid-point on the $\overline{AB}$ section, because the cross-ratio is invariant facing this type of projective projection. Thus $(ABP_iO_1) = (ABP_i) = \frac{\overline{P_iA}}{\overline{P_iB}} = \frac{\lambda-1}{\lambda-0} : \frac{x1_i-1}{x1_i-0}$. From this result $\frac{\overline{P_iA}}{\overline{P_iB}} \cdot \frac{x1_i-1}{x1_i} = \frac{\lambda-1}{\lambda}$ follows. We have to express $x1_i$ from $\frac{x1_i-1}{x1_i} = \frac{\lambda-1}{\lambda} \cdot \frac{\overline{P_iB}}{\overline{P_iA}}$. Since $x1_i \cdot (1 - \frac{\lambda-1}{\lambda} \cdot \frac{\overline{P_iB}}{\overline{P_iA}}) = 1$, so $x1_i = \frac{1}{1-\frac{\lambda-1}{\lambda}\cdot\frac{\overline{P_iB}}{\overline{P_iA}}}$. The same way $\frac{\overline{P_iA}}{\overline{P_iB}} = \frac{\mu-0}{\mu-1} : \frac{y1_i-0}{y1_i-1}$ and $\mu = 2$, so $\frac{\overline{P_iA}}{\overline{P_iB}} \cdot \frac{y1_i}{y1_i-1} = 2$. From the above equalities $y1_i = 2\cdot(y1_i-1)\cdot\frac{\overline{P_iB}}{\overline{P_iA}} = 2\cdot\frac{\overline{P_iB}}{\overline{P_iA}}\cdot y1_i - 2\cdot\frac{\overline{P_iB}}{\overline{P_iA}} = \frac{2\cdot\frac{\overline{P_iB}}{\overline{P_iA}}}{2\cdot\frac{\overline{P_iB}}{\overline{P_iA}}-1}$ follows.

If we consider $A'D'O$ and $B'C'O$ planes the basic planes, then the line of intersection of these planes will also be the holder of a plane-sequence. Let us indicate an element of this plane-sequence by $(x2_j, y2_j)$ parameters. So, the P'_{ij} points adequate to the P_{ij} grid-points of the original bitmap can be calculated as the point of intersection of the sphere and the line of intersection of the plane having $(x1_i, y1_i)$ parameters related to the first basic plane pair and the plane having $(x2_j, y2_j)$ parameters related to the second basic plane pair. During the producing of the algorithm, it is worth calculating with the normal vectors of the planes, because we can get the projected points of the grid points of the bitmap as the vectorial product of the normal vectors of the appropriate planes of the plane sequences having length r. We have to fill up the area having the vertices `Bitmap_Grid[i][j]`, `Bitmap_Grid[i][j+1]`, `Bitmap_Grid[i+1][j+1]` and `Bitmap_Grid[i+1][j]`, which are projected grid points, with the proper `Bitmap[i][j]` color, where `Bitmap_Grid` is an array of `Point_3D` elements and stores the transformed grid-points. At this point, if we have accomplished the previous operation on each i and j pairs that are different from each other and which fulfill the conditions, then we get the slide of the grid of the original bitmap projected to spherical surface (see Figure 3.2.8).

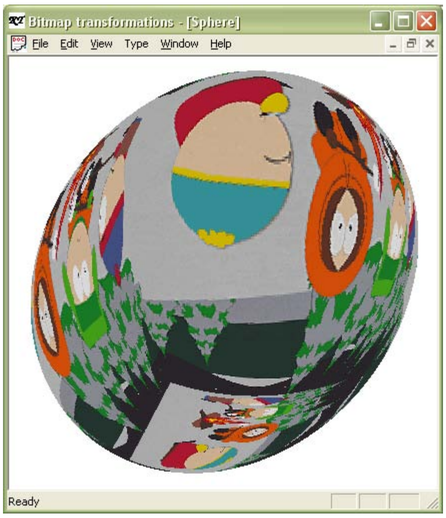


Figure 3.2.8: Six bitmaps transformed to a sphere

Consequently, we could transform to spherical surface taking advantage of the invariance of the cross-ratio without using trigonometrical functions in the algorithm, which are too time-consuming for computers.

3.3 Methods for Free-form Patches

When we take a non-symmetrical shape and do not want a break in the pattern, then we shall seek new methods. The following algorithms try to widen the shapes that can be handled.

3.3.1 Shrinking the Borders

In the sequel we suppose that the bitmap to be transformed is $N \times N$. It is not a restriction in practice, because each rectangular bitmap easily can be transformed into a $N \times N$ shape. Let the transformed borders given by $(x_1(t_1), y_1(t_1))$, $(x_2(t_2), y_2(t_2))$, $(x_3(t_3), y_3(t_3))$, $(x_4(t_4), y_4(t_4))$ function pairs, where $t_1, t_2, t_3, t_4 \in [0,1]$.

The points of intersections of the border curves defined by function pairs $(x_i(t_i), y_i(t_i))$ $(i = 1,2,3,4)$ are the followings: $(x_1(1), y_1(1)) = (x_2(0), y_2(0))$, $(x_2(1), y_2(1)) = (x_3(1), y_3(1))$, $(x_1(0), y_1(0)) = (x_4(0), y_4(0))$ and $(x_4(1), y_4(1)) = (x_3(0), y_3(0))$. Let $f_i(t_i) = (x_i(t_i), y_i(t_i))$ and $a_i(t_i, s) = f_i(t_i) + (KP - f_i(t_i)) \cdot s/(N/2)$, $i = 1,2,3,4$. Let KP denote the point of intersection of the diameters defined by the transformed vertices of the original grid. The diameters divide the form into four parts. The following constraints have to be satisfied for each $f_i(t_i)$: neither $f_i(t_i)$ can have a point of intersection with the diameters and each $f_i(t_i)$ must has one and only one point of intersection with those lines which fit on KP and lie in the appropriate angular region. Changing any of the $f_i(t_i)$ will only affect the appropriate quarter of the image.

The transformed edge points of the original grid are the followings:

$$\begin{array}{lllll}
(x_1(0), y_1(0)) & (x_1(1/N), y_1(1/N)) & (x_1(2/N), y_1(2/N)) & \dots & (x_1(1), y_1(1)); \\
(x_2(0), y_2(0)) & (x_2(1/N), y_2(1/N)) & (x_2(2/N), y_2(2/N)) & \dots & (x_2(1), y_2(1)); \\
(x_3(0), y_3(0)) & (x_3(1/N), y_3(1/N)) & (x_3(2/N), y_3(2/N)) & \dots & (x_3(1), y_3(1)); \\
(x_4(0), y_4(0)) & (x_4(1/N), y_4(1/N)) & (x_4(2/N), y_4(2/N)) & \dots & (x_4(1), y_4(1)).
\end{array}$$

The transformed inner grid points can be obtained by giving the parameters of functions a_i. These points will be defined from the edge moving toward KP. In each step less and less points by two have to be defined.

The parameters for a quarter are:

$$\begin{array}{ccccc}
a_i(0,0) & a_i(1/N,0) & a_i(2/N,0) & \dots & a_i(1,0); \\
a_i(0,1/(N/2)) & a_i(1/(N-2),1/(N/2)) & a_i(2/(N-2),1/(N/2)) & \dots & a_i(1,1/(N/2)); \\
a_i(0,2/(N/2)) & a_i(1/(N-4),2/(N/2)) & a_i(2/(N-4),2/(N/2)) & \dots & a_i(1,2/(N/2));
\end{array}$$

$$\ldots$$

$$a_i(0,(N/2-1)/(N/2)) \quad a_i(1/2,(N/2-1)/(N/2)) \quad a_i(1,(N/2-1)/(N/2));$$
$$a_i(1,1).$$

The common points of the neighbouring quarters will be computed twice and the point of intersection of the diameters will be computed four times in this way. Naturally, it is enough to compute these points only once.

The advantage of this method is that the transformed image can be modified locally. Changing only one bordering curve implies the recalculation and redrawing of only one quarter of the image. The four quarters of the image are distorted according to the bordering curves. The quarters are independent from each other, but the projection is continuous at the common edge of the quarters (see Figure 3.3.1).

Figure 3.3.1: The original bitmap transformed to a free-form patch

3.3.2 Using Bilinear Weight Function

Transforming to a free-form patch can be interpreted the following way: two function sequences have to be defined between two pair of functions and we have to use the points of intersections of these function sequences (see [27]). The elements of the function sequences will be obtained similarly to the functions of a Coons-patch which is fitting the quadrilateral defined by the four functions. Let the original bitmap be $N \times M$.

Let $a_1(u)$, $a_2(u)$, $u \in [0,1]$, and $b_1(v)$, $b_2(v)$, $v \in [0,1]$ curves be given so that they are in the same plane and both a curve have one point of intersection with both b curves. These curve pairs will define the transformed bitmap. Our objective is to find a $s(u,v)$, $u,v \in [0,1]$ function for which $s(u,0) = a_1(u)$, $s(u,1) = a_2(u)$; $s(0,v) = b_1(v)$, $s(1,v) = b_2(v)$ are satisfied.

We will use function sequences to solve this problem. Let us consider $I_a(u,v) = (1-v)\cdot a_1(u) + v \cdot a_2(u)$ function sequence defined by $a_1(u)$ and $a_2(u)$ and let us consider $I_b(u,v) = (1-u)\cdot b_1(v) + u\cdot b_2(v)$ function sequence defined by $b_1(v)$ and $b_2(v)$. These function sequences are interpolating the opposite curves but their end points do not follow the other curve pair. The bilinear interpolation of the four vertices will solve this problem (see Figure 3.3.2).

$$I_{ab}(u,v) = \begin{bmatrix} 1-u & u \end{bmatrix} \cdot \begin{bmatrix} s(0,0) & s(0,1) \\ s(1,0) & s(1,1) \end{bmatrix} \cdot \begin{bmatrix} 1-v \\ v \end{bmatrix}$$

Figure 3.3.2: Transformation based on bilinear weight function

The $s(u,v) = I_a(u,v) + I_b(u,v) - I_{ab}(u,v)$ function yields the solution of the problem. The transformed grid can be obtained by using the following parameters:

$$\begin{bmatrix} s(0,0) & s(0,1/M) & s(0,2/M) & \dots & s(0,1) \\ s(1/N,0) & s(1/N,1/M) & s(1/N,2/M) & \dots & s(1/N,1) \\ & & & \dots & \\ s((N-1)/N,0) & s((N-1)/N,1/M) & s((N-1)/N,2/M) & \dots & s((N-1)/N,1) \\ s(1,0) & s(1,1/M) & s(1,2/M) & \dots & s(1,1) \end{bmatrix}.$$

Unfortunately, we have a problem at this point. The $s(u, v)$ function can yield such co-ordinate pair for inner points that defines a point that is out of the area defined by the four curves which define the mapping. We have to take into consideration this fact at giving the bordering curves.

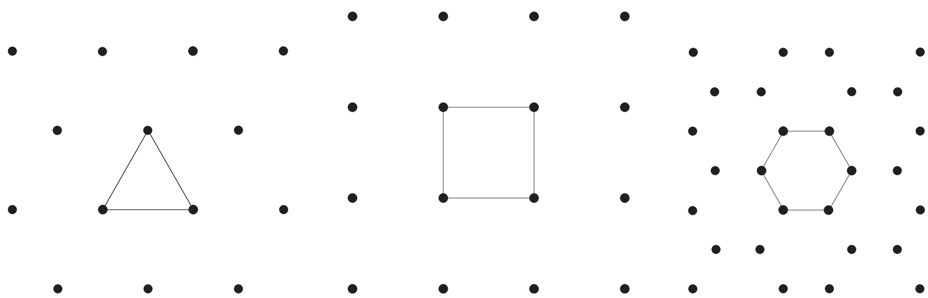
Changing the borders effects the whole picture, so we have to recalculate all of the points of the transformed grid. In spite of this fact, we can speed up the algorithm by reusing partial results.

3.4 Using Neural Networks

This section will present a new method based on neural networks which performs bitmap transformation onto a free-form patch. Earlier methods have pros and contras. In several cases they can be used very effectively, but in some cases they yield unacceptable result. An important advantage of this new method that arbitrary topology can be chosen for a pixel (as might as well hexagon or triangular instead of the usual quadrilateral, see Figure 3.4.1), because this topology remains fixed during the algorithm. On the other hand, it has a big disadvantage considering the solutions so far: this method is rather slow, thus making animations is not possible.

3.4.1 Introduction into Neural Networks

Having the theorem of neural networks been developed, many mathematicians thought this new approach will solve many problems they could not handle before. After the beginning enthusiasm people have calmed down. Although this new approach gave valuable results in many fields, it did not solve all of the problems. In a word, the theorem of artificial neural networks yields a very effective alternative method (see [28], [29]). The learning process in most common neural nets evolves sequentially over time. This process of learning is also combinatorial and nonlinear in nature. Theoretically, there is no guarantee



(a) Triangular topology (b) Quadrilateral topology (c) Hexagonal topology

Figure 3.4.1: Different topologies

for the existence of a solution to a nonlinear differential equation at all times, let alone the complicated task of solving them. In practice however, there exists a variety of techniques and tools that analyze the existence of stable states in nonlinear systems, without actually the need for solving any differential equation.

In the followings a Kohonen (see [30], [31], [32], [33]) network will be used because of its strong ability to learn data. Besides, the Kohonen network has another desirable important facility, which is the ability to keep the original topology of a predefined grid during the training procedure.

The approximation algorithm consists of two separated parts.

1. The neural network is taught by random points and by this teaching procedure prepares a grid of a chosen topology. The starting grid points will spread over the free-form patch.

2. The areas corresponding to the pixels have to be filled up with the appropriate color. This can be done by a scan-line algorithm or we can use OpenGL which is a platform independent system.

3.4.2 The Training Procedure

The network will be trained by points selected from a uniformly distributed area, see [30]. We will generate random points of the free-form patch that will be the input vectors of the network. In order to obtain the desired form, we put the points of the border in regularly. Since our problem is two-dimensional, the input layer consists of two nodes. The output layer consists of $m = (k+1)\cdot(l+1)$ where k and l are the number of the rows and columns of the original bitmap. These m output nodes form a grid. However, the topology of this grid can be arbitrary, in our case a quadrilateral grid is required. Each nodes of the input layer is connected to each nodes of the output layer. All of the connections have a weight: w_{ij} denotes the weight between the input node i and output node j. Figure 3.4.2 shows the desired form after the weights do not change. The form is given as a B-spline, in the figures you can notice the control points of the B-spline curve.

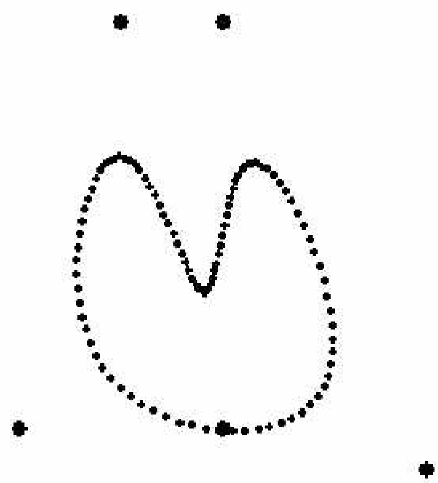


Figure 3.4.2: The desired form

Notation:

- Coordinates of the input points are: $p_i(x_{1i}, x_{2i})$ $(i = 1, \ldots, n)$, where n is the number of input vectors.
- Coordinates of the output points are: $q_j(w_{1j}, w_{2j})$ $(j = 1, \ldots, m)$, where m is the number of the nodes in the network.

The training procedure is defined in the following way:

Step 1. Initialize the weights w_{sj}, $(s = 1, 2; j = 1, \ldots, m)$ according to the chosen topology. Let the training time $t = 1$. Figure 3.4.3 shows initial weights.

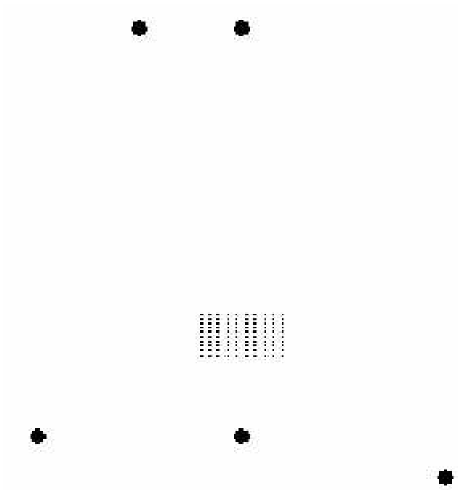


Figure 3.4.3: Initial weights

Step 2. Present new input values (x_{1i_0}, x_{2i_0}), as the coordinates of a randomly selected input point p_{i_0} from the free-form patch. Figure 3.4.4 (a) shows random input points. We take a point from the border regularly, too (see Figure 3.4.4 (b)).

Step 3. Compute the Euclidean distance of all output nodes to the input point:

$$d_j = \sum_{s=1}^{2} (x_{si_0} - w_{sj})^2.$$

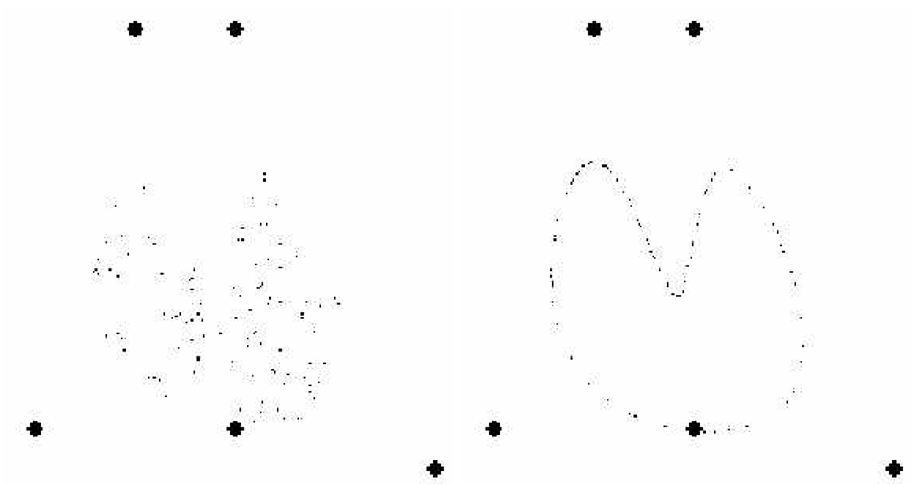
Step 4. Find the winning unit q_{j_0} as the node which has the minimum distance to the input point, so where j_0 is the value for which $d_{j_0} = min(d_j)$.

Step 5. Compute the neighborhood $N(t) = (j_0, j_1, \ldots, j_k)$.

Step 6. Update the weights of the nodes in the neighbourhood by the following equation:

$$w_{sj}(t+1) = w_{sj}(t) + \eta(t) \cdot (x_{si_0} - w_{sj}(t))^2 \quad \forall j \in N(t),$$

where $\eta(t)$ is a gain term decreasing in time.



(a) Random inner points (b) Random border points

Figure 3.4.4: Input points

Step 7. Let $t = t + 1$. Repeat Step 2-7 until the network is trained (see a run result on Figure 3.4.5). This means that the change of the grid is under a predefined limit.

3.4.2.1 Initializing the Weights

First of all, the weights of the network have to be initialized. They have to be determined in a way that the starting grid points form a rectangle within the free-form patch.

Run results show that different types of patches need different initializations. One of the simplest way is to set the weights to small random values. Best results can be achieved by arranging the weights around the centroid of the patch.

3.4.2.2 The Radius and the Gain Terms

Considering the computing time the radius and the gain terms are important factors. Both of them have to decrease in time and they shall be Gaussian. In

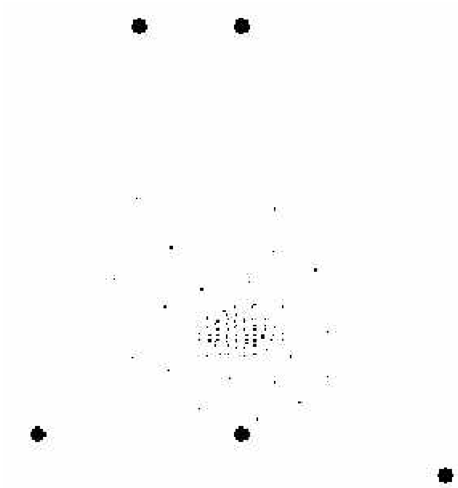


Figure 3.4.5: A run result

the algorithm I have used the following formulas to calculate the radius and the gain term, see [32] and [33]:

$$Gain(t) = \frac{1}{\sqrt{2\pi}} e^{-\frac{1}{2}(\frac{t}{q})^2} \quad Radius(t) = \frac{n}{2} e^{-\frac{1}{2}(\frac{t}{s})^2},$$

where t denotes the number of iterations, q and s are the scales on the time axis, while n denotes the number of input points.

3.4.3 Features of Neural Network Solutions

3.4.3.1 General Features

- The method is not deterministic. Usually this feature is a basic requirement. We cannot reproduce the same result for the same input.

- The network oscillates around the optimal solution considering the current parameters.

3.4.3.2 Advantages

- This method is applicable when traditional methods fail. With the help of projective invariants it is possible to transform a bitmap to a quadrilateral, onto spherical, conic or cylinder surface. Besides, it is easy to create patches that are symmetrical longwise and crosswise as well. (For example a circle satisfies these conditions.) This special patches can be textured with the help of projective invariants also. With some restrictions it is possible to transform bitmaps to free-form patches by shrinking the borders or by using bilinear weight function. By using neural networks even more cases can be handled.
- Instead of the usual quadrilateral topology the pixels can be arranged in an arbitrary predefined way. For example the grid can have triangular or hexagonal topology which are popular in the field of image processing.

3.4.3.3 Disadvantages

- The algorithm is rather slow. It cannot be applied in real-time applications. Extra code is necessary to keep the grid points within the free-form patch during the teaching procedure.
- The border of the grid obtained by the neural network does not fit exactly the original curve. This makes it hard to texture forms that have common border. One solution could be if we firstly determine the transformed border points and only after this step begin to localize the inner points.
- Forms which border is heavily curved cannot be textured this way, either. The reason is the Euclidean distance. This problem might be solved by shaping the patch step by step starting from a rectangular shape. During this shaping we have to teach the network all the time. It is important to give in the actual border points in each step. This will guarantee that the transformed grid will follow the shape of the desired free-form patch. It is not obvious how to decrease the radius and the gain factor.

3.4.4 Concluding Remarks

By using artificial neural networks we get very nice transformed pictures, but unfortunately the speed is not satisfactory. The pixels from which the picture is made up have similar size if the parameters are chosen expediently.

During the tests it has turned out clearly that this method does not solve the free-form problem entirely, but helps in many cases where traditional concepts fail. It helps us to extend the set of free-forms which can be textured.

Chapter 4

Summary

4.1 MMA Technology

The greatest result is that by studying the customs and work procedures of the users of the MMA technology, we succeeded in making up and documenting such a method that can be supported by computer programs. What is more, we drew up the automation of evaluating of results. Furthermore, we worked out how to set up and arrange the results for printing and how to handling the authorization questions in a proper way.

The advantage of the MMA (Multiplex Microbead Assay) technology is that several reactives can be used in a measurement, provided that several stains are used with homogeneously mixed beads that are covered with different reactives. This way the necessary amount of sample (blood, plasma or tincture) will be significantly smaller. If only one stain is used with numerous concentrations, then maximum 10-16 beads can be distinguished. By using two channels' data two stains can be used, so the maximum number of distinguishable beads rise to 256. The number of distinguishable beads can be extended by using different sized beads.

Researchers can prepare their experiments and produce kits for them in advance with the help of the ASF Creator and SAT Flow softwares. Such a kit is a set of reagents and tinctures for the given experiment's type. The resulted description file of a kit is called Assay Specification file (abbreviated form: AS file).

The first conception was to have two programs. One for creating AS files and one for planning and evaluating experiments. This means two distinct groups of users. Only the producer of a kit is authorized to make changes to the AS files. This is why ASF Creator is protected by an encryption algorithm. Later, as the expectations have changed, Soft Flow Kft. has decided to unite these programs together. The resulted programs make it easier to plan and evaluate an experiment for biology researchers.

4.2 Cross Platform Developing

Several critical issues arose during porting the softwares onto Macintosh platform. It was my responsibility to find the most suitable solutions. I think that the whole team made its best and this way the ported programs are as similar to the PC versions as possible.

The chosen tools were satifying, they supported an appropriate team work and automated the archiving process. The C++ language was easily portable, its standard functions and classes could be used comfortably.

Some wrapping class had to be implemented (printing, preferences, etc.) and some control had to be ported to Macintosh (tooltip, editable table, message boxes, etc.).

4.3 Bitmap Transformations

Using the invariance of the cross-ratio I could transform a conventional box shaped texture onto a convex quadrilateral or onto various surfaces that have a lay-out to the plane. I have considered a bitmap as a quadrilateral grid in the plane. I have transformed this grid into a new position. I have defined the texels with the grid points in their vertices. The transformed texels yield the transformed bitmap together.

By developing further methods, the bitmap transformation became available for free-form shapes in real-time with some restrictions. The transformation for patches that are symmetrical in longitudinal and crosswise direction takes place in more steps. Firstly the center of the patch is textured, then the patch can be covered with the bitmap from the inner part to the border during the steps. The patches that have totally free-form shapes can be handled by shrinking the the borders of it or by using a bilinear weight function.

The neural networks helped me to texture further free-form shapes with less restrictions. According to my personal tests, unfortunately the developed transformation can not be done quick enough to maintain an animation in real-time.

Researchers published methods in connection with surface fitting, where they managed to transform a box shape in two steps onto a concave shape bounded by straight sections. By choosing enough small boundary sections, practically any shape can be textured on a screen that is made up from pixels. The formula is rather time consuming and since I have studied real-time transformations only, this result is beyond my scope.

Moore's law no longer seems to be true (see [26]). Nevertheless, I think that this kind of transformations will determine the future of bitmap transformations to free-forms.

Chapter 5

Összefoglalás

5.1 MMA technológia

Legnagyobb eredménynek azt tartom, hogy tanulmányozva az MMA technológiát használók szokásait, kialakult munkafolyamatait, sikerült kialakítani és dokumentálni egy olyan módszert, ami számítógépes programokkal támogatható. Ezen túl, elkészítettük az eredmények értékelésének automatizálását is, illetve az eredmények összeállítását, nyomtatását és azok hitelességi problémáit is kielégítő módon kezeltük.

Az MMA (Multiplex Microbead Assay = mikrogyöngy alapú multiplex analitika) technológia legnagyobb előnye az, hogy egy mérés során több reagenst használhatunk, ha különböző festékkel jelöljük meg a különböző reagensekkel fedett gyöngyöket. Ily módon a szükséges (vér, plazma vagy oldat) minta mennyisége jelentősen kisebb. Ha egy festéket használunk több különböző koncentrációban, akkor maximum 10-16 gyöngyöt különböztethetünk meg. Két csatorna adatainak használatával két festéket különböztethetünk meg, így a megkülönböztethető gyöngyök maximális száma 256-ra emelkedik. A megkülönböztethető gyöngyök maximális száma különböző méretű gyöngyök használatával terjeszthető ki.

A SAT Flow szoftver segítségével a kutatók megtervezhetik a kísérleteiket és az ASF Creator program segítségével kit-eket készíthetnek hozzájuk. Egy kit reagensek és oldatok készlete, ami az adott kísérlet típusához igazodik. A kit-ekhez mellékelt leíró állományt Assay Specification fájlnak (rövidítve: AS fájlnak vagy ASF-nek nevezzük).

Az eredeti elképzelés szerint két program készült. Egy (ASF Creator) az AS fájlok létrehozásához és egy (SAT Flow) a kísérletek tervezéséhez és kiértékeléséhez. Ez két különböző felhasználói csoportot jelentett. Csak a kit összeállítója volt jogosult módosításokat eszközölni az AS fájlokon. Ennélfogva az ASF Creator program egy titkosító algoritmussal védett. Később az elvárások megváltozásával a Soft Flow Kft. a programok egyesítése mellett döntött. Az elkészült programok nagymértékben megkönnyítik a biológiai témákban kutatók kísérleteinek tervezését és kiértékelését.

5.2 Keresztplatformos fejlesztés

Számos kritikus kérdés merült fel a szoftverek Macintosh platformra való portolásakor. Az én felelősségem volt, hogy a legjobb megoldásokat megtaláljuk. Úgy gondolom, hogy a teljes csapat a legjobb tudása szerint dolgozott, és a Macintoshra portolt programok annyira hasonlítanak a PC-s verziókra amennyire az csak lehetséges.

A választott eszközök kielégítően működtek, nagyon jó csapatmunkát tettek lehetővé, mindamellett automatizálták az archiválási folyamatot. A C++ nyelv könnyen portolható, a standard függvényei és osztályai pedig kényelmesen használhatók.

A munka során néhány burkoló osztályt kellett létrehoznunk (nyomtatási, beállítási, stb.) és néhány vezérlő elemet át kellett ültetnünk Macintoshra (tooltip, szerkeszthető táblázat, üzenet ablakok, stb.).

5.3 Bittérkép transzformációk

A kettősviszony invarianciáját kihasználva hagyományos téglalap alakú bittérképeket sikerült transzformálnom konvex négyszögre illetve olyan különböző felületekre, melyek síkba kifejthetőek. A bittérképeket síkbeli rácsnak tekintettem, és ennek a rácsnak a pontjait transzformáltam az új pozicíóba. A bittérkép texeleit a sarkaiban található rácspontokkal írtam le. A transzformált texelek együttesen a transzformált bittérképet szolgáltatják.

További módszerek kifejlesztésével a szabad határvonalú alakzatokra is lehetővé tettem bittérképek transzformációját valós időben bizonyos megszorítások mellett. A hossz- és keresztirányban szimmetrikus alakzatokra való transzformálás a tartomány fokozatos darabolásával oldható meg. Az alakzat közepe rögzül az első lépésben, majd belülről kifelé haladva teljesen lefedhető az alakzat a bittérképpel. A teljesen szabad határvonalú alakzatokat pedig a határvonal kicsinyítésével, illetve bilineáris súlyfüggvény alkalmazásával lehet kezelni.

A neurális hálózatok segítségével további, kevesebb megszorítással rendelkező szabad határvonalú alakzatokat is textúrázni lehet. Az általam kidolgozott módszer a személyes tesztek alapján már nagyon lassú, és így már nem alkalmas valós idejű animációk létrehozására.

Vannak olyan felület-illesztéssel kapcsolatos eredmények, melyek segítségével téglalap alakú területet lehet egy körön keresztül két lépésben megfeleltetni egyenes szakaszokkal határolt konkáv alakzatnak. Ha megfelelően rövid szakaszokat választunk a határvonalhoz, akkor gyakorlatilag akármilyen alakzat megjeleníthető egy pixelekből felépülő képernyőn. A képlet kiszámítása meglehetősen időigényes és mivel én a valós idejű transzformációkat kutattam, így ez az eredmény kívül esik a vizsgálatom tartományán.

Úgy tűnik, hogy Moore törvénye nem érvényes többé (lásd [26]). Mindamellett úgy gondolom, hogy a gépek fejlődésével az ilyen típusú algoritmusok fogják meghatározni a bittérképek szabad határvonalú alakzatokra való transzformációjának jövőjét.

References

[1] J. Dassoa, J. Leeb, H. Bacha and R. G. Mage: *A comparison of ELISA and flow microsphere-based assays for quantification of immunoglobulins*. Journal of Immunological Methods, Volume 263, Issues 1-2, 1 May 2002, Pages 23-33

[2] T. Lilliehorn, M. Nilsson, U. Simu, S. Johansson, M. Almqvist, J. Nilsson, T. Laurell: *Dynamic arraying of microbeads for bioassays in microfluidic channels*. Sensors and Actuators B-Chemical 106 (2): May 13 2005, Pages 851-858

[3] L. A. Ugozzoli: *Multiplex assays with fluorescent microbead readout: A powerful tool for mutation detection*. Clinical Chemistry 50 (11): Nov 2004, Pages 1963-1965

[4] V. Moalic, B. Mercier and C. Ferec: *Technologie Luminex^{TM} : principe, applications, et perspectives; Luminex^{TM} Technology: technical approach, applications and future prospects*. Immuno-analyse & Biologie Spécialisée, Volume 19, Issue 4, August 2004, Pages 181-187

[5] B. L. Ray, D. Ebner and I. Balazs: *Identifying HLA antibodies using luminex microspheres*. Human Immunology, Volume 63, Issue 10, Supplement 1, October 2002, Page S85

[6] Dario A. A. Vignali: *Multiplexed particle-based flow cytometric assays*. Journal of Immunological Methods, Volume 243, Issues 1-2, 21 September 2000, Pages 243-255, Flow Cytometry

[7] http://www.cytometry.com

[8] http://www.appliedcytometry.us

[9] Human Interface Guide http://developer.apple.com/documentation/ /UserExperience/Conceptual/OSXHIGuidelines/index.html

[10] Soft Flow Kft. http://www.softflow.com

[11] Dr. Szirmay-Kalos László, Márton Gábor, Dobos Balázs, Horváth Tamás, Risztics Péter, Kovács Endre: *Theory of Three Dimensional Computer Graphics (PDF version).* http://www.iit.bme.hu/ szirmay/book.html

[12] Dr. Szirmay-Kalos László: *Számítógépes grafika.* Computerbooks, Budapest, 2001

[13] Dr. Szirmay-Kalos László, Antal György, Csonka Ferenc: *Háromdimenziós grafika, animáció és játékfejlesztés.* Computerbooks, Budapest, 2003

[14] E. Angel: *Computer Graphics.* Addison-Wesley Publishing Co., 1990

[15] A. Watt: *3D Computer Graphics.* Addison-Wesley, 1993

[16] Bácsó S., Papp I., Szabó J.: *Projektív geometria.* Mobidiák rendszer, http://mobidiak.inf.unideb.hu, Debrecen, 2004

[17] Hajós György: *Bevezetés a geometriába.* Tankönyvkiadó vállalat, Budapest, 1971

[18] Reiman István: *A geometria és határterületei.* Gondolat, Budapest, 1986

[19] F. P. Preparata, M. I. Shamos: *Computational Geometry.* Springer-Verlag, 1985

[20] Varga Ottó: *Projektív geometria.* Felsőoktatási Jegyzetellátó Vállalat, 1954

[21] I. Herman: *The Use of Projective Geometry in Computer Graphics.* Springer Verlag, Berlin, 1992

[22] J. D. Foley, A. van Dam, S. K. Feiner, J. F. Hughes: *Computer Graphics Principles and Practice.* Addison-Wesley, New York, 1992

[23] D. F. Rogers, J. A. Adams: *Mathematical Elements for Computer Graphics.* McGraw-Hill, 1990

[24] M. O'Rourke: *Principles of Three Dimensional Computer Animation (Modeling, Rendering & Animating with 3D Computer Graphics).* W. W. Norton & Company, New York, 1998

[25] J. D. Foley, A. van Dam, S. K. Feiner, J. F. Hughes and R. L. Phillips: *Introduction to Computer Graphics.* Addison-Wesley Publishing Co., 1994

[26] G. Moore: *Cramming more components onto integrated circuits.* Electronics, 38, 8, 1965

[27] Juhász Imre: *Számítógépi geometria és grafika.* Miskolci egyetemi kiadó, Miskolc, 1995

[28] J. Freeman, D. Skapura: *Neural Networks; Algorithms, Applications and Programming Techniques.* Addison-Wesley, 1991

[29] R. Rojas: *Neural Networks. A Systematic Introduction.* Springer-Verlag, 1996

[30] T. Kohonen: *Self-organization and associative memory. 3rd edition.* Springer-Verlag, 1989

[31] M. Alder, R. Togneri, E. Lai, Y. Attikiouzel: *Kohonen's algorithm for the numerical parametrisation of manifolds.* Pattern Recognition Letters 11, 1990, Pages 313-319

[32] M. Hoffmann: *Modified Kohonen Network for Surface Reconstruction.* Publ. Math., Debrecen, 1998

[33] M. Hoffmann, L. Várady: *Free-form Surfaces for Scattered Data by Neural Networks.* Journal for Geometry and Graphics, 1998

Appendix A

Publications and Teaching Material

A.1 Reviewed Publications

1. CS 0997.68168MA Schwarcz, Tibor; Tornai, Róbert
   Bitmap transformations. (English)
   Kovács, Emöd (ed.) et al., Proceedings of the 4th international conference on applied informatics. Education and other fields of applied informatics, computer graphics, computer statistics and modeling. Eger-Noszvaj, Hungary, August 30-September 3,
   1999. Eger: Molnár és Társa, 169-178 (2001). CR: *G.4 I.3

2. MR2125603 Tornai, Róbert(H-LAJO-II) *Shape modification of cubic B-spline curves by means of knot pairs. (English. English summary)*
   Acta Acad. Paedagog. Agriensis Sect. Mat.
   (N.S.) 31 (2004), 61–68. 65D17 (68U05)

A.2 Publications

1. Schwarcz Tibor, Tornai Róbert:
   Bitmap transformation based on projective invariants
   GEOMETRIA I GRAFIKA INŻYNIERSKA z. 2
   15-23, Gliwice 1998

2. Tornai Róbert:
   Free-form Bitmap Transformations by Neural Networks
   5^{th} International Conference on Applied Informatics '01
   konferenciakiadvány, 229-234, Eger 2001 (under review)

3. Tornai Róbert:
   3D visualization for mobile phones
   II. Magyar Számítógépes Grafika és Geometria Konferencia
   konferenciakiadvány, 129-134, Budapest 2003, ISBN 963 420 766 9

4. Tornai Róbert:
   3D graphics for mobile equipments
   GEOMETRIA I GRAFIKA INŻYNIERSKA z. 6
   53-59, Gliwice 2004

5. Tornai Róbert:
   MMA Technology
   6^{th} International Conference on Applied Informatics '04
   konferenciakiadvány, 449-456, Eger 2004 (under review)

6. É. Széles, B. Kovács, R. Tornai, Z. Győri, H. Berndt:
   HHPN for sample introduction in Flame AAS using subcritical CO_2 as liquid/gas pressure pump
   4^{th} Aegean Analytical Chemistry Days
   29 September - 3 October, Kusadasi/Aydin - Turkey,
   Proceedings book, p.: 584-586
   Abstracts book, p.: 147

A.3 Publications Joint with the Dissertation

1. Schwarcz Tibor, Tornai Róbert:
   Bitmap transformation based on projective invariants
   GEOMETRIA I GRAFIKA INŻYNIERSKA z. 2
   15-23, Gliwice 1998

2. CS 0997.68168MA Schwarcz, Tibor; Tornai, Róbert
   Bitmap transformations. (English)
   Kovács, Emöd (ed.) et al., Proceedings of the 4th international conference on applied informatics. Education and other fields of applied informatics, computer graphics, computer statistics and modeling. Eger-Noszvaj, Hungary, August 30-September 3,
   1999. Eger: Molnár és Társa, 169-178 (2001). CR: *G.4 I.3

3. Tornai Róbert:
   Free-form Bitmap Transformations by Neural Networks
   5^{th} International Conference on Applied Informatics '01
   konferenciakiadvány, 229-234, Eger 2001 (under review)

4. Tornai Róbert:
   MMA Technology
   6^{th} International Conference on Applied Informatics '04
   konferenciakiadvány, 449-456, Eger 2004 (under review)

5. MR2125603 Tornai, Róbert(H-LAJO-II) *Shape modification of cubic B-spline curves by means of knot pairs. (English. English summary)*
   Acta Acad. Paedagog. Agriensis Sect. Mat.
   (N.S.) 31 (2004), 61–68. 65D17 (68U05)

A.4 Teaching Material

1. Tornai Róbert:
   Fejezetek a számítógépi grafikából (lektorált egyetemi segédanyag)
   85 oldal, 2003, Mobidiák rendszer, http://mobidiak.inf.unideb.hu

Appendix B

Glossary

analyte	a kind of reagent with the connecting stain, 4, 13–17, 21, 25, 26, 29, 48
assay	a kind of biological analysis, 3, 5, 6, 8–10, 12, 18, 25, 27, 30, 31, 34, 36, 77, 81
batch	preparation produced together ('sarzs' in hungarian), 9
bead	the shorter form of the microbead term, in fact it designates a little sphere, a 'perl', 3, 4, 6, 7, 13, 14, 17, 21, 25, 29, 32, 48, 77, 81
COI	Cut-Off Index, 21
combo box	a control that combines an edit control with a list box; this allows the user to type in an entry or choose one from the list, 10, 13, 14, 26, 47
cuvet	exposition box, 6
floating toolbar	a toolbar that can appear anywhere on the user's display and is always on top of all other windows; its size or position can be modified when floating, 50

instrument cytometer machine, 7, 10–12, 17, 21, 25, 26, 35, 39, 50

IS Instrument Standard, 17, 29, 30, 32

kit a set of reagents and beads and the AS file together, 5, 8, 9, 12, 15, 17–23, 29, 30, 40, 77, 78, 81, 82

list box a control element that displays a list of character strings; the user selects a string from the list by tapping it with the stylus; when a string is selected, it appears highlighted; you can use a vertical or horizontal scroll bar with a list box to scroll lists that are too long for the control window; the list box automatically hides or shows the scroll bar, as needed, 15

LOT serial number of the released kit, 9, 23

MBCS MultiByte Character Sets - MFC specifically supports double-byte character sets (DBCS), 50, 51

MFC Microsoft Foundation Classes: class hierarchy to support quick visual program developing, 44, 50, 52

MMA Multiplex Microbead Assay, or it is also referred to SAT (Suspension Array Technology) or CBA (Cytometric Bead Array) technology, 1, 3, 77, 81

NC Negative Control, 21, 29, 34

PC Positive Control, 21, 29, 34

pixel atomic building unit of a screen, 55, 57, 69, 70, 75, 79, 83

reactive reagent, 3, 4, 30, 34, 77

RFI Relative Fluorescent Intensity, 17, 29

SAT Suspension Array Technology, 5, 10, 16, 30, 36, 40, 46–49, 77, 81, 82

SSC Side SCatter, 10, 26

stain special color, 3, 4, 6, 29, 32, 77

texel atomic building unit of a bitmap or texture, 57, 78, 82

UNICODE Unicode is a worldwide character-encoding standard that uses 16-bit character values to represent all the characters used in modern computing, including technical symbols and special characters used in publishing, 47, 50

MMA Software Developing and Bitmap Transformations

Értekezés a doktori (Ph.D.) fokozat megszerzése érdekében
a Matematika és Számítástudományok tudományágban

Írta: Tornai Róbert
okleveles programtervező matematikus – angol-magyar szakfordító

Készült a Debreceni Egyetem Matematika és Számítástudományok
Doktori Iskolája (Informatika programja) keretében.

Témavezető: Dr. Szabó József

A doktori szigorlati bizottság:

elnök:	Dr. Kormos János	....
tartalék elnök:	Dr. Fazekas Attila	....
tagok:	Dr. Kovács Emőd	....
	Dr. Kuki Attila	....
tartalék tagok:	Dr. Juhász Imre	....
	Dr. Bognár Katalin	....

A doktori szigorlat időpontja: 2005. ....

Az értekezés bírálói:

	Dr. Fazekas Attila	....
	Dr. Lustyik György	....
tartalék bíráló:	Dr. Hajdu András	....

A bírálóbizottság:

elnök:	Dr. Dömösi Pál	....
tartalék elnök:	Dr. Papp Gyula	....
tagok:	Dr. Arató Mátyás	....
	Dr. Fazekas Gábor	....
	Dr. Szirmay-Kalos László	....
	Dr. Juhász Imre	....
tartalék tagok:	Dr. Kozma Laszló	....
	Dr. Kovács Emőd	....

Az értekezés védésének időpontja: 2005. ....