Skin Cancer Detection System
Registration of moles in skin images

Name author: ing. M. Rieck

Date: June 1997

Supervisors: prof. dr. ir. W.M.G. van Bokhoven
SES
Department of Electrical Engineering
Eindhoven University of Technology

dr. tech. associate prof. Juha Röning
Computer Vision Group
Department of Electrical Engineering
University of Oulu

Erasmus coordinator: dr. ir. L.K.J. Vandamme
Eindhoven University of Technology
This thesis is the start of the development of a skin cancer detection system, the main focus of the study is the automatic registration of moles in successive images of skin. The project was carried out in the Computer Vision Group of the Computer Laboratory at the University of Oulu (Finland) during the period from October 1996 until June 1997.

This project to obtain a Masters degree in Electrical Engineering of the Eindhoven University of Technology (The Netherlands) was realized in framework of an ERASMUS exchange program between the Eindhoven University of Technology and the University of Oulu.

The outline of this thesis is as follows. Chapter 1 discusses the importance of digital imaging and computer vision in the field of dermatology, points out the aims of this study and gives a brief overview of this report. The overview of the Skin Cancer Detection System (SCDS) that is developed in this study is presented in chapter 2. Before the successive images of skin can be registered the location of the lesions or moles in these images must be found, chapter 3 describes how this is accomplished. Chapter 4 and 5 form the heart of this study namely the registration of the images. Here different algorithms are introduced and tested which correctly label the moles in the successive image that represent the same mole. Chapter 6 introduces different features of the moles that can be measured to check if corresponding moles in successive images have changed. The actual comparison of these features of corresponding moles is dealt with in chapter 7. This chapter also treats the part of the system that indicates to the physician at which moles, suspected of skin cancer, he or she must have a closer look. The whole system in implemented in the Khoros Scientific Software Development Environment, chapter 8 explains how this is done. And finally chapter 9 concludes this thesis with conclusions and suggestions for further research.

Oulu, June 1997
Marcel Rieck
Abstract

The only cure for malignant melanoma (skin cancer) is early detection. Surgical removal of a still thin melanoma will result in a complete cure. In this study the first steps are made in the development of a skin cancer detection system. The Skin Cancer Detection System is a computer vision tool for physicians that automatically screens images of skin for changes that are suggestive for melanoma. The study features two main subjects: automatic registration of skin lesions (or moles) in successive skin images and the detection and measurement of changes within moles.

The automatic registration can be defined as correctly labelling of lesions that represent the same mole in successive images and identifying lesions that have no corresponding lesion in the other image. The last ones are new lesions and could be skin cancer. Two moles that are labelled as the same mole are called a mole pair. For this registration a number of algorithms were implemented and tested, the best one of these finds the mole pairs correctly in 99% of the cases and needs two initial mole pairs and the rest of the mole pairs are found automatically. For the selection of these initial mole pairs a algorithm was implemented that finds in more than 99.2% of the cases three correct initial mole pairs. When these two algorithms are combined to form the total registration process, 98-99% of the cases the correct mole pairs are found.

After registration the moles in the successive skin images that represent the same mole are checked for changes. For this purpose features of the moles are calculated that describe some characteristic of the mole in question. The features used here are specially tailored to recognize malignant moles. The moles of which these calculated features have changed over time are suggestive of melanoma.

The moles that are identified, by the registration process, as new moles and the mole that are identified as changed after the comparison of mole features are indicated to a physician for further investigation.

All algorithms that make up the Skin Cancer Detection System are implemented in the Khoros Scientific Software Development Environment.
1 Introduction .......................................................... 1
  1.1 Why machine vision in dermatology? .......................... 2
  1.2 Aim of this work ................................................... 2
  1.3 Overview of this work .............................................. 3

2 Overview of the system ............................................ 5
  2.1 The Skin Cancer Detection System (SCDS) ......................... 6
  2.2 The individual parts of the system ............................... 6
    2.2.1 Segmentation .................................................. 7
    2.2.2 Registration ................................................... 8
    2.2.3 Comparison .................................................... 8
    2.2.4 Visualization .................................................. 8

3 Images and Segmentation ............................................ 9
  3.1 Images of the patient ............................................. 10
    3.1.1 Acquiring images .............................................. 10
    3.1.2 Standardizing images ......................................... 10
  3.2 Detecting the moles ............................................... 11
    3.2.1 Image segmentation .......................................... 11
      3.2.1.1 Region growing .......................................... 12
      3.2.1.2 Clustering techniques .................................... 13
      3.2.1.3 Edge detection ........................................... 13
    3.2.2 Segmentation of mole images .................................. 13
  3.3 Indicating moles and markers manually ........................... 15
    3.3.1 Marking the moles ............................................. 15
    3.3.2 Marking the corners of the marker ......................... 16
    3.3.3 Choosing colours for marking ................................ 17
  3.4 Finding all mole pixels ......................................... 17

4 Registration .......................................................... 19
  4.1 Registration of the images ....................................... 20
  4.2 Algorithms of Perednia and White ............................... 20
    4.2.1 Matching algorithm: 1 initial match point .................. 21
    4.2.2 Matching algorithm: 2 initial match points ................ 22
    4.2.3 Matching algorithm: 3 initial match points ................ 23
    4.2.4 Comparison of the algorithms ............................... 24
    4.2.5 Testing the 3 point algorithm on SCDS images .......... 25
  4.3 The baseline algorithm ......................................... 27
    4.3.1 Description of the algorithm ............................... 27
8.4 The complete SCDS in Khoros ........................................ 66
8.5 Summary ........................................................................ 66

9 Conclusions ................................................................. 67

References ........................................................................ 71

A1 Principle Axes .......................................................... 75
  1.1 Physical description of the principle axes of inertia ........ 76
  1.2 Converting formulas to the binary mole image .......... 78

A2 The Khoros system ....................................................... 81
  2.1 What is the Khoros system? ........................................... 82
  2.2 What does Khoros consist of? ...................................... 82
    2.2.1 Programming services ......................................... 82
    2.2.2 Toolboxes ......................................................... 82
    2.2.3 Programming tools ............................................. 83
  2.3 Cantata ................................................................. 83
    2.3.1 Out of what does Cantata consist? ....................... 83
    2.3.2 Why use Cantata? ............................................ 83
  2.4 More information and licensing ................................. 84

A3 File formats ............................................................... 85
  3.1 Format of Images to Extract File .................................. 86
  3.2 Format of the Mole Pattern File ................................. 86
  3.3 Format of the Unmapped File ..................................... 87
  3.4 Format of the Mole Pair File ...................................... 87
  3.5 Format of the Graph File ......................................... 88
  3.6 Format of the Evaluation File .................................... 89
  3.7 Format of the Statistics File ...................................... 89
  3.8 Format of the Reference and Match Feature File ......... 91
  3.9 Format of the Mole Difference File ............................ 91

A4 Man pages .................................................................. 93
  4.1 Man-page of the program Extract ............................. 94
  4.2 Man-page of the program DetectFeatures ................... 96
  4.3 Man-page of the program InitPntArt ........................... 98
  4.4 Man-page of the program InitPntOwn ........................ 100
  4.5 Man-page of the program PntRegistrationArt1 ........... 101
  4.6 Man-page of the program PntRegistrationArt2 .......... 103
  4.7 Man-page of the program PntRegistrationArt3 .......... 104
  4.8 Man-page of the program PntRegistrationOwn .......... 105
  4.9 Man-page of the program Compare ........................... 107
  4.10 Man-page of the program Visualize ......................... 109
  4.11 Man-page of the program Randomize ....................... 111
  4.12 Man-page of the program Check .............................. 113
  4.13 Man-page of the program Stat ................................. 114
Chapter 1  Introduction

This Chapter deals about the importance of digital imaging and computer vision systems in the field of dermatology, special attention is paid to computer vision as a tool for detecting skin cancer. The aims of this work are pointed out and an overview of it is given.
1.1 Why machine vision in dermatology?

The research work in computerized biomedical image processing and analysis has made revolutionary advances since its experimental stages in the early 1970's [1]. Since then a large number of computer vision systems have been developed to aid in diagnosis, analysis and storage of medical images. A large number of the medical imaging systems have been developed to visualize parts of the body that can not be seen from the outside. X-ray scans, magnetic resonance imaging (MRI), positron emission tomography (PET), X-ray computer tomography (CT) and ultra sound provide information such as tissue characteristics, functional information of organs and bone structures by means of non-invasive action.

However dermatologists study an organ that is entirely visible, so why should they need imaging techniques for skin? This has been largely the opinion in the field of dermatology before 1985. Before this time reports on digital imaging in dermatology were rare [2]. In 1985 a widespread publicity campaign, initiated by the American Academy of Dermatology, was at its peak. The publicity effort regarding skin cancer include new reports, popular articles and increased public skin cancer screenings. In this same year, an issue of Ca - A Journal for Clinicians publicized the “ABCD's” (4 characteristics for separating malignant melanoma from benign moles) of malignant melanoma [3]. This article and a previous article in 1982 had noted the sharp rise in malignant melanoma, over 15 times the number of cases in the 1930's, which was the sharpest increase for any cancer except for lung cancer in women.

The only cure for malignant melanoma is early detection. Surgical removal of thin melanoma results in complete cure. Initially the tumour spreads radially in the superficial skin layers, but soon vertical infiltration occurs. The thicker the lesion is in depth, the more likely it is that tiny metastasis have already entered the blood, resulting in a more sinister diagnosis.

The increased public awareness of the disease and the importance of early detection has lead more and more people to seek professional advice on pigmented lesions. Biopsies became the most frequent reimbursed medical procedure in the USA [2], exceeding one billion dollars annually. It was hoped that image analysis through computer vision could help follow patients at risk for malignant melanoma without increasing the number of skin biopsies. Computer vision might also help increase the accuracy of the diagnose, because reports assessing diagnostic accuracy prior to excision have shown that dermatologists were able to detect melanoma in only about 75-80% of the cases [4]. And clinical diagnosis may be particularly difficult in developing lesions.

Even if vision systems are not able to provide a fully automatic diagnosis, it might still be a very valuable tool for the dermatologist since vision systems provide an objective and quantifiable judgement. It is also possible to add non-visual properties to the image such as skin thickness or tumour depth by making non-"visible light" images, e.g. with methods as ultra sound or MRI [2]. Also other fields of dermatology, such as the treatment of the skin disease psoriasis, have discovered how valuable digital imaging can be.

1.2 Aim of this work

Early melanoma are typically recognized when a new lesion appears or changes are noted in a preexisting pigmented mole. Currently a physician locates these changes by comparing the part of the body of the patient being screened to a picture that taken during a previous visit of the patient. Unfortunately a number of
features may complicate the process of finding malignant melanoma. E.g. people with large numbers of moles are at the highest risk for melanoma, but they are also the most difficult to screen.

The aim of this study is to use computer vision to screen pairs of skin images, which are taken at different times, for detecting changes that are suggestive of melanoma. The study will consist of two main parts: automatic registration of skin lesions and the detection and measurement of changes within lesions.

**Automatic registration:**
This can be defined as the correct labelling of lesions that represent the same lesion in successive images of skin. Lesions that have no corresponding lesion in the previous image must be brought to the attention of the attending physician.

**Detection and measurement of changes within lesions:**
Lesions that have a corresponding lesion in the previous image must be checked for changes. Different features of the lesions will be measured to detect if lesions have changed over time. Lesions that have changed must also be brought to the attention of the physician.

The whole vision system will be implemented in the Khoros. Khoros is a Scientific Software Development Environment, which supports a (visual) interface for end-users and powerful tools for fast prototype programming.

**1.3 Overview of this work**

Chapter 2 gives an overview of the Skin Cancer Detection System (SCDS) that has been developed at the University of Oulu and describes shortly what the function of every part in the system is and globally what it does. The first stage of the SCDS, namely taking the pictures of the patient and locating the moles in these images, is reviewed in Chapter 3. Chapters 4 and 5 form the main part of this study: the automatic registration of the moles in the images. Some algorithms found in literature are reviewed and new algorithms are introduced and evaluated. Measurements of mole features to detect changes in moles over a certain period of time are given and implemented in Chapter 6. Chapter 7 deals with detecting changes in moles and visualizing the changed and the newly appeared moles. The whole system SCDS is implemented in Khoros, Chapter 8 explains how this is done. And finally Chapter 9 concludes the report with conclusions and recommendations for further study.
Chapter 2  Overview of the system

This chapter gives a general overview of the vision system that is developed in this report. Definitions which will be used throughout this report are given. Directions are given as to how images should be taken of the patient such that the system is able to process them. Some attention is given to the segmentation of the images.
2.1 The Skin Cancer Detection System (SCDS)

The main principle of Skin Cancer Detection System (SCDS) is drawn in Figure 2.1. The SCDS has two images of the same part of the body of a patient as input. For every part of the body being checked there are two of such images. They are taken during two successive photo sessions. The images of the first photo session are called reference images. The ones of the second session, which are taken some period of time later, are called match or follow up images.

The SCDS compares the two images and searches for changed and newly appeared moles. First it Figures out where the moles are in both images by a process called segmentation, which produces positional and feature (such area, perimeter, colour etc.) information for all the moles in both images. After this the images are registered or matched, such that moles that represent the same mole in the two images are labelled as the same mole. Moles with the same label form a so-called mole pair. Moles that were not paired with another mole are new moles and are possible skin cancer candidates. Finally the moles that do from a pair with another mole are compared on their features to see if they have changed over a the period of time between the two photo sessions. The moles that have changed are also possible skin cancer candidates.

The possible skin cancer candidates have to be inspected closer by a physician to make a final diagnosis. For this purpose they are indicated to the physician on the attention image, which is the output of the system. The attention image is a compilation of the input images in which the moles that are new or have changed are indicated by a certain marker.

Figure 2.1 Main principle of the Skin Cancer Detection System. The reference and match images are compared and the changes are noted in the attention image.

2.2 The individual parts of the system

The whole SCDS can be divided in a number of processes, each with a different function. This is done in Figure 2.2. The processes are indicated by dashed boxes, the processes themselves are again subdivided into smaller parts. These parts are the programs in Khoros in which the whole system is implemented. The arrows between the programs indicate the information flow. Information exchange between programs is in the form of files. In the following paragraphs the function of various processes and programs that make up the SCDS will be briefly described.
2.2.1 Segmentation

The first stage of processing, as in most computer vision systems, is the segmentation of the images. Segmentation divides the images into regions that correspond to objects of interest. In this case the segmentation algorithm has the task to identify which pixels in the reference and match image represent the individual moles. After this the positional information of the moles is known and written to the Mole Pattern File. To be able to compare changes in the moles, features are calculated. The calculated features are written to the Reference Feature File and the Match Feature File for respectively moles of the reference and the match image.

![Diagram of the individual parts of the Skin Cancer Detection System](image)

Figure 2.2 The individual parts of the Skin Cancer Detection System. The dashed boxes are the processes that take place in the SCDS. The processes are implemented by programs in the Khoros programming environment, depicted in the solid boxes. Communication between the programs is done using files.

At the moment of this study there is no reliable segmentation algorithm available. The problem of finding the moles in the images is “solved” by marking the positions manually in the “marked reference image” and the “marked match image”. These images are the same as the original reference and match image with the only difference that the mole positions are marked. Chapter 3 describes the way this marking should be done in order that the program Extract can read those files and produce the Mole Pattern File from them.

The information in the Mole Pattern File is used by the program DetectFeatures, which calculates the mole features from the original reference and match images. The algorithms for calculating these features are discussed in Chapter 6.
2.2.2 Registration

The registration process is the main focus of the study. It has the task of determining which moles in the reference and match image represent the same mole. The mole pairs it finds are written to the Found Mole Pair File for further processing. The moles of which no corresponding mole in the other image is found are written to the Unmapped File, to be displayed as possible skin cancer candidates.

For reasons explained in Chapter 4, the registration process is divided in two parts, here represented as the InitialPair and Registrate program. The first one, of which the algorithms are featured in Chapter 5, selects a few highly likely initial matches or mole pairs which the Registrate program, see Chapter 4, uses to find the other remaining mole pairs.

2.2.3 Comparison

The moles, of the mole pairs found by the registration process, have to be compared to see if they have changed, i.e. might have become cancerous. This is done by the comparison process which is implemented by the Khoros program Compare. This program recalls the features of the paired reference and match mole in the reference and match features file and checks if the features are different. If the features in the successive reference and match image have changed more than a certain threshold then the mole is labelled as “changed”. The results are written to the Mole Difference File. A complete description of this process can be found in Chapter 7.

2.2.4 Visualization

The final stage of the SCDS is visualizing which moles might have become cancerous so that the attending physician can subject them to a closer inspection. The visualization process is performed by the program Visualize, which is treated in Chapter 7.
Chapter 3  Images and Segmentation

This chapter describes how the images of the patient are taken in order that the SCDS will be able to process them. It also explains how the moles are located in the images in the absence of a segmentation process.
3.1 Images of the patient

To diagnose the patient using the SCDS, a number of pictures are taken of the patient. This paragraph describes how and of which body parts these images should be taken in order that they can be processed by the SCDS and what materials are used to convert the images to a digital form.

3.1.1 Acquiring images

To be able to check the whole body of a patient for skin cancer, the whole body should be photographed. The total body surface can be captured on film by taking eight pictures, namely:
- front and back of the upper trunk (2 pictures);
- front and back of the left and right arm (4 pictures);
- front and back of the legs (2 pictures).

The SCDS checks for changes in the moles and for new moles over a period of time, this means that there must be two photo sessions with a certain period of time in between. This implies that there are sixteen pictures for every patient (eight per session) that will serve as input to the SCDS. The pictures are taken using a normal 36x24mm² photo camera with a 35 mm objective. After development the dia slides are digitized using a colour scanner that has a resolution of 2000 dpi and codes every primary colour in 8 bits. The primary colours are red (R), green (G) and blue (B). Examples of these images from one photo session are depicted in Figure 3.1 and 3.2.

![Figure 3.1 Examples of pictures of the upper trunk and the legs.](image)

a. Upper trunk front  b. Upper trunk back  c. Legs front  d. Legs back

3.1.2 Standardizing images

To make the SCDS less complex the images are taken in a certain standard fashion. The patient is photographed against a blue background to simplify the separation of body pixels and background pixels. A red marker is placed as a scale reference on each body part. This marker is exactly 120 mm long and 25 mm wide, the colour makes the marker pixels easy to separate from the body pixels. The markers can be used to measure the size of moles and in the SCDS they are used to remove scale differences between the pictures in order to compare the moles on features such as area and diameter.

The body parts in the pictures of the first session and the pictures of the second
Detecting the moles

session have about the same orientation and scale. This means e.g. that if the first session picture of an arm shows a horizontal arm with the hand on the left side, then the second session picture of this arm shows the same kind of configuration (horizontal arm, hand on the left side). Special attention is paid to the way the upper trunk is photographed. If the arms in the first session picture and the follow up picture are not in the same position then the amount of elastic distortion in the position of the moles between the two images is considerably large. This distortion makes it much more difficult for the registration process to deduct which moles represent the same mole in both images.

![Examples of pictures of the arms.](image)

The easiest way to reproduce the same position of the arms when taking pictures of the upper trunk is to keep them horizontal. The scale of the first session and follow up images are kept about the same by taking the photos from about the same distance.

### 3.2 Detecting the moles

The first step after image digitalization finds out which pixels in the image represent body pixels, mole pixels, marker pixels and background pixels. This process is called image segmentation. The segmentation of the images is not part of this study, but just to provide a short general overview of segmentation techniques some techniques will be discussed in the following paragraph. Paragraph 3.3 will introduce the manual method we used to indicate to the SCDS where the moles and markers are located in the images.

#### 3.2.1 Image segmentation

This paragraph gives a short general overview of image segmentation, it is a summary of an article of Umbaugh et al. [5]. Image segmentation is important in many computer vision and image processing applications. Partitioning of the image into regions corresponding to objects of interest is necessary before any processing can be done at a level higher than that of the pixel. Identification of real objects, pseudo objects, shadows or actually finding anything of interest within the image requires some form of segmentation. Conceptually, image segmentation methods will look for objects that either have some measure of:

- homogeneity within themselves or
- have some measure of contrast with the objects on the border.
Most image segmentation algorithms that are used are modifications, extensions or combinations of these two basic concepts.
The following review divides image segmentation techniques into three main groups:
- region growing;
- clustering methods;
- edge detection.
In Figure 3.3 a graphical representation of these three major image segmentation groups is given.

**Figure 3.3** Image segmentation techniques are divided into three major categories. a) Region growing is performed within the image by finding homogeneous regions and growing them until they no longer meet the homogeneity criteria. b) Clustering techniques look for groups, or clusters, in domains other than the spatial domain of the image. Here clusters are found in a 3-D space, e.g. RGB space. c) Edge detection methods look for edges, boundaries, or lines usually via a difference operator.

### 3.2.1.1 Region growing
Region growing refers to a class of image segmentation methods where the goal is to find regions that represent objects or meaningful parts of objects. The method is based primarily on spatial considerations. Some of the techniques used are local, in which small areas of the image are processed. Others are global, where the entire image is considered during processing. There are also methods that combine local and global techniques such as split and merge techniques.
In general, the split and merge technique proceeds as follows. First, the image is split into equal size regions, then some type of statistical method is applied to determine a measure of similarity within each of the regions. This measure may include texture variation, colour variation, intensity variation, or other features of interest. Once this information has been calculated for each of the regions, the image is ready for the next step in processing. The next step will normally be a homogeneity test for each of the regions. The regions that are acceptable by the criteria of the homogeneity test will be left alone. The regions that do not pass the
test will be split into more subregions. After the split is made, a merge is attempted. This merge procedure will attempt to merge each region with its neighbouring regions, or sub-regions, and if the resulting region is acceptable to the homogeneity test, these regions will be merged. This procedure stops when all regions that have been formed pass the homogeneity test.

### 3.2.1.2 Clustering techniques

Clustering techniques are image segmentation methods whereby individual elements are placed into groups. These groups are based on some measure of similarity within the group. The major difference between these techniques and region growing techniques is that domains other than spatial may be considered as the primary space being used for the segmentation.

One method of image segmentation, based on clustering, which is in widespread use, is the method of taking the space of interest and splitting the space into regions by setting limits on each of the dimensions for each separate region. In the case of using an RGB colour space, this would mean taking the three-dimensional colour space and dividing it into rectangular parallelepipeds with edges parallel to the axes in the colour space.

Recursive region splitting is a clustering method that has become a standard technique. This method uses a histogram thresholding technique to segment the image. A set of histograms is calculated for a specific set of features, then each of these histograms is searched for distinct peaks. The best peak is selected and the image is split into regions based on this thresholding of the histogram.

Many methods may actually be a combination of region growing methods and clustering methods. The segmentation method of dividing the image based on clusters in colour space can be implemented strictly as a clustering method or spatial considerations may also be included. Optimal image segmentation will most likely be a combined approach.

### 3.2.1.3 Edge detection

Edge detection, as a method of image segmentation, is performed by finding the boundaries between objects, thus defining the object themselves, indirectly. This method is usually implemented by first marking points that may be part of an edge. These points are merged into line segments, and the line segments are then merged into object boundaries.

The most common method of finding edges in a digitized image is to apply a spatial differentiation operation to small blocks of pixels, local neighbourhoods, within the digitized image. Places in the image where the first order differentiation returns a large number, mark points of rapid change, thus indicating the possibility of an edge. These edge points represent local discontinuities in a specific feature, such as brightness, colour, or texture. Many edge detection operators have been defined, but most are based on these fundamental concepts.

### 3.2.2 Segmentation of mole images

Accurate and reliable outline detection is important in the automated diagnosis of moles in order to segment the image into mole and background skin, thereby ensuring that all kinds of measurements are carries out only on the mole pixels. In addition, given an accurate outline, important diagnostic features of the mole shape can be measured to provide a quantitative measure of size, asymmetry, and border irregularity [6], [7], [8], [9]. The characteristics of skin images are very
variable (e.g. lighting, mole size, nature of the mole, skin texture, hairs, pores, background objects such as rulers) and these produce problems in obtaining methods which are reliable, repeatable and robust. A number of methods have been proposed to solve the mole boundary detection problem. However none of the methods have proved sufficiently reliable on a wide range of images.

It has been found that the general methods and algorithms used for edge detection in digital images are not well-suited for the detection of the whole object boundaries. In particular it has been found that classical operators such as the Sobel operator do usually not find the edge of a tumour. Instead, they often identify elements of skin texture such as pores and hair [13].

Laplacian-of-a-Gaussian (LoG) edge detection with subpixel interpolation, has been used by Perednia et al. [10] for boundary detection. This type of border detection with a fixed size operator is found to be very unreliable because of the nature of the images. Small scale edge detectors provide large numbers of unwanted edges caused by skin texture and hairs. While large scale detectors have the same effect as low pass filtering, the boundary is less effected by unwanted edge features, but the locational accuracy is reduced as result of an inadequate resolution. Denton et al. [11] developed a variation on the LoG edge detection which they call edge focusing. In this method the size of the detector is adapted to the scale of the image.

Another method that is commonly used to identify an object in an image is adaptive thresholding. Although this adaptive thresholding can be highly successful in certain areas, features such as variegated colouring (the mixed shades of tan, brown and black that are usually associated with malignant melanoma) make this an unreliable method for differentiating moles from normal skin.

Thresholding, preceded by filtering and a linear colour transformation, followed by region growing and contour tracing and smoothing has been used by Ercal et al. [12]. In this method, the colour transform and the threshold level is based on the colour of two windows which have been identified to be inside and outside the mole. These windows are found by histogramming and approximate colour segmentation. It can be used for some classes of moles, but not all.

Golston et al. [13] developed the radial search algorithm which starts at the centre of the mole and searches outward along radial lines for a sustained change in image brightness. This method can be inaccurate if the transition between mole and background skin is not reasonably sharp (the blurry mole border is an indication of a possible malignant melanoma [14]). It also assumes that no radius intersects the border more than once (radial connectedness).

Umbaugh et al. [15], [16], [5], [17], use colour segmentation using a Principle Components Transform (PCT) followed by a median split algorithm after which only large colour object are retained. An adapted radial search algorithm is then used to detect the boundary of the mole. They report that this method detects the mole border correctly in about 66% of all mole images they tested, which is hardly reliable enough for a fully automatic system.

Most methods mentioned above are not only unreliable over a wide range of images but are also developed for images that contain only one mole with the goal of finding the mole border and diagnosing the mole on the basis of certain features that can be calculated. The images of the SCDS contain more moles which have to be segmented using other techniques. From the amount of research that has been done on the segmentation of mole images, one can conclude that the segmentation of mole images is a complicated task for vision systems.
3.3 Indicating moles and markers manually

As mentioned before, this study does not include the segmentation of the images but is mostly focused on the registration of the images and to a lesser extend in determining mole features to compare individual moles. That is why another method is used to tell the SCDS where the moles are. For every image there is a marked image. This image differs only from the original image in the fact that all mole centres and corners of the marker are marked by changing the colour value of the pixel to a value that does not appear in the original image. In this way it is an easy task for the program Extract, see Figure 2.2, to extract the mole and the marker positions in each image. It just walks through the image pixel by pixel, starting in the upper left corner of the image and then moving from left to right and from top to bottom. Of all the mole centres that are found, the position and the mole label (moles are labelled with a number, see paragraph 3.3.1) are written to the so-called Mole Pattern File. The same is done for the markers, the positions of the corners are written to the same file. By giving corresponding moles in the reference and the match image the same labels, the Extract program is able to Figure out which moles form pairs. These correct mole pairs are written to a file which is called the Correct Mole Pair File. This file can be used to test if the registration process works correctly, i.e. if it finds the mole pairs correctly on basis of the mole positions information. The following paragraphs describe how moles and markers are marked in such a way that the marked images can be used for the Extract program.

3.3.1 Marking the moles

Using a drawing program all the moles in the marked images are indicated by a pixel of a certain colour at the centre. From this the Extract program derives the position of the moles. Furthermore, every mole is given a label in the form of a 7 bits number. The labels are formed by 7 pixels immediately next to the marked moles centre in the marked image, see Figure 3.4. Each pixel in this label

![Figure 3.4](image-url)

Figure 3.4 The mole centre is marked to indicate its position and a label is given, representing the mole number. One of the possibilities of labelling the mole must be chosen.
Images and Segmentation

represents one bit of the label and has one of two colours, representing either an one or a zero. In this way it is possible to mark up to 128 (0..127) moles per image. The label can have one of four orientation: horizontal to the left of the marked mole centre pixel, horizontal to right of the mole centre or vertical above or below the mole centre pixel, see Figure 3.4. When assigning a label to a mole one of these orientations must be chosen. The most significant bit of the label (b6) is always represented by the pixel that is closest to the marked mole centre.

The moles in the reference and match image are labelled in such a way that moles that represent the same mole are assigned the same label. The Extract program assumes that the moles in the marked reference and match image that have the same label for a mole pair. These correct mole pairs are written to the Correct Mole Pair File and is used later to measure the performance of the registration algorithms by counting the number of these correct mole pairs that these algorithms find correctly, incorrectly or do not find at all.

In a real set up where the moles are found by means of segmentation it is of course not possible to label the moles in this way (otherwise there would be no need for a registration process) but for testing it is very convenient.

### 3.3.2 Marking the corners of the marker

Also the red markers in the images are marked to be able to find them without image segmentation, though finding the marker by means of segmentation is not that difficult because of the colour the marker has.

![Marker colour 1](image1)

![Marker colour 2](image2)

*Figure 3.5 The red marker on the images is marked at the corners. One long side is marked with “Marker colour 1”, the other long side is marked using “Marker colour 2”.*

The characteristic points of a marker are of course its corners and these are marked by changing the colour of the corner pixel. The most important feature of a marker is its length and to be able to calculate this one must know which of the corners belong to one of the long sides of the marker and which of the corners belong to the other long side. To make this a bit easier the corners of one long side are marked with a certain colour and the corners of the other long side are marked with another colour. Figure 3.5 depicts a marker of which its corners are marked.
3.3.3 Choosing colours for marking

The colour of the pixels in the images consist of three components, namely Red (R), Green (G) and Blue (B). Every component is coded in 8 bits, which can take the values 0..255. The colours used to mark the images must be chosen in such a way that they do not appear as colour in the original image, otherwise the SCDS will process false mole and corner positions. The colours that are quite save to choose are those for which one or more of the RGB-components have the value 255 while the rest of the components are 0. This gives seven different colours for which it is very unlikely that they appear in a digitized image. Of these seven colours we only need five, since there are only five different marks (mole centre, mole label one, mole label zero, corners marker one long side and corners marker other long side). It is completely irrelevant which colour is chosen for the different marks, for this study the colours are chosen as mentioned in Table 3.1.

<table>
<thead>
<tr>
<th>Mark type</th>
<th>Red</th>
<th>Green</th>
<th>Blue</th>
<th>Colour name</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mole centre</td>
<td>0</td>
<td>255</td>
<td>0</td>
<td>Green</td>
</tr>
<tr>
<td>Mole label: zero</td>
<td>0</td>
<td>0</td>
<td>255</td>
<td>Blue</td>
</tr>
<tr>
<td>Mole label: one</td>
<td>255</td>
<td>0</td>
<td>0</td>
<td>Red</td>
</tr>
<tr>
<td>Corners long side marker 1</td>
<td>255</td>
<td>255</td>
<td>0</td>
<td>Yellow</td>
</tr>
<tr>
<td>Corners long side marker 2</td>
<td>0</td>
<td>255</td>
<td>255</td>
<td>Cyan</td>
</tr>
</tbody>
</table>

*Table 3.1* Colours used to mark the images.

3.4 Finding all mole pixels

Now the mole positions are known, but in order to calculate mole features, such as area, perimeter etc., it must be known which pixels belong to certain a mole. Since there is no segmentation process in the current SCDS, a radial search algorithm [13] is used to find the border of the mole. All the pixels that lie within the mole border are defined as the mole pixels of that specific mole. This algorithm takes advantage of the fact that the mole centre is known. Starting from this point it searches along a large number of radial lines for a significant sustained intensity change in the image. This intensity change is most likely the border of the mole. The radial search algorithm is used as a preprocessing stage in the DetectFeatures program, which calculates the features of the moles. More details on the implementation of the radial search algorithm can be found in Chapter 6.
Chapter 4  Registration

This chapter focuses on the registration of the moles, i.e. determining which moles in the successive images represent the same mole. Three algorithm found in literature are discussed and a new, better performing, algorithm is introduced and tested.
4.1 Registration of the images

After segmentation of the reference and match images of the patient the moles in both images can be regarded as points with unique positional information and a vector of mole feature values. The next major step is to determine which moles in the reference and match image represent the same mole. This is important in order to be able to compare the feature values of the moles to check for changes. Furthermore, if the match image contains a new mole then this will be found as well, as it does not have a corresponding mole in the reference image. In both cases, a changed mole or a new mole, we are dealing with suggestive melanoma, which must be brought to the attention of a physician. The process of finding similar objects in successive mole images is called matching or image registration.

The matching can be based on the spatial relationship between the objects to be matched, or it can be based on a similarity metric for comparing the feature values between object. Combinations of these two are also possible. Methods based on a similarity metric for comparing feature values have some drawbacks. The mole images contain many similar moles and an actual change occurring in a mole could give rise to mismatches. These techniques also use similarity metrics that must be weighted appropriately for each feature value being used, which can be quite a difficult task [18]. Spatial techniques, also called point matching techniques, do not have these problems and tend to be quite robust [18]. There are a number of techniques for point matching problems. Relaxation is one of these techniques, it has been created to satisfy a broad set of matching problems, but they can be quite slow and because of the iterative nature there is no fixed running time. Another aspect is that relaxation matching often relies on a set of thresholds to determine whether two points are to be paired as match or not. Establishing appropriate levels for these thresholds can be quite troublesome.

Correlation techniques represent an alternative to relaxation methods. While these are often time consuming, it is possible to accelerate the matching process by adding special constraints into the matching process. The point matching techniques of Perednia and White [18] use these kind of constraints to reduce the number of dimensions of the problem. Their algorithms rely on the presents of a number of initially confirmed matches between points of both images. In principle these initial matches can be provided by the attending physician or by means of an algorithm that finds the most likely initial match points. Algorithms that do this are described in Chapter 5. In this chapter first the registration algorithms of Perednia and White are treated, implemented and tested. After this a new algorithm, the baseline algorithm, is introduced.

4.2 Algorithms of Perednia and White

In [18] Perednia and White introduce three algorithms for point pattern matching. All these algorithms use initial match points to reduce the dimensionality of the matching problem by correlation. The reduction in dimensionality speeds up the process enormously. In their article they make some assumptions about the images which are being point matched:

- Images can be treated as a spatial pattern of points in a 2-dimensional plane (influences of the curvature of the body are neglected, or must be removed by a preprocessing stage).
- The images are taken at the same camera-patient distance, in which case scale differences should not be a major problem (in our case scale differences
are removed by scaling the images in such a way that the red markers have the same size).

- Most of the remaining differences between the images will be in the form of rotation and translation, caused by differences in the camera-patient position between both images.

- A small amount of stretching (elastic deformation) may be present in the images, caused by little differences in posture of the patient during the two photo session. The assumption is that this deformation is locally highly correlated and the overall movement of points between the images can be closely approximated by a rigid-body motion of the point pattern in a 2-dimensional plane.

- Initial match points may be selected by a user or may be provided in a preprocessing stage.

All three algorithms introduced in [18] are described in the next paragraphs. The first one takes one initial match point to reduce the problem from a 2-dimensional one to a 1-dimensional one. The correlation problem can be eliminated completely if two or three initial matches are provided, since in that case a set of points of one image can directly be transformed to the set of points on the other image. The algorithms in the last two paragraphs are each based on such a geometrical transformation.

4.2.1 Matching algorithm: 1 initial match point

All moles in the reference and match image are represented as a point with an unique position indicated by the vector $\hat{A}_i$ for the reference points and $\hat{B}_j$ for points of the match image. All the points in the reference image can be seen as the point set $R = \{\hat{A}_1, \hat{A}_2, \ldots, \hat{A}_K\}$ the points in the match image can be represented by the point set $M = \{\hat{B}_1, \hat{B}_2, \ldots, \hat{B}_L\}$.

Let the point pair $\left(\hat{A}_p, \hat{B}_q\right)$, $1 \leq p \leq K$ and $1 \leq q \leq L$, be the confirmed matching pair provided by the user or by an algorithm. The idea is now to translate both set $R$ and $M$ in such a way that the points $\hat{A}_p$ and $\hat{B}_q$ are translated to the origin, resulting in the new sets $R'$ and $M'$:

$$R' = \{\hat{A}_1 - \hat{A}_p, \hat{A}_2 - \hat{A}_p, \ldots, (\hat{A}_K - \hat{A}_p)\} = \{\hat{A}'_1, \hat{A}'_2, \ldots, \hat{A}'_K\}$$

$$M' = \{\hat{B}_1 - \hat{B}_q, \hat{B}_2 - \hat{B}_q, \ldots, (\hat{B}_K - \hat{B}_q)\} = \{\hat{B}'_1, \hat{B}'_2, \ldots, \hat{B}'_L\}$$

After the translation the $M'$ set is rotated around the origin over the angle $\theta$:

$$M'' = \begin{bmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{bmatrix} \cdot M' = \{\hat{B}_{1r}, \hat{B}_{2r}, \ldots, \hat{B}_{rL}\}$$

For this angle a correlation value between the sets $R'$ and $M''$ is calculated:

$$C(\theta) = \sum_{i=1}^{K} \sum_{j=1}^{L} \frac{1.0}{\|\hat{A}_i - \hat{B}_j\| + \varepsilon}$$

The value of $C(\theta)$ is calculated for many values of $\theta$ over the range $[-\frac{\pi}{2}, \frac{\pi}{2}]$. When two points of $R'$ and $M''$ are close together then their term in the sum of $C(\theta)$ will have a high value and contribute to a high correlation value. The angle for which $C(\theta)$ has its maximum is considered to be the best fit, here the points of $R'$ and $M''$ almost coincide with each other, producing high term values in the sum. The variable $\varepsilon$ is a small number, e.g. $10^{-3}$. This makes sure that no division by zero
Registration

will occur if a point from $R'$ and $M'$ coincide, making $\|A'_i - B'_j\|$ equal to zero. To reduce noise in $C(\theta)$ when locating the maximum, the correlation values are smoothed using a running average filter. An example of such a filtered $C(\theta)$ can be seen in Figure 4.1. The set $M''$ for which $C(\theta)$ reaches its maximum value is called $M''_{max}$. Matches or point pairs are made by finding for every reference point in $R'$ the nearest match point in $M''_{max}$ and vice versa. Only if a reference point from $R'$ and a match point from $M''_{max}$ are each others mutually closest neighbours, then they are confirmed as being a matching pair. For all the points for which no matching point in the other image is found represent the possible new moles on the body of the patient.

![Figure 4.1 Correlation as function of the angle of rotation after smoothing using a running average filter.](image)

4.2.2 Matching algorithm: 2 initial match points

If there are two initial matching pairs known, $(A'_{p_1}, B'_q)$ and $(A'_{p_2}, B'_q)$, then it is possible to eliminate the correlation step and perform a geometric transform. This transformation consists of three steps. First both sets are translated so that $A'_{p_1}$ and $B'_q$ are at the common origin.

$$
R' = \{A'_1 - A'_{p_1}, A'_2 - A'_{p_1}, \ldots, (A'_K - A'_{p_1})\} = \{A'_1, A'_2, \ldots, A'_K\}
$$

$$
M' = \{B'_1 - B'_q, B'_2 - B'_q, \ldots, (B'_L - B'_q)\} = \{B'_1, B'_2, \ldots, B'_L\}
$$

Then both sets are rotated in such way that points $A'_{p_2}$ and $B'_q$ are on a common axis, resulting in the sets $R''$ and $M''$. Here we take the x-axis as the common axis.

$$
A''_i = \frac{1}{\|A'_{p_2}\|} \begin{bmatrix} A'_{p_2} & A'_{p_2} \\ -A'_{p_2} & A'_{p_2} \end{bmatrix} \cdot A'_i
$$

$$
B''_i = \frac{1}{\|B'_q\|} \begin{bmatrix} B'_q & B'_q \\ -B'_q & B'_q \end{bmatrix} \cdot B'_i
$$
The last step is a stretch or scale operation. The points in set \( M' \) are scaled in such a way that the points \( \tilde{A}''_{p_2} \) and \( \tilde{B}'_{q_2} \), which lie on the x-axis coincide exactly.

\[
\tilde{B}'_{q_2} = \frac{\tilde{A}''_{p_2}}{\tilde{B}'_{q_2}} \cdot \tilde{B}'_{q_2}
\]

Once this transformation has been performed, the mutually closest neighbours are determined between \( R'' \) and \( M''' \) which are defined as matching point pairs. In Figure 4.2 the 2-point transformation process is shown.

4.2.3 Matching algorithm: 3 initial match points

If three initial match pairs are given, \( (\tilde{A}_{p_1}, \tilde{B}_{q_1}) \), \( (\tilde{A}_{p_2}, \tilde{B}_{q_2}) \) and \( \tilde{A}_{p_3}, \tilde{B}_{q_3} \), the two point geometric transformation can generalized into an affine transformation:

\[
\tilde{A}_i = (\tilde{a}_1 \ \tilde{a}_2) \cdot \tilde{A}_i + \tilde{a}_3 \quad \tilde{a}_n = \begin{pmatrix} a_{nx} \\ a_{ny} \end{pmatrix}, \quad n = 1..3
\]

\[
\tilde{B}_i = (\tilde{b}_1 \ \tilde{b}_2) \cdot \tilde{A}_i + \tilde{b}_3 \quad \tilde{b}_n = \begin{pmatrix} b_{nx} \\ b_{ny} \end{pmatrix}, \quad n = 1..3
\]
The vector constants \((\hat{\alpha}_n, b_n, n = 1, 2, 3)\) can be found using any standard method of solving simultaneous linear equations. For the vector constants \(\hat{\alpha}_n, n = 1, 2, 3\), this results in:

\[
\begin{align*}
\begin{pmatrix} a_{1x} \\ a_{2x} \end{pmatrix} &= M^{-1} \cdot \begin{pmatrix} B_{q1x} - B_{q2x} \\ B_{q1x} - B_{q3x} \end{pmatrix} \\
\begin{pmatrix} a_{1y} \\ a_{2y} \end{pmatrix} &= M^{-1} \cdot \begin{pmatrix} B_{q1y} - B_{q2y} \\ B_{q1y} - B_{q3y} \end{pmatrix} \\
\begin{pmatrix} a_{3x} \\ a_{3y} \end{pmatrix} &= \begin{pmatrix} B_{q1x} \\ B_{q1y} \end{pmatrix} - \begin{pmatrix} a_{1x} & a_{2x} \\ a_{1y} & a_{2y} \end{pmatrix} \cdot \begin{pmatrix} A_{p1x} \\ A_{p1y} \end{pmatrix}
\end{align*}
\]

With:

\[
M = \begin{bmatrix}
A_{p1x} - A_{p2x} & A_{p1y} - A_{p2y} \\
A_{p1x} - A_{p3x} & A_{p1y} - A_{p3y}
\end{bmatrix}
\]

\[
\hat{\alpha}_n = \begin{pmatrix} a_{nx} \\ a_{ny} \end{pmatrix}, \quad \hat{b}_n = \begin{pmatrix} a_{nx} \\ a_{ny} \end{pmatrix}, \quad \hat{A}_{p_n} = \begin{pmatrix} A_{pnx} \\ A_{pny} \end{pmatrix}, \quad \hat{B}_{p_n} = \begin{pmatrix} B_{pnx} \\ B_{pny} \end{pmatrix}, \quad n = 1...3
\]

Using these formulas the set of reference points \(R = \{\hat{A}_1, \hat{A}_2, \ldots, \hat{A}_K\}\) can be transformed to \(R'\). After this sets \(R'\) and \(M\) are overlaid and the initial match points will coincide now. The points are now matched by pairing the points in \(R'\) and \(M\) that are each others mutually closest neighbours as is the previous matching algorithms.

This technique will only work if the three initial match points are not colinear. Even when the three points are only approximately colinear, the transformation becomes too sensitive to small deviations in the point positions of the initial matches. To avoid this problem the initial points must form a triangle of which the three interior angles must be above some predetermined threshold. This threshold is called the "Minimum angle".

The computational requirements of the whole 3 points registration algorithm are of \(O(N^2)\) because of the mutual nearest neighbour approach which is used find the matches. In this part of the algorithm all possible individual distances between reference and match moles are calculated.

### 4.2.4 Comparison of the algorithms

All three algorithms are implemented in Khoros, more as to how this is done and what kind of parameters they have will be described in Chapter 8.

In the article of Perednia and White [18] the algorithms are compared with each other on the basis of data that is collected from real images of the back of a person on which artificial moles were drawn with an ink marker. The images cover an area of about 150x150mm\(^2\). A wide range of images were obtained from the subject. These images included rapid repeated imaging of a still subject, repeated images in which the subject was completely moved and repositioned (sit/stand), positioning the camera at different angles across the surface of the skin to simulate extreme effects of contour distortion (panning), and rotating the camera parallel to the imaging plane. Using this set of images the three algorithms were tested under a wide range of experimental conditions.

Under all of the imaging conditions it is the 3 point algorithm that performs the
best, if the 3 initial point pairs are chosen in such a way that the algorithm does not become unstable. The next best is the 2 point algorithm which is a reasonable replacement for the 3 point algorithm if it is not possible to satisfy the minimum angle criterion.

The repeated imaging conditions and rotation do not prove to be a problem for any of the algorithms, as could be expected. The most problematic are the panned images, the error rate is considerably high (the average percentage of the incorrectly matched points is respectively 13%, 17% and 28% for the 3-point, 2 point and 1 point algorithm). The images for the SCDS are taken with about the same patient position, which implies that panning is not a major concern. In the sit/stand images the 3-point algorithm reaches an error rate of 3.1%, significantly lower than the 2 point (6.7%) and 1 point algorithm (10.2%). The fact that the 3-point algorithm appears to be the best algorithm of those mentioned in the article and the fact that the statistical analysis is very time consuming are the reasons that only this algorithm is tested on the images that are used for the SCDS. The next paragraph presents the statistics of the 3-point algorithm when tested on these SCDS images.

### 4.2.5 Testing the 3 point algorithm on SCDS images

The 3 points algorithm is statistically tested on the images that were taken to be used in the SCDS. At the moment of this study there were only 10 complete sets of images available and only 2 of these were usable for this test. Only these two sets had images with a large enough quantity of moles, necessary for testing the statistical behaviour of the algorithm as function of the number of moles in the image, and were taken in the correct way (i.e. arm horizontal in both reference and match image). From these sets the images of the back of the upper trunk are used for testing, both images show 60 moles.

![Figure 4.3 Percentage of correctly found mole pairs as a function of the number of moles in the image and the number of removed reference moles.](image1.png)

![Figure 4.4 Percentage of incorrectly found mole pairs as a function of the number of moles in the image and the number of removed reference moles.](image2.png)

From these image pairs all test images are generated, starting with the first image pair, by selecting n reference moles from the reference image and the n corresponding match moles from the match image. Then m reference moles are randomly selected and removed to simulate the appearance of new moles in the match images. This is repeated 3000 times. The same is done for the other image pairs. The result is 6000 pairs of test images of which the match image count n moles and the reference image n-m moles. These test images are generated for the
values \( n = 10, 12, 14, \ldots, 60 \). For every value of \( n \) the number of removed reference moles (\( m \)) takes the values 0, 2 and 5. In every image pair three initial match pairs are selected at random. This whole process is performed by the Khoros program Randomize, see Chapter 8.

The images and the initial match pairs are presented to the 3 point algorithm, implemented as the Khoros program PntRegistrationArt3, after which the results are checked by the Khoros program Check and the statistics are made up by the program Stat. The results of the statistical analysis are shown in Figure 4.3, 4.4 and 4.5.

The first one shows the percentage of moles that were matched correctly, Figure 4.4 shows the percentage of moles that were incorrectly matched and the last Figure depicts the percentage of moles that were not found by the algorithm. From these Figures it can be seen that the overall performance goes down when the number of moles in the images increases. Figure 4.3 shows that the percentage of correctly matched moles increases if the number of removed reference moles (or number of new match moles) increases. This is odd since the number of incorrectly matched moles also increases, though less drastically. The contradiction is explained by Figure 4.5 in which it can be seen that the increase in removed reference moles causes the algorithm to miss less mole pairs.

The results shown in Figures 4.3, 4.4 and 4.5 do not take in account the minimum angle criterion, that is why the percentage of correctly matched moles does not reach the 97% mentioned the Perednia-White article. If a single case of 6000 test images, with 45 reference and 50 match moles is taken and the results are sorted according to the smallest angle of the triangle that is formed by the three initial match points then the results can be plotted as in the Figure 4.6. This Figure shows clearly that the performance of the algorithm is very much dependent on the minimum angle of the initial mole pair triangle.

The percentage of 97% as stated in [18] is reached if the initial points are chosen in such a way that the minimum angle exceeds about 0.5 radians or 30°. Thus in order to get a high performance using this 3 point algorithm, quite severe constraints are placed on the selection of the initial match pairs. This means that the algorithms that selects the initial mole pairs must produce more than three of them from which the best ones are selected. The drawback of this is, as will be shown in paragraph 4.2.5, that the error rate of the initial mole pair selecting algorithm increases if the number of generated initial pairs increases. For this
reason a new matching algorithm was developed which is described in the next paragraph. Its performance does not depend on the placement of the initial mole pairs and it takes only 2 initial mole pairs thereby reducing the error rate of the initial mole pair selecting algorithm.

4.3 The baseline algorithm

This algorithm was developed because the results of the 3 points match algorithm were not satisfactory enough to be implemented in the SCDS. The performance of the 3-point algorithms of Perednia and White reaches about 97% for finding mole pairs correctly is quite reasonable but the constraints put on the distribution of the initial mole pairs to get this performance are too high, see paragraph 4.2.5. The new algorithm, the baseline algorithm, performs better than the 3 point algorithm when tested on the same data (more than 98.8% of the mole pairs are found correctly, almost independent of the number of moles in the image). It needs only 2 initial mole pairs and the performance is independent of their distribution.

4.3.1 Description of the algorithm

The principle of the algorithm is explained by means of Figure 4.7, in which the top images represent the match images and the bottom ones the reference images. It starts by defining the line between the two initial point pairs as a baseline (hence the name of the algorithm), the line between the points in the reference images is called the reference baseline, the one in the match image is called the match baseline, both are indicated by the letter a in Figure 4.7a. In the reference image

![Figure 4.7 The baseline algorithm in action.](image)
the point that is the closest to the reference baseline is taken as a third point of a triangle, which is formed by the reference point and the two points of the baseline (the corners are labelled 1, 2 and 3 in Figure 4.7a). Of this triangle some geometrical properties, see paragraph 4.3.2, are calculated which are used for some kind of similarity metric. In the match image these geometrical properties are calculated for every point with the points of the match base forming the other corners of the triangle. The point in the match image that has the most similar geometrical properties when compared to those of the point in the reference image, is said to be the matched point, see point 3 in Figure 4.7a. This reference and match point are defined as a point pair. Together with the two points of the baseline two new baselines can be defined, namely baseline b and c in Figure 4.7b. The reference point is labelled as “registered” and will never again be selected as being the closest to any baseline. Now the point in the reference image is taken that is the closest to one of the available baselines, which is reference point 4 and baseline c in Figure 4.7b. For this reference point and the selected baseline the most similar matching point in the match image is determined on the basis of the similarity metric, in Figure 4.7b match point 4 was selected as most similar. This results in a new point pair with which again two new baselines are formed, namely baseline d and e in Figure 4.7c. Again the reference point is labelled as “registered” and will not take part in the process any more. This process goes on and on until all reference points are matched (paired) to a point in the match image, Figure 4.7d, 4.7e and 4.7f. In the end all points in the match image that are not paired with a reference point are defined as unmapped and represent the new points that do not occur in the reference image.

In the algorithm as described performs well as long as every reference point has a corresponding match point but a problem arises if this is not the case. see Figure 4.8a. The algorithm above would just take the best fitting match point. This problem arises e.g. when a mole in the reference image was diagnosed as melanoma and excised before the match image was made in the follow up photo session. To overcome this problem an extra step is added after finding the most similar point in the match image, see Figure 4.8a. The step of finding the most similar match point is reversed. For all the points in the reference image the geometric properties are calculated with respect to the reference baseline (baseline a in Figure 4.8b) and using the similarity metric the reference point is found that is the most similar to the point in the match image, found in the previous step, see

![Figure 4.8 Finding point pairs when a reference point does not have a corresponding point in the match image.](image-url)
Figure 4.8b. This references point is a different one than point 3 in Figure 4.8a, from which the matching process started. This means that the points labelled with number 3 in Figure 4.8a are not a point pair. The reference point is labelled as "registered" and will never again be selected as being the closest to a baseline. The next reference point being selected as the closest to a baseline is point number 3 in Figure 4.8b. For this point the match point labelled number 3 in Figure 4.8b will found as most similar. Now for this point in the match image the most similar reference point is found, which is point number 3 in Figure 4.8b. So we returned to the same reference point as where we started from. This means that this reference and match point form a point pair. The point pair results in two new baselines, namely baseline b and c in Figure 4.8c.

In the steps listed below the algorithm is described in a more accurate and structured fashion. The phrases printed in bold are definitions and labels which are used in Figure 4.7 and 4.8.

1. Label all reference points as **unregistered**
2. Define the 2 reference initial points as **current ref base**
   Define the 2 match initial points as **current match base**
   These two bases represent the same base in reference and match image and are combined and registered as a base pair. Together they are defined as **current base pair**
3. Define the reference point that is the closest to the midpoint of the **current ref base** as **current ref point**
4. While not all reference points have the label **registered**
   1. Calculate the geometric properties of the **current ref base - current ref point** combination.
   2. Calculate the geometric properties of all the points in the match image with respect to the **current match base**
   3. Compare all match point geometric properties to the **current ref base - current ref point** geometric properties using the similarity metric and select the match point with the most similar geometric properties. Define the point as **matched match point**
   4. Calculate the geometric properties of all the points in the reference image with respect to the **current ref base**
   5. Compare all reference point geometric properties to the **current match base - matched match point** geometric properties using the similarity metric and select the reference point with the most similar geometric. Define the point as **matched ref point**
   6. If **matched ref point** and **current ref point** are the same points then
   1. Register **current ref point** and **matched match point** as a point pair and define it as **current point pair**
   2. Register point pair 1 of **current base pair** and **current point pair** as a new base pair
   3. Register point pair 2 of **current base pair** and **current point pair** as a new base pair
   End if.
4. Label **current ref point** as **registered**
5. Find a reference point with the label **unregistered** and a basepair for which holds that the midpoint of the reference base and the position of the reference point are the closest possible combination
Define the found reference base pair as **current ref base**. The corresponding match base is define as **current match base**. Together these two bases form the **current base pair**.

Define the found reference point as **current ref point**

End while

(5) Register all reference and match points that are not registered as point pairs as unmapped

The computational requirements of the baseline algorithm mainly depend on the part of the algorithm that finds the closest possible combination of reference baseline and reference point (step 4.8). At the start of the registration process there is only one baseline and N points that need to be registered. As the process advances, the number of baselines grows and the number of not registered reference points decreases. At the last step of the registration process there are more than N baselines and only one not registered reference point left. So the computational requirements of this search process of comparing the distances between all possible not registered reference points and baselines are on average O(N^2) and since this process is repeated N times to register all N reference moles, the whole algorithm is of O(N^3).

### 4.3.2 The similarity metric

A similarity metric is needed to determine whether or not the combination of a baseline and a point in the reference image corresponds to a baseline-point combination in the match image. This similarity metric is based on spatial information only. A way of characterizing a point with respect to a base is shown in Figure 4.9a. The distances \( l_1, l_2 \) and the angles \( \alpha, \beta \) are characteristic geometric properties of the point \( P \) with respect to the baseline. If the differences between the reference and match image could all be accounted for with a rigid-body motion then only the angles or only one distance and one angle would be enough to characterize the point with respect to the baseline. In practice, however, it is not a rigid-body motion because of effects like elastic and contour distortion. By using all four geometric properties the algorithm will be more robust. There however is one problem, to be able to compare angles the angles must be between 0 and \( \frac{\pi}{2} \) (always the smallest angles between two lines) which means that characterizing as is done in Figure 4.9a is ambiguous. This is shown in Figure 4.9b. A point that is almost the mirror image, when mirrored in the baseline, of point \( P \) will produce about the same values for \( l_1, l_2, \alpha \) and \( \beta \). This problem is solved by adding a third point, which is derived from the baseline, by rotating the baseline pair over an 90° angle

![Figure 4.9 Characterizing point P with respect to a baseline B1-B2.](image)
The baseline algorithm around baseline point 2, see Figure 4.9c. The extra properties $l_3, \gamma$ will ensure a unique characterization. If the positions of the baseline point 1 and 2 are represented by the vector $\hat{b}_1$ and $\hat{b}_2$ then the third point $\hat{b}_3$ can be derived in the following way:

$$\hat{b}_3 = \hat{b}_2 + \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix} \cdot (\hat{b}_2 - \hat{b}_1)$$

When the geometric properties are derived in the reference image they are called $l_{R1}, l_{R2}, l_{R3}, \alpha_R, \beta_R$ and $\gamma_R$, for the match image they are called $l_{M1}, l_{M2}, l_{M3}, \alpha_M, \beta_M$ and $\gamma_M$. The similarity metric is now defined as an error:

$$Error = \frac{|l_{R1} - l_{M1}|}{l_{R1}} + \frac{|l_{R2} - l_{M2}|}{l_{R2}} + \frac{|l_{R3} - l_{M3}|}{l_{R3}} + \frac{|\alpha_R - \alpha_M|}{\alpha_R} + \frac{|\beta_R - \beta_M|}{\beta_R} + \frac{|\gamma_R - \gamma_M|}{\gamma_R}$$

The error value is smaller if the reference and match baseline-points combination are more similar to each other.

### 4.3.3 Testing the baseline algorithm on SCDS images

The baseline algorithm is implemented as a Khoros program that goes by the name PntRegistrationOwn. This program is subjected to the same test data as the 3 point algorithm in paragraph 4.2.5. The results of this statistical analysis are presented in Figure 4.10, 4.11 and 4.12.

![Figure 4.10](image1.png) **Figure 4.10** Percentage of correctly found mole pairs as a function of the number of moles in the image and the number of removed moles.

![Figure 4.11](image2.png) **Figure 4.11** Percentage of incorrectly found mole pairs as a function of the number of moles in the image and the number of removed moles.

All Figures show that the performance of the algorithm is almost independent of the number of mole in the image. Also the number of missing reference moles has little influence. The overall ability to find correct matches is maintained at about 99%, which is about 2% more than the Perednia-White 3 point algorithm supplied with correctly chosen initial point pairs.

The critical parameter for the success rate of the 3 point algorithm is the minimum angle of the triangle formed by the 3 initially matched points. The baseline algorithm might have such a dependency for the distance between the initial mole pairs. The idea is that a small distance will make the algorithm better capable of following the curvature caused by elastic deformation than a large distance. To test this, images containing 45 reference and 50 match moles are processed by the algorithm. The results were sorted as function of the distance between the initial mole pairs using a histogram. The resulting graph is shown in Figure 4.13.
this graph follows that the distance between the initial mole pairs has almost no influence on the performance of the baseline algorithm. Apparently a long initial baseline formed by the initial two mole pairs is quickly broken down into many smaller ones that do follow the curvature caused by the elastic deformation.

4.4 Comparing the 3 point and the baseline algorithm

The ability of the 3-point registration algorithm to register the moles correctly is very much dependent of the number of moles in the image. A higher number of moles leads to a decreased performance. The performance is strongly dependent on the distribution of the initial matches. When the three initial matches satisfy the minimum angle criterion, i.e. form a triangle of which the smallest angle is larger than 30°, then the algorithm finds the correct matches in about 97% of the cases. The baseline registration algorithm only needs 2 initial match points of which the distribution is of no concern. Furthermore, the number of moles in the image has not much influence on the ability to make the correct matches. Over the range of 10 to 60 moles in the image this ability is maintained at about 99% and deviates less than 1%. Also the removal of reference moles (to represent new match moles) does not effect the baseline algorithm as much as the 3-point algorithm. The drawback of the baseline algorithm is that its computational requirements which are of O(N3) while the requirements of the 3-point algorithm are only O(N2).

4.5 Summary

Three algorithms from literature were introduced of which the 3 point algorithm appeared to be the best. But because of the high constraints it placed on the distribution of the initial matches a new algorithm was developed, the baseline algorithm. The baseline algorithm has the ability to find the correct matches in about 99% of the cases when tested on SCDS images. Where as the 3 point algorithm reaches 97% if the distribution of the initial matches meets the minimum angle criterion.

Furthermore, the performance of the baseline algorithm is largely independent of the number of moles in the image, in contrast to the 3-point algorithm of which the ability to find the correct mole pairs decreases when the number of moles grows. The drawback of the baseline algorithm is that its computational requirements are bigger, namely O(N3) instead of O(N2) for the 3-point algorithm.
Chapter 5  Initial match point selection

The algorithms introduced in Chapter 4 require a number of initially matched point pairs to be able to find the rest of the point pairs in the images. This chapter deals with two of these algorithms for selecting those initial point pairs.
5.1 Initial point pairs

In Chapter 4 a number of algorithms were introduced for which one, two or three initially matched points must be obtained prior to computation. The initial points are used to constrain the registration problem and greatly reduce the amount of processing time needed to find a solution. While it is relatively easy for a human operator to provide these initial matches, registering large numbers of images would be easier if these points could be derived automatically. Human intervention could then be limited to particularly difficult cases.

High probability matches may be found using a variety of methods based on individual lesion characteristics, spatial considerations, or both. Initial matches based only on mole parameters have several drawbacks. In images containing many similar lesions, actual changes occurring within the mole themselves could easily give rise to mismatches. These techniques also use similarity metrics which should be appropriately weighted for each lesion parameter used. Methods based on the spatial patterns of many moles tend to be more robust, but their computational requirements must be examined carefully. Relaxation methods have the potential of performing well in selecting initial matches but are iterative in nature and do not have a fixed running time. The resulting computational costs can be too much for microcomputer based systems. Another possible spatial matching technique is to treat each mole (point) as a node in a graph. Each node is connected to every other node representing a point in the image. The edges connecting any two nodes have two values associated with them: the distance between the nodes and the orientation of the edge relative to some arbitrary axis. To select initial matches, each node of the graph of one image is compared to each node in the graph in the other image. A metric is then used to compare the edges of the two nodes and rate their similarity. This approach is both robust and capable of being computed in a fixed number of steps. Unfortunately, brief inspection of this technique shows that the number of steps is \( O(N^4) \) (\( N \) operations to calculate the similarity metric, every node has \( N \) different orientations if the edges are sorted according to their angle and \( N \) of those nodes have to be compared to \( N \) other node, which results in \( N^4 \) operations) if all possible node pairs are considered. This is a direct consequence of connecting each node to every other node and leads to unacceptable running times.

In [19] Perednia and White address this problem by reducing the number of edges associated with each node while maintaining the same overall approach of comparing nodes across images with a similarity metric. In [19] a Gabriel graph is used to reduce the number of edges. This type of graphs is very consistent from image to image and is globally insensitive to local changes in the point pattern (such as adding or removing a point). Furthermore, the graphs are sufficiently sparse that the running time required to select initial matches is almost always of \( O(N^2) \). The Gabriel graphs themselves are generated with running times of \( O(N^3) \). The method of Perednia and White is worked out further in paragraph 5.2.

Another method to make a sparser graph is to connect a node only to a few of its closest neighbours. This kind of graph can be generated in \( O(N^2) \) steps and selecting the initial matches is done in the same number of steps. This method will be introduced in paragraph 5.3.

5.2 The Gabriel graph algorithm

This algorithm for selecting initial matches was introduced by Perednia and White [19]. It is based on the work of Skolnick [20][21] who used Gabriel graphs for the
automatic comparison of two-dimensional electrophoresis gels. In [19] the same assumptions are made about the images being point matched as in paragraph 4.2. First is explained what a Gabriel graph is and how it can be generated, this is followed by the actual point matching algorithms where the Gabriel graphs of the reference and the match image are compared to select the initial matches. The last paragraph is devoted to testing the algorithm on SCDS images.

5.2.1 The Gabriel graph

The Gabriel graph is a planar graph that belongs to the family of the relative neighbourhood graphs. The definition of it is fairly simple. Any node is said to be adjacent (i.e. connected by an edge) to another node B, if and only if for every other node C, point C does not lie within the circle of the diameter $|AB|$ centred at the midpoint of the line between A and B. It can be shown that, because the decision to label a point as being adjacent relies solely on the position of the other surrounding points and is made independently of other adjacency decisions, this definition results in a unique graph. Figure 5.1 explains this definition graphically and gives an example of a Gabriel graph.

![Figure 5.1](image)

Figure 5.1 a) Two node are connected if the circle, that has its centre on the midpoint of the line connecting the two nodes and that has a diameter equal to the distance between the two nodes, does not contain any other nodes. This means that the points Q and R and the points P and R are connected since C3 and C2 do not contain other nodes but the points P and Q are not connected because the point R lies within their circle C1. b) This is an example of a Gabriel graph.

This definition requires $O(N^3)$ because all potential edges ($N^2$) between nodes must be compared to all other nodes (N) to see if the edge criterion is met. In practice, the time required for generating can be shortened by setting bounds as to the maximum edge length, so that only comparisons within the neighbourhood defined by that length need to be considered.

5.2.2 The principle initial point selecting algorithm

The first step in selecting initial matches is to generate the Gabriel graphs of both reference and match image. Each possible initial match is tested, which means taking one node from the reference image Gabriel graph and one node of the match image Gabriel graph and computing a similarity metric for those two nodes. If the two nodes do not have the same number of adjacent nodes, then the pair is immediately rejected as being a potential initial match. Otherwise the edges are ordered by angle (e.g. clockwise) from some arbitrary axis. The algorithm must be insensitive to rotation between successive images, that is why nothing is known about the orientation of the edges and every possible combination of ordered edges
is tested. For example, assume that the node from the reference image has four (ordered) edges labelled \((a, b, c, d)\), see Figure 5.2, and the node in the match image has edges (also ordered) labelled \((a', b', c', d')\). Then there are four possible mapping that could occur without restricting image rotation. These mapping are: \((a, b, c, d) \rightarrow (a', b', c', d')\), \((a, b, c, d) \rightarrow (b', c', d', a')\), \((a, b, c, d) \rightarrow (c', d', a', b')\) and \((a, b, c, d) \rightarrow (d', a', b', c')\).

![Figure 5.2 Node and its anti-clockwise ordered edges \((a, b, c, d)\).](image)

Notice how the angular ordering of the symbols for the match node is always preserved. For each of these mappings an error measure is calculated using the following formula:

\[
Error = \sum_{i=1}^{n} \frac{|D_r(i) - D_m(i)|}{D_r(i) + D_m(i)} + \sum_{i=1}^{n} \frac{|\theta_r(i) - \theta_m(i)|}{\theta_r(i) + \theta_m(i)}
\]

Where

- \(D_r(i)\) = the distance associated with edge \(i\) of the reference Gabriel graph for the reference node being tested;
- \(D_m(i)\) = the distance associated with edge \(i\) of the match Gabriel graph for the match node being tested;
- \(\theta_r(i)\) = the angle between edge \(i\) and its succeeding edge \(i+1\) of the reference Gabriel graph for the reference node being tested;
- \(\theta_m(i)\) = the angle between edge \(i\) and its succeeding edge \(i+1\) of the match Gabriel graph for the node being tested.

The smallest error value for all mappings is kept as the value of the similarity metric for the nodes being tested. Once every possible combination of nodes between reference and match image has been tested then those node pairs with the lowest similarity metric are selected as being true initial matches.

### 5.2.3 Testing the Gabriel graph algorithm on SCDS images

The Gabriel graph algorithm is implemented as a Khoros program named InitPntArt, see Chapter 8. This program is subjected to the same test data as the 3 point algorithm in paragraph 4.2.5. The results of this statistical analysis is shown in Figure 5.3 and 5.4.

The Gabriel graph algorithm selects its initial matches on the basis of a similarity metric and possible matches are sorted according to their similarity. In this way a list of matches is produced in which the most likely initial match is listed as the first rank, the second best match holds the second rank etc. In Figure 5.3 this list is shown for an image pair that contains 50 match moles and 45 reference moles. For each rank the percentage of the cases is indicated for which the match of a
The local node algorithm 37
certain rank in the list is correct.
The registration algorithms in Chapter 4 have to be provided with one, two or three
initial matches, therefore an important measure for the performance of the
algorithm will be its ability to select three initial matches correctly. To calculate
the chance of the first n ranks being correct, the percentages of those ranks have
to be multiplied together. In Figure 5.4 the percentage of the cases is shown in
which the first three selected matches are correct. It appears that the ability to
select the first three initial matches correctly is almost independent of the number
of moles in the image pair. In more than 99.2% of the cases all three selected
matches are correct. Also removing reference moles does not seem to have much
influence on the performance, the ability to select correct matches decreases only
slightly when the number of removed reference moles increases. Only when a
relatively large number of reference moles is removed (e.g. in the situation with
around 20 moles in the image pair and 5 removed reference moles, this means that
25% of the reference moles are removed) a large drop in the performance is noticed.

From Figure 5.4 it can be seen that the performance increases again when the
number of moles in the images is larger than 45. This is an artifact caused by the
fact the random image pairs are generated from just original image pairs.

5.3 The local node algorithm

This algorithm is in principle the same as the Gabriel graph algorithm. Every node
in the reference image is compared to every node in the match image on the basis
of a similarity metric that measures the similarity in the distribution of the
neighbours surrounding the node. The only difference is how the neighbours are
defined. In Gabriel graph algorithm the neighbours of a node are defined by the
Gabriel graph which can be generated in $O(N^3)$ steps. The local node algorithm
defines its neighbours much faster, it needs only $O(N^2)$ steps.

5.3.1 The principle of the local node algorithm

This algorithm considers only the two closest nodes to describe the location of a
node relative to its neighbours. The idea behind this is that the two neighbouring
nodes are close enough that the elastic deformation between the reference and the

![Figure 5.3 Performance of the Gabriel initial match algorithm as function of the rank order of the selected initial matches.](image)

![Figure 5.4 Percentage of the cases in which the first three initial matches selected by the algorithm are correct. It is given as function of the number of moles in the image and the number of removed reference moles.](image)
match image does not have any or only very little influence on their position relative to the main node of which the neighbourhood is being described. The main node is combined with the two closest node to form a triplet, see Figure 5.5. Each triplet can be characterized by three parameters, namely the distances \(D_1\) and \(D_2\) from the main node \(n\) to the two others \((n_1\) and \(n_2\)) and the angle between the lines that connect the two closest nodes to the main node of the triplet \(\theta\).

When comparing a triplet from the reference image, say \((n, n_1, n_2)\), to a triplet in the match image \((n', n'_1, n'_2)\), there are only two possible mappings which have to be checked: \((n, n_1, n_2)\) \(\rightarrow\) \((n', n'_1, n'_2)\) and \((n, n_1, n_2)\) \(\rightarrow\) \((n', n'_2, n'_1)\). The mapping for which the reference and match triplet are the most similar, has the following similarity value:

\[
\text{Similarity} = \min \left( \frac{|D_{r1} - D_{m1}|}{D_{r1} + D_{m1}}, \frac{|D_{r2} - D_{m2}|}{D_{r2} + D_{m2}}, \frac{|D_{r1} - D_{m2}|}{D_{r1} + D_{m2}}, \frac{|D_{r2} - D_{m1}|}{D_{r2} + D_{m1}} \right) + \frac{|\theta_r - \theta_m|}{\theta_r + \theta_m}
\]

Where \(D_{ri}\) = the distance between the main node \(n\) and the node \(n_i\) in the reference image;
\(D_{mi}\) = the distance between the main node \(n\) and the node \(n_i\) in the match image;
\(\theta_r\) = the angle between the lines connecting the main node \(n\) and the nodes \(n_1\) and \(n_2\) in the reference image;
\(\theta_m\) = the angle between the lines connecting the main node \(n\) and the nodes \(n_1\) and \(n_2\) in the match image.

This similarity value is calculated for every possible combination of node pairs of nodes in the reference and nodes of the match image. The node pairs with the lowest similarity value are selected as initial matches.

### 5.3.2 Testing the local node algorithm on SCDS images

The local node algorithm is implemented as the Khoros program InitPntOwn. This program takes the mole positions of reference and match image in the Mole Pattern File as input and generates a list of selected initial matches and writes them to the Initial Mole Pair File. The list is ordered according to their similarity value, which means that the most likely match is in the first place, the next most likely in the second and so on.

The program is subjected to the same test data as the 3 point algorithm in paragraph 4.2.5. The results of this statistical test are shown in Figure 5.6 and 5.7. Figure 5.6 shows the ranks in the ordered list and indicates the percentage of cases in which a certain rank contains correctly selected initial matches for an image pair with 50 match moles and 45 reference moles. The percentage of cases in which the first three selected initial matches are correct is an important criterion for the registration algorithms of Chapter 4. These
percentages are drawn in Figure 5.7 as function of the number of moles and the number of removed reference moles.

Figure 5.6 Performance of the local node initial match algorithm as function of the rank order of the selected initial matches

Figure 5.7 Percentage of the cases in which the first three initial matches selected by the algorithm were correct. It is given as function of the number of moles in the image and the number of removed reference moles.

5.4 Comparing Gabriel graph and local node algorithm

Choosing between the Gabriel algorithm and the local node algorithm is trade off between speed and performance. When Figures 5.4 and 5.7 are compared then it is clear that the Gabriel graph algorithm produces more likely initial matches than the local node algorithm (if the number of removed nodes is small compared to the total number of nodes) but selecting initial mole pairs using the Gabriel graph algorithm is an $O(N^3)$ process, while the local node algorithm is only $O(N^2)$. Furthermore the local node algorithm is less sensitive to the removal of moles than the Gabriel graph algorithm.

5.5 Combined initial match and registration performance

In practice the initial points are selected by the initial match algorithm and passed on to the registration algorithm. In the next two paragraphs it is tested what the combined performance is. Only the combination of the Gabriel graph algorithm and the baseline algorithm and the combination of the local node algorithm and the baseline algorithm are tested. No tests are performed using the 3 point registration algorithm because 1) the performance is quite poor compared to the baseline algorithm if nothing is done about the minimum angle criterion and 2) if the minimum angle criterion is taken in account then the performance of the initial match selecting algorithm goes down since more initial moles have to be selected correctly in order to find three initial matches that apply to the minimum angle criterion.

5.5.1 Gabriel graph algorithm and baseline algorithm

In Figure 5.8 the result is shown when the total registration process is tested on the test images of 4.2.5, using the Gabriel graph algorithm for selecting the initial matches and the baseline algorithm for registering the rest of the moles. On average this combination finds the correct mole pairs in 98-99% of the cases. Furthermore, the ability to find correct mole pairs is relatively independent on the
number of moles in the images. Only when a relatively large number of moles is removed (e.g. 20% of the total number of moles) the performance drops under 98%.

In Figure 5.8, when no reference moles are removed, the performance shoots up until a 100% in the right part of the graph. This is an artifact caused by the fact the random image pairs are generated from just original image pairs.

5.5.2 Local node algorithm and baseline algorithm

If the local node algorithm, selecting the initial matches, and the baseline algorithm registering the rest of the moles, are combined in the registration process then this results in Figure 5.9 when tested on the images of 4.2.5. On average this combination finds the correct mole pairs in about 97.5-98.5% of the cases.

When compared to the combination of the Gabriel graph algorithm and the baseline algorithm this combination performs less, but on the other hand it is less sensitive to the removal of relatively large numbers of moles. The combination is just as the Gabriel graph/baseline algorithm combination of $O(N^3)$.

Figure 5.8 Percentage of correctly registered moles when the Gabriel graph algorithm (selecting initial matches) and the baseline algorithm (registration) are combined.

Figure 5.9 Percentage of correctly registered moles when the local node algorithm (selecting initial matches) and the baseline algorithm (registration) are combined.
5.6 Summary

This chapter introduced two algorithms for the selection of the initial matches, needed in the registration algorithms of Chapter 4. The Gabriel graph algorithm was tested as being the best of the two when looked at the ability to select the first three initial matches correctly (above 99.2%), however it is slower than the local node algorithm. The local node algorithm, which selects the first three initial matches correctly in more than 98.6% of the cases, is of $O(N^2)$ while the Gabriel graph algorithm is of $O(N^3)$. Choosing between the two is a trade off between speed and performance.

The algorithms of this chapter were combined with the best registration algorithm of Chapter 4, the baseline algorithm, to form the complete registration process. The combination of the Gabriel graph algorithm and the baseline algorithm performed the best (98-99% of the cases the correct matches were made). The local node/baseline combination which is also of $O(N^3)$, finds the correct matches in 97.5-98.5% of the cases. The last combination is less sensitive to the removal of relatively large numbers of moles.
Chapter 6  Feature detection

This chapter explains why the moles are compared on the basis of mole features that are specially tailored for the diagnosis of melanoma. Several mole features are introduced and of some of them it is shown how they are implemented. Finally the relation between the feature accuracy and the image resolution is discussed.
6.1 Features

After registering the moles of the reference and match image, it is known which moles in the reference image correspond to which moles in the match image. The next step in the SCDS is to check whether or not a mole in the match image has changed during the time between the first photo session and the follow up session. The segmentation process only gives information as to which pixels belong to the mole and which pixels are normal skin. A very crude approach to detect changes could be to subtract the reference mole pixels from the match mole pixels and calculate an overall error using the pixel differences. A mole would be defined as changed if the overall error exceeded a certain threshold. In practice, however, this will not be a reliable method because, among other things, the lighting conditions can vary from image to image. The approach taken here is to characterize the moles using features such as area, perimeter, colour variations etc. These features are then used to check if a mole has changed.

Much research has been done in this area of feature extraction, see e.g. [6], [7], [8] and [9]. In most cases these features are specially tuned to diagnose a mole as a benign mole or malignant mole on the basis of just one mole image. This means that these features describe some special characteristic of the mole that gives some indication to whether the mole is benign or malignant. For example a mole with an irregular border has a higher possibility of being malignant than a mole that has a very regular border. Thus a feature that measures the irregularity of the mole border can be used to discern between a malignant and a benign mole. These tailor-made features for the diagnosis of moles are also very useful when comparing moles for changes because if a mole changes from benign to malignant, it will certainly be one of these features that will change drastically.

Most diagnostic features can be summarized in the ABCD-rule which was introduced by Friedman et al. [22] to improve the diagnostic accuracy when diagnosing moles. The mnemonic ABCD stands for features that describe early malignant melanoma:

- Asymmetry: One half of the mole does not match the other half;
- Border irregularity: The edges are ragged, notched and blurred;
- Colour: The pigmentation is uniform. Shades of tan, brown and black are present;
- Diameter: Bigger than 6 mm and growing.

In [6], [8] and [9] different features of a mole in a colour image are suggested and examined, using correlation techniques, as to how useful they are as a diagnostic feature.

The most basic features of a mole are its area and perimeter, using these features the compactness and irregularity index can be calculated. The compactness of an object is defined as the ratio of area to perimeter and it measures if the object is strongly concentrated around a point or whether it has a more elongated structure. The irregularity of the mole border is measured by the irregularity index which is defined as the ratio of the area and the square of the perimeter. Both features are defined in such a way that higher values indicate a higher chance that the mole is malignant. In [7] fractal dimensions are used to describe the irregularity of the mole border.

Other features can be calculated if the centre of mass of the mole is known. Polar distances are distances from the centre of mass to the boundary of the mole. Especially, a high variance in the polar distances correlates with the existence of a
Detecting features in the SCDS 45

Detecting features in the SCDS

In the SCDS the features of the moles in the reference and match images are calculated by the Khoros program DetectFeatures. It uses the mole positions in the Mole Pattern File to locate where the moles are in the images and then copies the mole into a smaller image to increase processing speed. The mole will be at the centre of this image, see Figure 6.2a. Using the marker information in the Mole Pattern File, the mole images are scaled such that the reference and match mole images have the same scale. Otherwise it is not possible to compare the features of the reference and the corresponding match mole. After this the features are calculated. This process of copying, scaling and feature calculation is done for every mole in the reference and match image. The features of the reference moles are written to the Reference Feature File and the match mole features are written to the Match Feature File. These files are later used by the comparison process, see Figure 2.2, which checks for changes between corresponding reference and match moles.

In paragraph 6.4 it is described what features are calculated and how they are implemented in the DetectFeatures program. But first an algorithm is introduced in the next paragraph to determine which pixels belong to the mole. This algorithm replaces, for the time being, the not yet existing segmentation process of the SCDS.

6.3 Separating mole pixel from the background skin

Before any features of a mole can be calculated, the pixels belonging to the mole and the normal skin have to be known. Since the current SCDS does not have a segmentation process (yet), and only the positions of the moles are indicated manually, another way must be applied to identify a pixel as part of the mole or of the background skin. The method adopted here is the radial search algorithm. This algorithm, introduced by Golston et al. [13], provides an accurate way of detecting the border of a mole in an intensity image. The algorithm makes some assumptions about the mole: 1) a point inside the mole is known, this will be called the initial point and 2) the border can be described by a function in polar coordinates, using the initial point as origin (this is called radial connectedness). The first assumption is always true since the centres of the moles are indicated on the marked reference and match images (see Chapter 3). The second assumption depends on the mole and will be discussed later. First the principle of this fairly simple algorithm is given.

The mole in question is first copied from the reference of the match to a smaller
image, called the mole image, and scaled to remove scaling differences between the reference and match images. The mole is now in the middle of the mole image and the centre point of this image will be used as initial point for the algorithm. From this initial point a large number of radial lines are cast until the edge of the mole image. The algorithm searches along these lines, see Figure 6.1a, for a sustained intensity increase that marks the border of the mole. On each radial line such an intensity is defined as border point. The more radial line are used, the more accurate the border will be known. To make the mole border complete, the individual points on the border of the mole are connected by means of a spline interpolation technique to produce a closed curve that defines the border of the mole.

![Figure 6.1](image.png)

*Figure 6.1 a) Definitions of the terms used in the text. b) The border of this mole is not radially connected, the dark area will remain undetected.*

All pixels that lie within this border are defined as mole pixels, the pixels immediately outside the curve are assumed to be normal skin. The drawback of this algorithm is caused by assumption number two (radial connectedness). If the mole border can not be described by a function in a polar coordinate system, with the initial point as origin then some parts of the mole will remain undetected. Figure 6.1b shows a mole that violates the so-called radial connection criterion. In this Figure a radial line crosses the mole border more than once but only one crossing is noted by the algorithm and stored as border point. The dark gray area in this Figure indicates the part of the mole that is not detected. However, the number of moles with borders that can not be found completely by the radial search algorithm is rather small and, although this kind of border does indicate the presents of a malignant mole. The radial search algorithm will suffice for the moment until a proper segmentation process for the SCDS is developed. The next paragraphs describe the algorithm in more detail.

### 6.3.1 Marking flash areas

When a photo is taken of the patient a flashlight is used to properly illuminate the area being photographed. Some shiny areas of the mole will reflect this flash, resulting in high luminosity areas inside the mole, see Figure 6.2. These so-called flash areas cause the radial search algorithm to detect false border points. To solve this problem the pixels in the flash areas are marked in a binary image $f(x,y)$ of
same size as the mole image, where \( f(x,y) = 1 \) for pixels that belong to a flash area and 0 elsewhere. The radial search algorithm does not take these marked pixels into account when searching for border points. Flash areas can be recognized in a colour image by the fact that those areas do not contain much colour when compared to the surrounding mole and skin pixels. This means that the RGB colour components in these areas have about the same value. A pixel is now defined as being part of a flash area if all RGB components deviate less than a certain threshold from the average of the RGB components. The result is shown in Figure 6.2b, where the original image is overlaid with the binary image of the flash areas.

![Figure 6.2](image1.png)  
Figure 6.2 a) The original mole image that is extracted from the bigger image by means of windowing. b) The mole image overlaid with the binary flash image, the flash areas are indicated with black.

### 6.3.2 Radial search algorithm

The radial search algorithm operates on an intensity image, so first the pixels of the mole image are converted to form an intensity mole image, using the formula:

\[
i(x, y) = \frac{R(x, y) + G(x, y) + B(x, y)}{3}
\]

Where 
- \( i(x, y) \) = the intensity value of the pixel at location \((x,y)\);
- \( R(x,y) \) = the red component of the pixel in the colour mole image at location \((x,y)\);
- \( G(x,y) \) = the green component of the pixel in the colour mole image at location \((x,y)\);
- \( B(x,y) \) = the blue component of the pixel in the colour mole image at location \((x,y)\).

Taking the centre of the intensity mole image as initial point, the algorithm casts \( n \) radial lines emanating from this point at equal angles of \( \left( \frac{360}{n} \right) \) degrees. Figure 6.3 depicts a situation with \( n = 5 \) radial lines. Line by line the algorithm searches for a border point along each of the radial lines. The position of the mole border along such a line is found by determining the location where the image intensity suddenly increases. In order to detect this change, first all pixel values along the radial line (from initial point to the edge of the mole image) are sampled. By using the Bresenham line algorithm [24] the pixel positions along the line are calculated. For each pixel the average value of its surroundings, defined by the averaging window of a certain size, is calculated and stored in the array \( P_i \). The averaging acts as a low pass filter and will reduce false border detection by removing image noise caused by pores, hairs etc. The pixels
marked as flash area are not taken into while calculating this average. In Figure 6.4 the values of such an array are drawn. To determine which array position corresponds to the border of the mole a correlation technique is used. The values in the array are correlated with a step function ($u_i$) which serves as model or template for the intensity change at the border. The template function is shifted over the array of pixel values and at each position an error value is determined which indicates the difference between the template and the pixel value array, see Figure 6.4. The position where the error reaches its minimum value is defined as the mole border. The error value at position $i$ is calculated using the formula:

$$error_i = \sum_{j=1}^{N} \left( \frac{P_i - P_{\min}}{P_{\max}} - u_{j-i} \right)^2$$

Where $error_i$ = the error between pixel value array and the template; $P_i$ = the value at position $i$ in the pixel value array; $P_{\max}$ = the maximum value in the pixel array; $P_{\min}$ = the minimum value in the pixel array; $u_i$ = the step function which is 1 if $i \geq 0$ and 0 elsewhere; $N$ = the number of element in the pixel value array.

![Figure 6.3 Sampling the average pixel values along a radial line using an averaging window.](image)

![Figure 6.4 Correlating a template with the intensity values along a radial line to detect the border of the mole.](image)

### 6.3.3 Defining the mole pixels

All the border points that were found along the $n$ radial lines are marked by a 1 in the binary mole image $b(x,y)$. In the mole binary image, which is of the same size as the mole image, all pixels that are part of the mole will be indicated by a 1, the rest is set to 0.

To reduce the computing time, not all border points are detected (by increasing $n$) using the radial search technique. If $n$ is sufficiently large then the other border points can be approximated by means of a cubic spline interpolation [25]. The result is a closed border, indicated by ones in the binary mole image $b(x,y)$. Figure 6.5a shows the original mole image overlaid with the binary mole image after spline interpolation. The closed border makes it possible to mark all pixels inside the border as mole pixels by setting all these pixels to 1 in $b(x,y)$ using a region fill algorithm. Figure 6.5b shows the original mole image overlaid with the resulting binary mole image $b(x,y)$.
6.3.4 Radial search and sensitivity to the initial point

It appears that the border found by the radial search algorithm depends to some extent on the choice of the initial point from which the radial search lines depart. This means that the feature values calculated for the mole will also depend on the position of the initial point, which can cause problems when feature values are compared to detect changes. To reduce this effect the centre of mass of the binary mole image is calculated, see paragraph 6.4.2, which lies somewhere in the centre of the mole. The radial search algorithm detects the border of the mole again but now using the centre of mass as initial point, resulting in a new binary image of the mole. The centre of mass is calculated for this new binary mole image and used again as initial point for the radial search algorithm. After a number of iterations the centre of mass stays stable and so does the detected border, which means that the calculated features will be independent of the first initial point that is chosen for the radial search algorithm.

6.4 Calculating the features

Until now only some of the features mentioned in paragraph 6.1 are implemented in the DetectFeatures program. The currently implemented features are:

- Area;
- Perimeter;
- Compactness;
- Asymmetry index;
- Average of the polar distances;
- Variance of the polar distances;
- Eccentricity.

In the following paragraphs the features are treated in more detail, especially as to how they are implemented. In all paragraphs the binary mole image $b(x,y)$ found by the radial search algorithm is used in which $b(x,y) = 1$ inside the mole and 0 elsewhere. The binary image has the dimensions of $N$ by $M$, which means that: $x \in [1,N]$ and $y \in [1,M]$.

6.4.1 Area, perimeter and compactness

Of all the features, the area, the perimeter and the compactness are the most basic and most easily calculated. The area ($A$) of the mole is found by counting all the
pixels in the binary image that are 1:

\[ A = \sum_{x=1}^{N} \sum_{y=1}^{M} b(x, y) \]

The perimeter (P) is found by counting the number of pixels in b(x,y) that make up the border of the mole. A mole pixel is defined as a border pixel if one or more of its 8-connected neighbours is a skin pixel. The compactness is defined and the ratio of the area and the perimeter:

\[ \text{compactness} = \frac{A}{P} \]

### 6.4.2 Asymmetry index

Asymmetry is the “A” of the ABCD rule, the more asymmetric the mole is, the more likely that it is malignant. Stoecker et al. [23] describe an algorithm that provides an objective way of measuring asymmetry in mole which, according to their tests, agrees in 93% of the cases with the dermatologists determination of asymmetry. This measure of asymmetry has been defined about the principle axes of inertia of a body. In a symmetric body these principle axes through the centre of mass of the body form the axes of symmetry. Otherwise, if the body is not symmetrical, the principle axes form the best axes of symmetry in a nearly symmetrical body. The asymmetry index is then defined as the difference in area on both sides of the best axis of symmetry divided by the total area of the mole. The derivation of the principle axes and the centre of mass can be found in Appendix 1, here only the results are mentioned.

The principle axes that form the axes of symmetry pass through the centre of mass of the mole. The coordinates if it are defined by:

\[ x_{com} = \frac{1}{A} \cdot \sum_{x=1}^{N} \sum_{y=1}^{M} x \cdot b(x, y) \]

\[ y_{com} = \frac{1}{A} \cdot \sum_{x=1}^{N} \sum_{y=1}^{M} y \cdot b(x, y) \]

In which A is the area of the mole as defined in 6.4.1. The angle that the first principle axis makes with the positive x-axis is given by:

\[ \varphi_{p1} = \frac{1}{2} \cdot \text{atan} \left( \frac{2 \cdot A \cdot x_{com} \cdot y_{com} - I_{xy}}{I_x - I_y + A \cdot x_{com}^2 - A \cdot y_{com}^2} \right) \]

Where \( I_{xy} \) = the product of inertia of the mole, defined as:

\[ I_{xy} = \sum_{x=1}^{N} \sum_{y=1}^{M} x \cdot y \cdot b(x, y) \]

\( I_x \) = the moment of inertia about the x-axis, defined as:

\[ I_x = \sum_{x=1}^{N} \sum_{y=1}^{M} y^2 \cdot b(x, y) \]
Calculating the features

\[ I_y = \text{the moment of inertia about the y-axis, defined as:} \]
\[ I_y = \sum_{x=1}^{N} \sum_{y=1}^{M} x^2 \cdot b(x, y) \]

\[ A = \text{the area of the mole.} \]

The second axis of inertia that forms a symmetry axis is \( a \), by 90° rotated, version of the first axis that also passes through the centre of mass. In Figure 6.6 and 6.7 both axes are shown.

To calculate the asymmetry index, the mole in the binary image is first translated to make the centre of mass coincide with the origin of the image, then the image is rotated over \(-\phi_{p1}\) degrees to align the principle axes with the x- and y-axis. Finally the area difference about the x-axis (Figure 6.8a) and the area difference about the y-axis (Figure 6.8b) are calculated.

The minimum of the absolute value of these area differences

\[ \Delta A_{\text{min}} = \min(|A_{1a} - A_{2a}|, |A_{1b} - A_{2b}|) \]

is divided by the total area of the mole, to give the asymmetry index:

\[ \text{asymmetry index} = \frac{\Delta A_{\text{min}}}{A} \cdot 100\% \]
6.4.3 Polar distances

Using the polar distances, some features, that describe the mole border, can be calculated. The polar distance $d_i$ is defined as the distance between the centre of mass and one of the N border pixels of the mole.

The average polar distance

$$AvgPol = \frac{1}{N} \sum_{i=1}^{N} d_i$$

is a measure for the size of the mole. The variance of the polar distances

$$VarPol = \frac{1}{N-1} \sum_{i=1}^{N} (d_i - AvgPol)^2$$

is according to [9] a very discriminative feature, indicating the irregularity of the mole border.

Using the minimum and the maximum polar distances another feature can be obtained, namely the eccentricity:

$$Eccentricity = \frac{Max(d_i)}{Min(d_i)}$$

6.5 Resolution versus feature accuracy

The resolution of the reference and match image is a very important factor when calculating feature values. If moles are represented with too few pixels then a small number of pixels difference between a reference mole and the corresponding match mole, e.g. caused by the inaccuracy of the segmentation process, might result in an enormous difference in feature values. In the current setup of the SCDS a 2000 dpi scanner is used to scan the 36x24mm$^2$ dia slides. This means that for an image taken of the back of the upper trunk of a patient the resolution is about 3 pixels/mm (the length of the marker, which is 120 mm, in such an image is represented by a length of about 300 pixel). So a mole of 6mm, which should get special attention according to the D in the ABCD-rule, fits in a window of 18x18 pixels. If one looks at the area as feature of such a mole then it has a value of

$$A = \frac{1}{4} \cdot \pi \cdot D^2 = \frac{1}{4} \cdot \pi \cdot 18^2 = 255 \text{ pixels}$$

Assumed that the radial search algorithm finds the border with an accuracy of ±1 pixel, which is quite accurate if one considers that the images could have taken under slightly difference lighting circumstances, then the diameter is determined with an accuracy of ±2 pixels. This means that the maximum error in the area is about

$$dA = \frac{1}{4} \cdot \pi \cdot (20^2 - 18^2) = 60 \text{ pixels},$$

which is almost 25% of the total area of the mole.

A mole with a diameter of 6mm is regarded as a large one, for a smaller mole the error is even larger. It is save to say that the current scanning resolution of 2000 dpi is not high enough to calculate the feature values accurate enough for detecting changes in a mole.

If one wants to calculate e.g. the feature area A of a mole with diameter D to an accuracy of $\Delta A/A$ percent, what should then be the resolution of the scanner?
The relative error in the area is:
\[
\frac{\Delta A}{A} = 2 \cdot \frac{\Delta D}{D}
\]
Hence the absolute error in the diameter is:
\[
\Delta D = \frac{1}{2} \cdot \frac{\Delta A}{A} \cdot D \quad [\text{mm}]
\]
If the radial search algorithm determines the border of the mole with an accuracy of \( n \) pixels then the diameter \( D \) is determined with an accuracy of \( \pm 2n \) pixels. So \( \Delta D \text{ mm} \) must correspond to \( 2n \) pixels to get the desired accuracy. This results in a resolution \( R \) of:
\[
R = \frac{2n}{\Delta D} = \frac{4n}{\frac{\Delta A}{A} \cdot D} \quad [\text{pixel/mm}]
\]
If one wants to calculate the area at an accuracy of 10% of a mole that has a diameter of 3mm and the accuracy of the radial search algorithm is the same as before \( (n=1) \) then the resolution of the scanner should be:
\[
R = 13.3 \quad [\text{pixel/mm}]
\]
Which means that the resolution of the scanner should be more than 4 times of what it is now.

6.6 Summary

To be able to compare corresponding moles in the reference and match image, features of these mole are calculated. A number of the features introduced in this chapter are implemented in the Khoros program DetectFeatures. The features that were chosen, were those that were tailor-made to discriminate between benign and malignant moles on the basis of just one image. The idea behind the choice of the features is that when a mole becomes malignant then especially these features will change dramatically. The calculated features of the reference and the match moles are written respectively to the Reference Feature File and the Match Feature File. The comparison process of the next chapter will decide on the basis of these feature files whether or not corresponding reference and match moles have changed. In the absence of a segmentation process to determine which pixels belong to the mole and which do not, a radial search technique was introduced to find the mole border. The resolution of the reference and match image play a very important role in the accurate calculation of the features. It was shown that the resolution of scanner currently used in the SCDS (2000 dpi) is not high enough.
Chapter 7  Detect and visualize changes

In this chapter the operation of the comparison and the visualization process is discussed. The comparison process compares corresponding moles to detect changes. If changes are noted then they are reported to the visualization process that indicates in a picture which moles are new and which ones have changed.
7.1 Indicating suspicious moles

At this point the features of the moles in the reference and match image have been calculated and the registration process has pointed out which moles in the two successive images represent the same mole and which moles are newly appeared ones. So all data is available to continue with the last stage of the SCDS which is checking whether corresponding moles in the reference and match image have changed and providing an interface to the physician on which he/she can see which mole of then have changed or are new. Checking for change is done by the comparison process and the interface to the physician is produced by the visualization process that indicates the suspicious moles on the so-called attention image.

7.2 Comparison

The comparison process is performed by the Khoros program Compare. The registration process reports the comparison process, by means of the Found Mole Pair File which moles in the reference and match image are the same moles. For every mole pair in the Found Mole Pair File the comparison process looks up the features in the feature files. These features of the reference moles the Compare program finds in the Reference Feature File, the features of the match moles are in the Match Feature File. The reference and match mole of each mole pair are compared to each other feature by feature. For each feature the relative amount of change is calculated:

\[
\text{Relative change} = \frac{|\text{Reference feature value} - \text{Match feature value}|}{\text{Reference feature value}}
\]

If the relative change of the feature value is larger than a specific threshold, that belongs to this feature, then this feature is marked as changed. The mole itself is defined as changed if the number of changed features exceeds a user defined number, called maximum number of feature violations. All the information gathered in the comparison process (old feature values, new features values, relative change, feature changed or not, mole changed or not etc.) is written to the mole difference file. The visualization process uses this file to determine which moles it should indicate as changed.

7.3 Visualization

The very last step in the SCDS is the visualization process. It is the interface to the physician which indicates at which moles the physician must have a closer look. The Khoros program Visualize represents the visualization process in the SCDS. It determines from the mole difference file and the unmapped file which moles have changed or are new. Using the positional information of the moles in the Mole Pattern File it marks the new and changed moles, in the reference and match image and displays the images on the screen. Marking is done by drawing a cross of a certain colour (depending on whether it is a new or a changed) through the suspicious moles.

Figure 7.1 shows two original images, the left one is the reference image and the right one the match image. Among the images of the patients that were available at the moment of this study, no cases of changed moles were found. In order to show the results of the visualization process, the images in Figure 7.1 have been tampered with. One of the moles in the match image is replaced by a bigger one to simulate a changed mole, another mole was added in the match image to imitate a new one. In the reference image, a mole was diagnosed as malignant and was
excised before the follow up photo session took place. Of these images two marked images were made in which only the largest moles were marked as described in Chapter 3. The reason that only the largest moles were taken is because the resolution of the scanned images is not high enough to calculate the features of small moles accurately enough, see Chapter 6. For this reason also the thresholds, that decide if a feature has changed or not, were set quite high (30%).

![Figure 7.1](image1.png)

*Figure 7.1* Original reference and match image. Since the image record does not contain images which show moles that have changed, the changed moles in these images are simulated by replacing mole 2 with a larger version in the match image. Mole number 1 is a mole that was excised before the follow up session and mole 3 is also a fake, if was added to simulate the appearance of a new mole.

These images are processed by the SCDS which produces the attention images of Figure 7.2. The left image is again the reference image and the right one is the match image. In both images the changed mole is indicated by a white cross. The black cross in the reference image marks the excised mole and the black cross in the match image shows the location of the new mole.

![Figure 7.2](image2.png)

*Figure 7.2* Images produced by the visualization proces. The moles indicated by the white crosses have changed over the period between the successive photo sessions. The black crosses indicate new moles or previously excised ones.
7.4 Summary

This chapter describes how the features of corresponding reference and match moles are compared to screen for changes. And it shows how the visualization process indicates the changed and newly appeared moles to the physician by drawing a cross through these moles on the reference and match images.
Chapter 8  Implementation in Khoros

This chapter describes the implementation of the algorithms which were described in the previous chapters. All the programs are implemented in the Khoros Scientific Software Development Environment.
8.1 SCDS implemented in Khoros

The SCDS is implemented in Khoros, which is a Scientific Software Development Environment with emphasis on information processing. The goal of Khoros is to provide software engineers a complete application development environment. It supports as well (visual) interfaces for end users as powerful tools for fast prototype programming. To aid the programmer, it provides a large number of programs for data management and arithmetics, data visualisation, information processing and visual programming. In Appendix 2 a more detailed description of Khoros is given. The following paragraphs will first introduce the individual programs that are developed to implement the SCDS and to analyse the algorithms that are implemented for the registration process, then the setup is shown as to how the registration process is tested and what Matlab scripts were used to do this, and finally the implementation of the complete SCSD is shown. The source code of all the implemented Khoros programs and Matlab scripts are printed in the accompanying book “The Source Code of the Skin Cancer Detection System”.

8.2 The individual programs

The programs developed under Khoros can be used by means of the Command Line User Interface (CLUI) as a normal Unix command or by means of a Graphical User Interface (GUI) in the visual programming environment called Cantata. In Cantata the programs are visually represented by a pictures called a Glyphs. The communication between the programs or the flow of the data is handled by files. The formats of these files are described in Appendix 3. When using the CLUI the files to which results are written or from which data is read are specified as a command line parameter just as the rest of the parameters. In Cantata the flow of the data is indicated by interconnecting the individual programs. The parameters of the programs are entered and adjusted by means of the GUI.

In total, there are 13 programs developed under Khoros to implement the SCSD and to test the programs that form the registration process. Together these programs form the SCDS-toolbox. In this toolbox the programs are divided into 5 different categories, namely:

- Segmentation:
  - Extract
  - DetectFeatures
- InitialPoints:
  - InitPntArt
  - InitPntOwn
- PointRegistration:
  - PntRegistrationArt1
  - PntRegistrationArt2
  - PntRegistrationArt3
  - PntRegistrationOwn
- Statistics:
  - Check
  - Randomize
  - Stat
- VisualizeChanges:
  - Compare
  - Visualize

The following paragraphs will only shortly describe the categories and the programs that belong to it. A complete description of all the programs in these categories, including their parameters, can be found in Appendix 4, where the man-pages of these programs are printed.
8.2.1 Category: Segmentation

The programs Extract and DetectFeatures form the segmentation process of Figure 2.2 in Chapter 2. The Extract program uses the marked image pairs to find the mole positions and the location of the markers. As input it takes a text-file in which all the image pairs are mentioned that are to be processed by the SCDS. The resulting mole and marker positions are written to the Mole Pattern File. Due to the fact that the moles in the marked images are labelled in a certain way (see Chapter 3) the program knows which moles in the reference image represent the same moles in the match image. These so-called mole pairs are written to the Correct Mole Pair File, which is used by the programs of the Statistics category to check if the programs that make up the registration process (category InitialPoints and PointRegistration) work correctly.

The DetectFeatures program uses this Mole Pattern File to locate the moles in the original image, in which it first locates all the pixels that are part of a certain mole (using the radial search algorithm) after that it calculates a number of mole features that are written to the Reference and Match Feature File for respectively the features of the reference and the match moles. These features are used by the programs of the VisualizeChanges category to indicate which moles have changed.

8.2.2 Category: InitialPoints

This category consists of the programs InitPntArt and InitPntOwn, which select the most likely initial mole pairs for the point registration programs of the category PointRegistration. The programs read from the Mole Pattern File the positions of the moles in the reference and match image and select on the basis of this combinations of reference and match moles that are most likely to represent the same mole. The selected mole combinations, or mole pairs, are written to the Initial Mole Pair File. The InitPntArt program makes use of the Gabriel graph algorithm while InitPntOwn is the implementation of the local node algorithm. Both algorithms are described in Chapter 5.

8.2.3 Category: PointRegistration

This category, which consists of the Khoros programs PntRegistrationArt1, PntRegistrationArt2, PntRegistrationArt3 and PntRegistrationOwn, and the InitialPoints category are the most important categories in this study of the SCDS. Together they form the registration process, which has the task of determining which moles in the match image represent the same moles in the reference image. The SCDS needs this information to be able to check whether or not moles have changed during the period between taking the reference and match image. The programs of this category read the Mole Pattern File and perform on the basis this the actual registration of the reference and match moles but in order to be able to do this they need some initial matches which are supplied through the Initial Mole Pair File which is generated by the one of the programs of the InitialPoints category. It results in a number of reference-match mole combinations, called mole pairs, that are written to the Found Mole Pair File. The moles of which no corresponding mole is found in the other image are written to the Unmapped File. The first three programs of this category implement algorithms described by an article of Perednia and White [18], which need respectively one, two and three initial mole pairs. The last program mentioned uses the baseline algorithm to determine the reference-match mole pairs.
8.2.4 Category: Statistics

To statistically test the programs of the previous two categories, which make up the registration process, the programs of this category are developed. The Program Randomize is developed to generate many random image pairs on the basis of just a few original image pairs. One reason for the development of this program was that for the statistical analysis of the registration process many image pairs are needed that have to be registered by hand which is a very time consuming task. Another reason is that there were only a few image pairs available at the time of this study. For every image pair present in the Mole Pattern File it generates a specified number of random images that contain a specified number of reference and match moles. The resulting image pairs are written to the Test Mole Pattern File. Along with these random image pairs it generates the Test Correct Mole Pair File which contains the correct reference-match mole combinations which are needed to check whether or not the programs of the registration process come up with the correct mole pairs. To be able to check the programs of the PointRegistration category without using the programs of the InitialPoints category, the program also generates the Test Initial Pair File which contains the initially selected mole pairs for every image pairs in the Test Mole Pattern File. The Check program compares the mole pairs found by the programs of the registration process to the mole pairs in (Test) Correct Mole Pair File to count the number of mole pairs that were correctly found, incorrectly found or not found at all. The results are written to the Evaluation File which is processed by the Stat program. The Stat program gathers and sorts all the information in the Evaluation File to form a number of histograms from which the performance of the tested programs can be interpreted. The histograms are written to the Statistics File and can be visualised using some Matlab programs which are treated in 8.3.

8.2.5 Category: VisualizeChanges

The last category embodies the programs Compare and Visualize. The Compare program compares the reference and match moles that form mole pairs in the Found Mole Pair File. It does this on the basis of their mole features in the Reference Feature File and the Match Feature File. All the feature changes noted by the program are reported in the Mole Difference File. Those moles of which the features have changed more than a certain specified threshold are indicated in this file as “changed”. The interface to the physician is provided by the program Visualize. This program displays the original image pair and indicates in this original reference and match image which moles are new and which moles have changed. Indicating is done by drawing a cross at the position of the mole. The information about which moles are new or which ones have changed is retrieved from respectively the Unmapped File and the Mole Difference File. The locations of the changed and new moles it finds in the Mole Pattern File.

8.3 Testing the registration process

The statistical testing of the registration process can be divided into three groups namely 1) testing the initial mole pair selecting programs, 2) testing the registration programs and 3) testing them both together. The last test is representative for the whole registration process but Chapter 4 and 5 also require the testing of the individual programs. The next paragraphs describe the tests by
Testing the registration process

means of the Khoros visual programming environment Cantata. The tests used in
Chapter 4 and 5 however are performed in Matlab (version 4.2c), which is a
numeric computation and visualization software package. The Matlab scripts used
for this described.

8.3.1 Testing the initial mole pair selecting programs

Figure 8.1 shows how the initial mole pair selecting programs, InitPntOwn and
InitPntArt, are tested in the Khoros visual programming environment, called
Cantata. The File sequencing tool selects the Images to Extract File and passes it
on to Extract. Extract reads the marked reference and match images that are listed
in this file and determines the mole and marker positions in these images. This
results in the Mole Pattern File and the Correct Mole Pair File which serve as input
to the Randomize program that produces a randomized version of the Mole Pattern
File and the Correct Mole Pair File. As third file it produces the Test Initial Mole
Pair File which normally used to test the registration programs but in this case it
is just supplied to the Stat program since a number of histograms which the Stat
program produces need this information even though these histograms have no
meaning when testing the initial mole pair selecting programs. The randomized
mole patterns in the Test Mole Pattern File are fed to the InitPntOwn or
InitPntArt program, depending on which of the programs is being tested. These
programs produce the Initial Mole Pair File which is checked for correctness by the
program Check. Check compares the selected mole pairs to the correct mole pairs
in the Test Correct Mole Pair File and reports its findings to the Stat program via
the Evaluation File. The Stat program gathers and sorts all the results and
generates a number of histograms which it writes to the Statistics File. This ascii­
file can be viewed using the FileViewer or it can be saved and viewed using the
Matlab script SS_Histogram.m, which draws all the histograms generated by the
program Stat. Figure 5.3 of Chapter 5 is an example of such an histogram.

Figure 8.1 Testing the initial mole pair selecting programs in the Khoros visual programming
environment. In this Table the InitPntOwn program is tested.

The test described above is just one test with a certain number of moles in the
reference and match images which are specified by the parameters, set in the
program Randomize. To produce the statistics of the tested program as a function
of e.g. the number of moles in the images, the test has to be repeated many times.
For this sort of testing some Matlab scripts were written that use the CLUI
(Command Line User Interface) of the programs.

There are two scripts for this, one for each program to be tested, namely:
- PS_InitPntArt.m (Testing program InitPntArt, which is based on the Gabriel graph algorithm);
- PS_InitPntOwn.m (Testing program InitPntOwn, which is based on the local node algorithm).

They measure the performance of the programs for 10, 12, ..., 60 moles in the reference and match image of which 0, 1, 2, ..., 5 reference moles are removed to simulate new match moles. This amounts to 156 tests and thus 156 Statistics Files are produced by the Stat program. The Matlab scripts

- SS_InitPntArt.m;
- SS_InitPntOwn.m;

read all these files and present the data they contain in a readable form, like e.g. Figure 5.4 in Chapter 5.

8.3.2 Testing the registration programs

The testing of the registration programs InitPntArt3 and InitPntOwn is largely the same as the testing of the initial mole pairs selecting programs of the previous paragraph. Figure 8.2 shows how it is done in Cantata.

![Figure 8.2 Testing the registration programs in the Khoros visual programming environment.](image)

The only real difference is that the registration programs need some initially selected mole pairs, which are provided via the Test Initial Mole Pair File. As in the previous paragraph these initial mole pairs are past on to the Stat program but now the histograms that use this information do have a meaning. They show the performance of the registration program as function of the distribution of the initial mole pairs.

For testing the programs under the many different circumstances, as mentioned in the last paragraph, two Matlab scripts are written:

- PS_Art3.m (Testing InitPntArt3, which is based on the registration algorithm of Perednia and White and needs 3 initial mole pairs);
- PS_Own.m (Testing InitPntOwn, which is based on the baseline algorithm).

To convert the generated Statistic Files into readable graphs, the Matlab scripts
Testing the registration process

- SS_Art3.m;
- SS_Own.m;

are written. Examples of the generated graphs are Figures 4.10, 4.11 and 4.12 in Chapter 4.
Two other scripts are provided to generate and visualize the statistical graphs of the performance of the PntRegistrationOwn program, an implementation of the baseline algorithm, when match moles are removed instead of reference moles. This script is written to prove that it does not matter to the performance whether reference moles or match moles are removed. The scripts are called:

- PS_Own_RemMatch.m (For generating the statistics);
- SS_Own_RemMatch.m (For showing the graphs of the produced statistics).

8.3.3 Testing the complete registration process.

Testing the complete registration process is basically the same as testing only the registration programs in the previous paragraph, see Figure 8.3. The only difference is that the initial mole pairs for the registration program are now generated by one of the initial mole pair selecting programs instead of being provided by the Randomize program.

Figure 8.3 Testing the complete registration process in the Khoros visual programming environment.

As explained in Chapter 5 only the PntRegistrationOwn program is tested in combination with the InitPntArt or the InitPntOwn initial mole pair selecting program. This means that only two Matlab scripts are needed for the exhaustive testing described in the last paragraphs. The script are:

- PS_InitPntArt_Own.m (The InitialPntArt program provides the initial mole pairs and the PntRegistrationOwn registers the moles);
- PS_InitPntOwn_Own.m (The InitPntOwn program provides the initial mole pairs and the PntRegistrationOwn program registers the moles).

The results of this exhaustive testing are presented in graphs which are
respectively drawn by:

- SS_InitPntArt_Own.m;
- SS_InitPntOwn_Own.m.

Figure 5.8 and 5.9 in Chapter 5 are examples of the resulting graphs.

### 8.4 The complete SCDS in Khoros

The Skin Cancer Detection System with the best performing registration process, formed by the programs InitPntArt and PntRegistrationOwn, is shown in Figure 8.4.

The Extract program receives the images, in which it must find the moles and markers, via the Images to Extract File which is provided to the Extract program by the File sequencing tool. The resulting Mole Pattern File is used by the registration process (InitPntArt and PntRegistrationOwn) to determine which moles in the reference and match image represent the same mole and which moles do not have a corresponding mole in the other image. The results are written to respectively the Found Mole Pair File and the Unmapped File.

![Figure 8.4 The complete SCDS in the Khoros visual programming environment.](image)

The program Compare reads the Found Mole Pairs File and checks if the reference and match moles, that make up these mole pairs, have changed. The reference and match moles are compared to each other on the basis of the mole features in the Reference and Match Feature File, which are both produced by the DetectFeatures program. The Compare program reports in the Mole Difference File whether or not corresponding reference and match moles have changed.

The Visualize program reads in the Mole Difference File and the Unmapped File which moles have to be indicated to the physician as a changed or new mole. In the Mole Pattern File it reads the positions of those moles and marks them on the reference and match image by drawing a cross through them. Figure 7.2 in Chapter 7 is an example of this.

### 8.5 Summary

In this chapter it is explained how the different programs constituting the SCDS are implemented in Khoros. It describes how the registration process and the individual parts of which it consists are tested using Matlab scripts and the Khoros visual programming environment, called Cantata. And finally it is shown how the SCDS is build up in Cantata.
Chapter 9  Conclusions

A short summary is given of the results reached in this study and some recommendations are made for further study.
In this study a first step is made in the development of a skin cancer detection system that detects skin cancer by screening the patient for changes in moles. The screening is performed on two images of the same body area which are taken at the first visit of the patient and a follow-up visit.

The main focus of this study is the registration of these images, determining which moles in the first image represent the same moles in the follow-up image. This results in a list of mole pairs of corresponding moles and a list of moles that do not have a corresponding mole in the other image. The moles of the first mentioned list are to be examined further by the system to see if changes in the moles have appeared. The moles in the second list are possibly new moles.

A number of registration algorithms from literature were implemented and tested on real images of patients. Of these algorithms, the one that performed the best is the 3-point algorithm of Perednia and White, it needs 3 initial mole pairs to reduce the dimensionality of the mole (point) matching problem and determines on the basis of these initial pairs the other pairs of corresponding moles in both images. When tested on real images of patients this algorithm found the correct mole pairs in about 97% of the cases but only if certain restraints were imposed on the distribution of the 3 initial mole pairs.

A new registration algorithm, the baseline algorithm, was developed which does not impose any constraints on the initial mole pair distribution and needs only 2 initial mole pairs instead of three. When tested on the same images as the 3-point algorithm, it registers the images correctly in about 99% of the case. Another positive point of this algorithm, when compared to the 3-point algorithm, is that its performance does not depend much on the number of moles in the image of the number of new moles that do not have a corresponding mole in the other image.

The initial mole pairs that the registration algorithms need, can be provided by a physician or, as it is done in the study, by an algorithm that select a number of highly likely initial mole pairs. The Gabriel algorithm of Perednia and White was implement and tested.

When tested on real images it selects 3 initial mole pairs correctly in more that 99.2% of the cases. Next to this another algorithm, the local node algorithm, was developed that performs less than the Gabriel graph algorithm, 3 correctly selected initial mole pairs in more than 98.5% of the cases, but is faster. It only needs $O(N^2)$ steps instead of the $O(N^3)$ for the Gabriel graph algorithm.

When the combined performance of initial mole pair selecting algorithm and registration algorithm was tested, it appeared that the combination of the baseline algorithm, for registration, and the Gabriel algorithm, for selecting the initial mole pairs, performed best. In 98-99% of the cases this combination finds the correct mole pairs when tested on real images.

After registering the images, the moles of both images that are registered as the same mole are compared on the basis of mole features that are specially tailored to diagnose malignant moles. It was shown that the scanning resolution of the images at the time of the study was not high enough to calculate the mole features accurate enough to detect changes in corresponding moles. Next to this resolution problem the number of implemented features should be explained to describe also the colour and texture of the moles. The mole features implemented at the moment nearly all describe some aspect of the shape of the mole.

Finally the skin cancer detection system must be provided with an appropriate segmentation process that determines automatically where the moles are located in the image and which pixels belong to these moles. At the moment the positions of the moles are indicated by marking the moles in a drawing program which is an
enormously labour intensive task. Without a segmentation process the whole skin cancer detection system is probably of less value to a physician.
References

[1] Dhawan, A.P.
A review on biomedical image processing and future trends.

[2] Stoecker W.V.; Moss, R.H.
Editorial: Digital imaging in dermatology.

Early detection of malignant melanoma: The role of physician examination and self examination of skin.

[4] Nilles, M.; Boedeker, R.H.; Schill, W.B.
Surface microscopy of naevi and melanomas: clues to melanoma.

[5] Umbaugh, S.E.; Moss, R.H.; Stoecker W.V.
Automatic color segmentation of images with application to detection of variegated coloring in skin tumors.

[6] Ercal, F.; Chawla, A.; Stoecker, W.V.; Hsi-Chieh Lee; Moss, R.H.
Neural network diagnosis of malignant melanoma from color images.

Shape analysis for classification of malignant melanoma

Classification of Melanocytic lesions with color and texture analysis using digital image processing.
Analytical and Quantitative Cytology and Histology, Vol. 15 (Feb. 1993), No. 1.
Initial results of automated melanoma recognition

[10] Perednia, D.A.; White, R.G.; Schowengerdt, R.A.
Localization of cutaneous lesions in digital images.

Boundary detection for skin lesions: an edge focusing algorithm.

[12] Ercal, F.; Moganti, M.; Stoecker, W.V.; Moss, R.H.
Detection of skin tumor boundaries in color images.

[13] Golston, J.E.; Moss, R.H.; Stoecker, W.V.
Boundary detection in skin tumor images: an overall approach and a radial search algorithm.
Pattern Recognition, Vol. 23 (1990), No. 11, p. 1235-1247.

[14] Hall, P.N.; Claridge, E.; Morris Smith, J.D.
Computer screening for early detection of melanoma - is there a future?

[15] Umbaugh, S.E.; Hance, G.A.; Moss, R.H.; Stoecker, W.V.
Unsupervised color image segmentation: with application to skin tumor borders.

[16] Umbaugh, S.E.; Moss, R.H.; Stoecker, W.V.; Hance, G.A.
Automatic color segmentation algorithms with application to skin tumor feature identification.

[17] Umbaugh, S.E.; Moss, R.H.; Stoecker, W.V.
An automatic color segmentation algorithm with application to identification of skin tumor borders.

[18] Perednia, D.A.; White R.G.
Automatic registration of multiple skin lesions by use of point pattern matching.
Automatic derivation of initial match points for paired digital images of skin.

[20] Skolnick, M.M.
An approach to completely automatic comparison of two-dimensional electrophoresis gels.

[21] Skolnick, M.M.
Application of morphological transformations to the analysis of two-dimensional electrophoresis gels of biological material.

Early detection of malignant melanoma: The role of physician examination and self-examination of the skin.

[23] Stoecker, W.V.; Li, W.W.; Moss, R.H.
Automatic detection of asymmetry in skin tumors.

[24] J.D. Foley; A. van Dam
Fundamentals of Interactive Computer Graphics

Appendix 1  Principle Axes

In this appendix the formulas for the determination of the principle axes of inertia are derived. First the formulas for a physical two-dimensional body are derived, which are then translated to be used for the binary mole image.
A1.1 Physical description of the principle axes of inertia

In the general case of the motion in a plane of a set of particles with respect to a single fixed point, the angular momentum can be written as having two components with respect to the origin of a coordinate system:

\[ H_x = \omega_x \cdot \sum m_i \cdot y_i^2 - \omega_y \cdot \sum m_i \cdot x_i \cdot y_i = \omega_x \cdot I_x - \omega_y \cdot I_{xy} \]

\[ H_y = \omega_y \cdot \sum m_i \cdot x_i^2 - \omega_x \cdot \sum m_i \cdot x_i \cdot y_i = \omega_y \cdot I_y - \omega_x \cdot I_{xy} \]

Where \( \omega_x, \omega_y \) are the components of the angular velocity and \( m_i \) is the mass of the \( i \)th particle at the coordinates \( (x_i, y_i) \). The terms \( I_x \) and \( I_y \) are called the moments of inertia with respect to the x and y axis. The term \( I_{xy} \) is called the product of inertia.

There always exists a set of axes such that the product of inertia vanishes and the angular momentum can be expressed in terms of moments of inertia alone. Such axes are called the principle axes of inertia. For a point of a body the moments of inertia about the principle axes are stationary.

The centre of mass in a body is a point in that body about which the sum of the moments of all individual masses, of which the body consists, is zero. The coordinates of the centre of mass can be expressed as:

\[ x_{com} = \frac{1}{m_t} \cdot \sum x_i \cdot m_i \]

\[ y_{com} = \frac{1}{m_t} \cdot \sum y_i \cdot m_i \]

Where \( m_t \) the total mass of the body is: \( m_t = \sum m_i \)

For a symmetrical body the principle axes of inertia through the centre of mass are also the axes of symmetry of the body. If a body is not symmetric, then the principle axes through the centre of mass are said to be the best axes of symmetry for a nearly symmetric body.

The principle axes of inertia form a coordinate system with respect to the orthonormal base \{\( \hat{e}_u, \hat{e}_v \)\} which a, by \( \phi \) rotated, version of the xy-coordinate system relative to base \{\( \hat{e}_x, \hat{e}_y \)\}, see Figure A1.1.

![Figure A1.1 Base of principle axes of inertia are a by \( \phi \) rotated version of the base of the xy-coordinate system.](image-url)
Hence:

\[ \begin{align*}
\dot{\mathbf{e}}_u &= R(\phi) \cdot \dot{\mathbf{e}}_x \\
\dot{\mathbf{e}}_v &= R(\phi) \cdot \dot{\mathbf{e}}_y
\end{align*} \]

Where \( R(\phi) \) is the rotation matrix:

\[
R(\phi) = \begin{bmatrix}
\cos \phi & \sin \phi \\
\sin \phi & -\cos \phi
\end{bmatrix}
\]

The relation between the coordinates \((u,v)\) relative to base \{\(\mathbf{e}_u, \mathbf{e}_v\}\) and the coordinates \((x,y)\) relative to base \{\(\mathbf{e}_x, \mathbf{e}_y\)\} is:

\[ u = x \cdot \cos \phi + y \cdot \sin \phi \]
\[ v = x \cdot \sin \phi - y \cdot \cos \phi \]

Using the fact that the product of inertia is zero in the \(uv\)-system, it is possible to calculate \(\phi\):

\[
I_{uv} = \sum_i m_i \cdot u_i \cdot v_i = 0
\]
\[
\sum_i m_i \cdot (x_i^2 \cdot \sin \phi \cdot \cos \phi - x_i \cdot y_i \cdot \cos^2 \phi + x_i \cdot y_i \cdot \sin^2 \phi - y_i^2 \cdot \sin \phi \cdot \cos \phi) = 0
\]
\[
(I_y - I_x) \cdot \frac{1}{2} \cdot \sin 2\phi - I_{xy} \cdot \cos 2\phi = 0
\]
\[
\tan 2\phi = \frac{-2 \cdot I_{xy}}{I_x - I_y}
\]

Thus the principle axes of body in the \(xy\)-coordinate system are at an angle \(\phi\) with respect of the positive part of the \(x\)-axis, with:

\[
\phi = \frac{1}{2} \cdot \arctan \left(\frac{-2 \cdot I_{xy}}{I_x - I_y}\right) + n \cdot \frac{\pi}{2} \quad n \in \mathbb{Z}
\]

To place the origin of the coordinate system formed by the principle axes at the centre of mass, the \(xy\)-system is translated to form a new coordinate system \(x'y'\):

\[ x' = x - x_{\text{com}} \]
\[ y' = y - y_{\text{com}} \]

To obtain the angle that the principle axis in the \(x'y'\)-system makes with the positive \(x'\)-axis, the moments and products of inertia, respectively \(I_{x'}, I_{y'}\) and \(I_{x'y'}\), must be calculated. First the product of inertia:

\[
I_{xy} = \sum_i m_i \cdot x_i \cdot y_i \Leftrightarrow
\]
\[
I_{xy} = \sum_i m_i \cdot x_i' \cdot y_i' + \sum_i m_i \cdot x_{\text{com}} \cdot y_{\text{com}} + \sum_i m_i \cdot x_i' \cdot y_{\text{com}} + \sum_i m_i \cdot x_{\text{com}} \cdot y_i' \Leftrightarrow
\]
\[
I_{xy} = I_{x'y'} + m_i \cdot x_{\text{com}} \cdot y_{\text{com}} + m_i \cdot x_{\text{com}}' \cdot y_{\text{com}} + m_i \cdot x_{\text{com}} \cdot y_{\text{com}}' \]
\[ x_{\text{com}}' = 0 \]
\[ y_{\text{com}}' = 0 \]
\[
I_{x'y'} = I_{xy} - (m_i \cdot x_{\text{com}} \cdot y_{\text{com}})
\]
For the moment of inertia in the x'y' plane follows:

\[ I_y = \sum_i m_i \cdot x_i^2 \]

\[ I_y = \sum_i m_i \cdot x_i^2 + 2 \cdot \sum_i m_i \cdot x'_i \cdot x_c + \sum_i m_i \cdot x_{com}^2 \]

\[ I_y = I'_y + m_t \cdot x_{com}^2 \]

\[ I_y = I'_y - m_t \cdot x_{com}^2 \]

And \( I_x \) follows in the same way:

\[ I_x = I'_x - m_t \cdot y_{com}^2 \]

The angles that the principle axes make with the x' axis in the x'y' plane is now:

\[ \varphi = \frac{1}{2} \cdot \tan \left( \frac{-2 \cdot I_{xy}}{I'_x - I'_y} \right) + n \cdot \frac{\pi}{2} \quad n \in \mathbb{Z} \]

\[ \varphi = \frac{1}{2} \cdot \tan \left( \frac{-2 \cdot \frac{I_{xy} - m_t \cdot x_{com} \cdot y_{com}}{I'_x - I'_y + m_t \cdot x_{com}^2 - m_t \cdot y_{com}^2}} \right) + n \cdot \frac{\pi}{2} \quad n \in \mathbb{Z} \]

Because the x'y'-system is just a translated version of the xy-system it follows that the principle axes through the centre of mass make the same angle with the positive x-axis of the xy-system.

### A1.2 Converting formulas to the binary mole image

Until now the definitions for calculating the principle axes were based on physical particles of a certain mass that all together constituted a planar body. The step to adapting the formulas to a binary image is fairly simple. In the binary image the mole is represented by the function \( b(x,y) \) where \( b(x,y) = 1 \) inside the mole and \( b(x,y) = 0 \) outside. The pixels of the image can now be seen as having the mass \( b(x,y) \).

The coordinates of the centre of mass are then:

\[ x_{com} = \frac{1}{A} \cdot \sum_x \sum_y x \cdot b(x,y) \]

\[ y_{com} = \frac{1}{A} \cdot \sum_x \sum_y y \cdot b(x,y) \]

Where \( A \) is the area of the mole:

\[ A = \sum_x \sum_y b(x, y) \]

The product and moments of inertia become:

\[ I_{xy} = \sum_x \sum_y x \cdot y \cdot b(x, y) \]

\[ I_x = \sum_x \sum_y y^2 \cdot b(x, y) \]
$$I_y = \sum_{x} \sum_{y} x^2 \cdot b(x, y)$$

The angles of the principle axes of inertia, that are the best axes of symmetry at the same time, with respect to the positive x-axis are:

$$\varphi = \frac{1}{2} \cdot \text{atan}
\left(-2 \cdot \frac{I_{xy} - A \cdot x_{com} \cdot y_{com}}{I_x - I_y + A \cdot x_{com}^2 - A \cdot y_{com}^2}\right) + n \cdot \frac{\pi}{2} \quad \text{n } \in \mathbb{Z}$$
Appendix 2   The Khoros system
This appendix explains what the Khoros system is and why it is used.
A2.1 What is the Khoros system?

Khoros is a Scientific Software Development Environment with the emphasis on information processing. The goal is to provide to software engineers a complete application development environment. It supports as well (visual) interfaces for end users as powerful tools for fast prototype programming. To aid the programmer, it provides a large number of programs for data management and arithmetics, data visualization, information processing and visual programming.

A2.2 What does Khoros consist of?

For application developers, the Khoros Toolbox Programmer's system consists of programming services and software development tools that support all aspects of developing new engineering and scientific applications.

A2.2.1 Programming services

It is the goal of Khoros to let the developer concentrate on the algorithms rather than on the programming of data storage, representation or visualization. A number of basic programming services are provided to make this easier for the programmer. It consists out of foundations services, data service and visualization services.

Foundation services

This are the underlying support services for the programs. They are a set of services that do not involve data processing, representation or visualization. It consists out of:

- general utilities (message parsing, list, memory allocation, ...);
- mathematical utilities (complex, double and matrix arithmetics, ...);
- mathematical expression parser;
- operating system services (data transport, distributed computing, ...);
- software services (toolbox management, configuration, ...);
- user interface services (CLUI and GUI, ...).

Data Services

Data services comprises a powerful system for accessing and manipulating data. The object of data services is to provide the application programmer with the ability to access and operate on data independent of its physical characteristics such as size or data type. Data services are designed to address the needs of a large number of application domains, from image and signal processing to geometry visualization and numerical analysis.

GUI and visualization services

These services provide an interface to the visual representation of data. It is a layer above the used X-Windows library. It provides all the capabilities of X-Windows but shield the programmer from X-Windows itself.

A2.2.2 Toolboxes

With these programming services the programmer can develop toolboxes. Toolboxes are a collection of programs that operate on data of a certain form, or offer the needed operations in a domain of applications.

1. CLUI: Command Line User Interface.
2. GUI: Graphical User Interface.
A number of toolboxes are standard provided with Khoros. They allow among others, operations on geometric and general polymorphic data, matrices, and images. The standard toolboxes also provide toolboxes for toolbox management, toolbox design and programming aids. It is possible for users to develop and distribute their own toolboxes for specific or general applications. To facilitate the process of toolbox programming a number of tools are provided.

**A2.2.3 Programming tools**

A number of programming tools are included in Khoros. They make it easier for the programmer to develop their own toolboxes, to use and modify existing ones or to program GUI’s and documentation for end users. Some of these programming tools are:

- **Craftsman**: to manage the toolboxes;
- **Composer**: to create and modify individual software objects;
- **Guise**: to create GUI’s and CLUI’s for each program;
- **Ghostwriter** and **Conductor**: to automatically create large parts of the source codes, e.g. standard main-format and parts of the documentation;
- **Cantata**: a visual data flow representation program.

**A2.3 Cantata**

Cantata is one of the most important programs of the Khoros system. It is presented as a Visual Programming Environment in Khoros and allows a visual representation of a data flow in the developed system. A special high-level language has been developed to program in Cantata.

**A2.3.1 Out of what does Cantata consist?**

The Khoros system and toolboxes consist out of a large number of building blocks, or programs. **Glyphs** are the visual representation of these programs from the different toolboxes. Every Glyph accepts its own parameters from a GUI and the output of one glyph can be connected to the input of the next Glyph. By interconnecting the different Glyphs, and specifying the parameters in the GUI, it is easy to visually program a total system. The program allows visual programming of while-loops, for-next and if-then-else structures. Together they make up the Cantata workspace.

An example of a workspace with some Glyphs and a GUI is given in Figure A2.1

**A2.3.2 Why use Cantata?**

The Cantata workspace offers a number of advantages:

**It is visual and easy to use**
All the needed functionality (which is also offered by the CLUI of the program) is readily available to the user by means of the GUI. The different parameters are easier to interpret and adjust when compared to the CLUI.

**It allows fast prototyping**
The building blocks can be connected to each other, parameters can be changed easily and new blocks can be inserted. The standard toolboxes and the ones of other contributors offer a wide range of functionality that can be reused and does not have to be programmed again.
It allows data flow observation
It is possible to redirect the current data flow to an analysis tool to see how the data changes throughout the flow, which parameters have to be adjusted and in what way.

Figure A2.1 The Cantata workspace is depicted on which a number of interconnected Glyphs are shown. The interconnections represent the data flow between the programs. The GUI shows the parameters of the DetectFeatures program.

A2.4 More information and licensing
Khoros is distributed through the Internet as free access software. It is available throughout the world free of charge. It is however not public domain. The software is owned by Khoral Research Inc. (KRI) and carries a Licence and Copyrights. It can be used by any organization free of charge, but it can not be distributed without licence.

More information about Khoros, licensing or questions can be obtained from KRI:

Khoral Research Inc.
6001 Indian School Rd., NE
Suite 200
Albuquerque, NM 87110

or by email: khoros-request@khoros.unm.edu
newsgroup: comp.soft-sys.khoros
Appendix 3  File formats

This appendix describes the structure of the files that are used to carry information from one Khoros program to the other.
A3.1 Format of Images to Extract File

This file contains the names and pathnames of all the reference images and the corresponding match images that the SCDS will process in one run.

<filename of reference image 1>
<filename of match image 1>
<filename of reference image 2>
<filename of match image 2>
.
.
e tc for the other filenames

A special convention is used to name the reference and match images, see the help page of the Extract program in Appendix 4.1.

A3.2 Format of the Mole Pattern File

The mole pattern file contains all the mole and marker positions of all reference/match image pairs. The format is:

<nr of image pairs>
<filename of ref image>
<dpi image>
<length marker in mm>
<x1 corner marker> <y1 corner marker>
<x2 corner marker> <y2 corner marker>
<x3 corner marker> <y3 corner marker>
<x4 corner marker> <y4 corner marker>
<nr of points in ref image 1>
<xpos point 0> <ypos point 0>
<xpos point 1> <ypos point 1>
.
.<xpos point i> <ypos point i>
<filename of match image>
<dpi image>
<length marker in mm>
<x1 corner marker> <y1 corner marker>
<x2 corner marker> <y2 corner marker>
<x3 corner marker> <y3 corner marker>
<x4 corner marker> <y4 corner marker>
<nr of points in match image 1>
<xpos point 0> <ypos point 0>
<xpos point 1> <ypos point 1>
.
.<xpos point j> <ypos point j>
.
etc for other image pairs

(x1,y1)-(x2,y2) and
(x3,y3)-(x4,y4) are the long sides of the marker

Same structure for the other image pairs
A3.3 Format of the Unmapped File

The unmapped mole file contains all the mole for which the registration process could not find a corresponding mole in the other image. The format is:

\[
\text{<nr of image pairs> } \\
\text{<filename of ref image> } \\
\text{<nr of unmapped ref moles> } \\
\text{<ref mole nr> } \\
\text{<ref mole nr> } \\
\text{.} \\
\text{.} \\
\text{.} \\
\text{<ref mole nr> } \\
\text{<nr of unmapped match moles> } \\
\text{<match mole nr> } \\
\text{<match mole nr> } \\
\text{.} \\
\text{.} \\
\text{.} \\
\text{<match mole nr> } \\
\text{.} \\
\text{.} \\
\text{etc for other image pairs}
\]

A3.4 Format of the Mole Pair File

The mole pair file contains the mole pair, i.e. the moles that represent the same moles in successive images. There are three of these files:

- (Test) Correct Mole Pair File: contains the correct mole pair given by Extract.
- (Test) Initial Mole Pair File: contains the mole pairs selected by the initial match program.
- (Test) Found Mole Pair File: contains the mole pairs found by the registration program.

The format is:

\[
\text{<nr of image pairs> } \\
\text{<filename of ref image> } \\
\text{<filename of match image> } \\
\text{<nr of mole pairs> } \\
\text{<ref mole nr> <match mole nr> } \\
\text{<ref mole nr> <match mole nr> } \\
\text{.} \\
\text{.} \\
\text{.} \\
\text{<ref mole nr> <match mole nr> } \\
\text{.} \\
\text{.} \\
\text{etc for other image pairs}
\]
A3.5 Format of the Graph File

This file contains the structure of the Gabriel graph produced by the initial match point algorithm of paragraph 5.2. The format is:

<nr of image pairs>
<filename of ref image>
<nr of ref vertices (moles)(n)>
<vertex nr>
<nr of edged this vertex has>
<vertex nr to which it has an edge>
<vertex nr to which it has an edge>
...
<vertex nr to which it has an edge>
<vertex nr>
<nr of edged this vertex has>
...
etc for other ref vertices
<filename of match image>
<nr of match vertices (moles)(m)>
<vertex nr>
<nr of edged this vertex has>
<vertex nr to which it has an edge>
<vertex nr to which it has an edge>
...
<vertex nr to which it has an edge>
<vertex nr>
<nr of edged this vertex has>
...
etc for other match vertices
<filename of ref image>
<nr of ref vertices (moles)>
...
etc for other image pairs
A3.6 Format of the Evaluation File

The evaluation file is produced by the Khoros program Check, it contains information about which mole pair were found correctly, incorrectly or not found at all.
The format is:

```
<nr of image pairs>
<filename of ref image>
<filename of match image>
<nr correct pairs>
<nr correctly found pairs in row>
<nr correctly found pairs>
<ref mole nr> <match mole nr>
<ref mole nr> <match mole nr>
.
<ref mole nr> <match mole nr>
<nr wrongly found pairs>
<ref mole nr> <match mole nr>
.
<ref mole nr> <match mole nr>
<nr not found pairs>
<ref mole nr> <match mole nr>
.
<ref mole nr> <match mole nr>
<filename of ref image>
<filename of match image>
<nr correct pairs>
.
.
```

Same structure for other image pairs

etc for other image pairs

A3.7 Format of the Statistics File

The program Stat produces this file that contains the statistical information of the images processed by the registration process.
The format is:
<table>
<thead>
<tr>
<th>Metric</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;nr of image pairs&gt;</td>
<td>Number of image pairs</td>
</tr>
<tr>
<td>&lt;avg correct pairs&gt;</td>
<td>Average correct pairs</td>
</tr>
<tr>
<td>&lt;std correct pairs&gt;</td>
<td>Standard deviation correct pairs</td>
</tr>
<tr>
<td>&lt;avg correct found pairs in row&gt;</td>
<td>Average correct pairs found in a row</td>
</tr>
<tr>
<td>&lt;std correct found pairs in row&gt;</td>
<td>Standard deviation correct pairs found in a row</td>
</tr>
<tr>
<td>&lt;nr histogram bins&gt;</td>
<td>Number of histogram bins</td>
</tr>
<tr>
<td>&lt;min bin value&gt;</td>
<td>Minimum bin value</td>
</tr>
<tr>
<td>&lt;max bin value&gt;</td>
<td>Maximum bin value</td>
</tr>
<tr>
<td>&lt;nr bin items&gt;</td>
<td>Number of bin items</td>
</tr>
<tr>
<td>&lt;avg bin&gt;</td>
<td>Average bin value</td>
</tr>
<tr>
<td>&lt;std bin&gt;</td>
<td>Standard deviation bin</td>
</tr>
<tr>
<td>Histogram of nr correct mole pairs</td>
<td>Histogram of number of correct mole pairs as function of the distance</td>
</tr>
<tr>
<td>Histogram of correct found pairs</td>
<td>Histogram of correct found pairs as function of the distance</td>
</tr>
<tr>
<td>Histogram of correct found pairs</td>
<td>Histogram of correct found pairs as function of the min angle</td>
</tr>
<tr>
<td>Histogram of incorrect found pairs</td>
<td>Histogram of incorrect found pairs as function of the distance</td>
</tr>
<tr>
<td>Histogram of incorrect found pairs</td>
<td>Histogram of incorrect found pairs as function of the min angle</td>
</tr>
<tr>
<td>Histogram of missed pairs</td>
<td>Histogram of missed pairs as function of the distance</td>
</tr>
<tr>
<td>Histogram of missed pairs</td>
<td>Histogram of missed pairs as function of the min angle</td>
</tr>
</tbody>
</table>

**Note:** The table above represents various statistical metrics that are commonly used to analyze the performance of algorithms in image processing or pattern recognition tasks. The metrics include the number of image pairs, average and standard deviation of correct and incorrect pairs, histogram bins, and average and standard deviation of bin values. Each histogram is associated with a specific type of metric, such as number of correct mole pairs, correct found pairs, incorrect found pairs, and missed pairs, which are then analyzed as functions of distance and min angle.
A3.8 Format of the Reference and Match Feature File

There are two feature files, one containing the features of reference moles called the Reference Feature File and one containing the features of the match moles called the Match Feature File. Both file are produced by the DetectFeatures program.

The format is:

```
<nr of image pairs>
<filename of image>
<nr of moles in image>
<scaling (to get next features)>
<x pos of mole nr 0> <y pos of mole nr 0>
<feature value 1>
<feature value 2>
...
<x pos of mole nr 1> <y pos of mole nr 1>
<feature value 1>
<feature value 2>
...
```

Same structure for
features of other moles

A3.9 Format of the Mole Difference File

The mole difference file is produced by the Compare program and contains information about which moles have changed in the two successive images.

The format is:

```
<nr of image pairs>
<nr of mole pairs>
<ref mole nr of pair 0> <match mole nr of pair 0>
<nr of features>
<feature1 ref value> <feature1 match value> <relative change> <above threshold>
<feature2 ref value> <feature1 match value> <relative change> <above threshold>
...
<mole changed>
<ref mole nr of pair 1> <match mole nr of pair 1>
<nr of features>
<feature1 ref value> <feature1 match value> <relative change> <above threshold>
<feature2 ref value> <feature1 match value> <relative change> <above threshold>
...
<mole changed>
<ref mole nr of pair 2> <match mole nr of pair 2>
<nr of features>
...
```

etc
Appendix 4  Man pages

This appendix contains the man-pages of all the programs implemented in Khoros. Every man page explains what the program does and what parameters it uses.
A4.1 Man-page of the program Extract

PROGRAM

Extract - Extract mole positions and marker position from images.
.syntax "SCDS" "Extract"

DESCRIPTION

The Extract program finds on the basis of a marked image the positions of mole and corners of the marker in that image. Through the parameter sImage2Extract it receives the name of a file that contains all the paths and filenames of the images that are to be processed by this program. The image names in this file are grouped in image pairs. The first image name is of the reference image of image pair 1, the second is the filename of the match image of image pair 1, the third is the reference image of image pair 2 etc. These filenames are the names of the original images, the Extract program assumes that in the same directory as the original image also the marked image can be found. See below how the images are named. Of the images, the program processes these marked images. In these images the mole centres are marked by a colour that does not appear in the original image, the same goes for the four corners of the marker. Next to each indicated mole centre a 7 bit label is placed to number the moles. See below how images are marked. First it reads the marked reference image and scans through the whole image pixel by pixel. For every marked mole centre it records the position and the label number, of the marked corners of the marker it only records the positions. The same is done for the match image. The positional information retrieved for this image pair is written to the sMolePatFile, sorted according to their label number. The moles are labelled in such a way that a mole in the reference image and a mole in the match image are labelled with the same number if they represent the same mole. On the basis of this labelling the program is able to find out which mole in the reference and match image are the same and form a so called mole pair (or point pair if the moles are perceived as points with a certain position and some feature values). These mole pairs are written to the sCorrPairsFile, and can later be used to check if the registration process finds the correct mole pairs on its own. Optionally the program produces a report file (sReportFile) to which the marker corner positions and the mole positions together with their label number are written of both reference and match image. This file can be used to check if the images were correctly labelled. If there are more image pairs in the sImages2Extract file then they are treated in the same way as the first image pair and their results are appended to the mentioned files.

Filename convention:

Original image: <name>_<scan resolution>.ppm
Marked image: <name>_<scan resolution>_mrk.ppm

For example if the original image called ‘hello’ and scanned with a resolution of 2000 dpi then the filename of the original and marked image are respectively: hello_2000.ppm and hello_2000_mrk.ppm.

Image marking:

The marked image is a copy of the original image in which the mole centre and marker corners are labelled by giving these pixels a certain colour that is not present in the original image. The moles are given a label that consists of 7 bits that have one of two colours which represent a 1 or a 0. These 7 bits make 128 different labels possible. The moles in the reference image and the match image, from the same image pair, that represent the same mole are labelled with the same number. The labels are directly next to the mole centre pixel. There are 4 different orientations of the labels from which one must be chosen:

1) horizontal directly to the right of the mole centre pixel.
2) horizontal directly to the left of the mole centre pixel.
3) vertical directly above the mole centre pixel.
4) vertical directly below the mole centre pixel.

In all cases the most significant bit is the closest bit to the mole centre. For marking the corners of the marker two different colour are reserved, the corners of one long side are marked with one colour, the other long side is coloured with the other colour. The images are RGB-images, stored as ppm-file, in which the colour components are in the range of 0..255. The following colour are used to mark the images:

- Mole centre: (R,G,B) = (0,255,0) -> Green.
- Label 0 bit: (R,G,B) = (0,0,255) -> Blue.
- Label 1 bit: (R,G,B) = (255,0,0) -> Red.
- Colour 1 corner long side: (R,G,B) = (255,255,0) -> Yellow.
- Colour 2 corner long side: (R,G,B) = (0,255,255) -> Cyan.

REQUIRED ARGUMENTS

- sImages2Extract
type: infile
desc: Contains the reference and match image filenames that will be processed by the Extract-program. The images are grouped as image pairs which consist of one reference image and its corresponding follow up match image.

- sMolePatFile
type: outfile
desc: Contains after the execution of Extract all the mole and marker positions of the image pairs in the sImages2Extract file.

OPTIONAL ARGUMENTS

- sCorrPairsFile
type: outfile
desc: File to which the correct molepairs are written.
default: {none}

- sReportFile
type: outfile
desc: All mole and marker corner positions are written to this file, along with the label numbers of the moles found in the marked images. The file is only used to check if the images are correctly marked.
default: {none}

- fMarkerLength_mm
type: float
desc: Length of the marker on body.
default: 120
bounds: value > 0.0

EXAMPLES

SEE ALSO

RESTRICTIONS

REFERENCES

COPYRIGHT

Copyright (C) 1993, 1994, 1995, Khoral Research, Inc. All rights reserved.
A4.2 Man-page of the program DetectFeatures

PROGRAM
DetectFeatures — Detects the features of moles to be able to detect mole changes.
.syntax "SCDS" "DetectFeatures"

DESCRIPTION
This program calculates the features of moles which can be used by the program Compare to check if a mole has changed over the period of time between taking the reference image and the follow up match image. It produces one file (sRefFeatureFile) with the features of all reference moles that are mentioned in the mole pattern file (sMolePatFile) and one file (sMatchFeatureFile) with the features of all match moles. For every mole the following features are calculated:

- Area.
- Perimeter.
- Compactness.
- Average of polar distances.
- Standard deviation of polar distances
- Minimum polar distance.
- Maximum polar distance.
- Eccentricity.

Before the features are calculated, the program first determines which pixels around the mole centre (of which the position is given in sMolePatFile) belong to the mole using the radial search algorithm. To do this, first the area with the mole around the mole centre, defined by a window of iWinWidth x iWinHeight, is copied to a smaller image. In this smaller image, called the mole image, the flash areas, caused by the flash of the camera, are detected. A pixel is defined as being part of a flash area if non of the colour components of the pixel deviate more than ubFlashTh from the average colour component value (R/3+G/3+B/3). The pixels defined as flash area are not taken in account by the radial search algorithm. After this the RGB mole image is converted to a intensity mole image on which the radial search algorithm is applied. The radial search algorithm starts in the centre of the intensity mole image (the mole centre) and casts iNrAngles radial lines from this point at equal angles. Along each radial line the pixels are averaged with their surrounding using a iWidthFilter x iHeightFilter averaging window (the result is a low pass LP filter operation, reducing noise caused by hairs, pores etc.). Then only the radial line pixels are filtered using a 1x1RadialFilter running averaging filter. For each line the border point is detected by finding a sudden intensity increase. All found border points are connected using a cubic spline algorithm, resulting in a closed mole border. And as last step the pixels inside the closest mole border are marked as mole pixels using a region fill algorithm.

REQUIRED ARGUMENTS
- sMolePatFile
type: infile
desc: Contains positions of moles and markers.

- sRefFeatureFile
type: outfile
desc: Features of moles on the reference image.

- sMatchFeatureFile
type: outfile
desc: Features of the moles on the match image.

- iWinWidth
type: integer
desc: Width window in which is searched for a mole.
bounds: value >= 0

- iWinHeight
type: integer
desc: Height of window in which is search for a mole.
bounds: value >= 0
-\texttt{-iNrAngles}
  \begin{itemize}
  \item \texttt{type: integer}
  \item \texttt{desc: Nr of angles used in radial search.}
  \item \texttt{bounds: value > 0}
  \end{itemize}

-\texttt{-iRadialFilter}
  \begin{itemize}
  \item \texttt{type: integer}
  \item \texttt{desc: Size of filter along radial line.}
  \item \texttt{bounds: value > 0}
  \end{itemize}

-\texttt{-iWidthFilter}
  \begin{itemize}
  \item \texttt{type: integer}
  \item \texttt{desc: Width of LP filter.}
  \item \texttt{bounds: value > 0}
  \end{itemize}

-\texttt{-iHeightFilter}
  \begin{itemize}
  \item \texttt{type: integer}
  \item \texttt{desc: Height of LP filter.}
  \item \texttt{bounds: value > 0}
  \end{itemize}

-\texttt{-ubFlashTh}
  \begin{itemize}
  \item \texttt{type: integer}
  \item \texttt{desc: Threshold for finding flash areas.}
  \item \texttt{bounds: value > 0}
  \end{itemize}

\section*{Optional Arguments}
none

\section*{Examples}

\section*{See Also}

\section*{Restrictions}

\section*{References}

\section*{Copyright}

Copyright (C) 1993, 1994, 1995, Khoral Research, Inc. All rights reserved.
A4.3 Man-page of the program InitPntArt

PROGRAM
InitPntArt - Finds initial match points, using a Gabriel graph.
.syntax "SCDS" "InitPntArt"

DESCRIPTION
InitPntArt selects the initial matches for the programs PntRegistrationArt1, PntRegistrationArt2, PntRegistrationArt3 and PntRegistrationPntOwn. The program is based on an algorithm from an article of Perednia and White that selects moles from the reference and match image as matches on the basis of similarity in the distribution of the neighbouring moles. The neighbours are defined by a graph in which the moles are represented by the nodes. The neighbours of a node are those nodes that are connected to the node by an edge. The graph defining the neighbours in the Perednia/White article is the Gabriel graph (see below for definition). For the moles in both the reference image and the match image a Gabriel graph is determined. Every reference node is compared to every match node on the basis of the distribution of its neighbours. The combinations of reference and match nodes that have the most similar distribution of neighbours are selected as initial matches. These initial matches are written to the sInitPairFile.

If the bUseMarkerInfo-switch is in the "yes" position then before the moles in the reference and match image are matched, the marker information in the sMolePatFile is used to remove the scaling differences between the reference and match image.
If required, it is possible to write the reference and match Gabriel graph to the sEdgeFile.

Definition of Gabriel graph: Any node A is said to be adjacent (i.e. connected by an edge) to another node B, if and only if for every other node C, point C does not lie within the circle of diameter \(|AB|\) centred at the midpoint of the line between A and B. In this program a more general definition is used. The maximum number of points inside the circle mentioned above does not have to be zero but can be any number of points (iViolation).

REQUIRED ARGUMENTS
- -sMolePatFile
type: infile
desc: File containing mole and marker positions

- -sInitPairsFile
type: outfile
desc: Contains, after execution, the initial mole pairs.

OPTIONAL ARGUMENTS
- -sEdgeFile
type: outfile
desc: Contains, after execution, the Gabriel graph.
default: {none}

- -bUseMarkerInfo
type: integer
desc: Whether or not marker info is used to remove scale differences.
default: 2
allowed values:
1 (No),
or 2 (Yes)

- -iViolation
type: integer
desc: Determines the way the Gabriel graph is build.
default: 0
bounds: value >= 0

EXAMPLES
SEE ALSO
RESTRICTIONS
A4.4 Man-page of the program InitPntOwn

PROGRAM

InitPntOwn – Finds initial match points, using the baseline algorithm
.syntax "SCDS" "InitPntOwn"

DESCRIPTION

InitPntOwn selects the initial matches for the programs PntRegistrationArt1, PntRegistrationArt2, PntRegistrationArt3 and PntRegistrationPntOwn. The program is selects moles from the reference and match image as matches on the basis of similarity in the distribution of the neighbouring moles. The neighbours are defined by a graph in which the moles are represented by the nodes. The neighbours of a node are those nodes that are connected to the node by an edge. The graph defining the neighbours in this program in the local node graph in which a node has only the two closest nodes as neighbours. For the moles in both the reference image and the match image a local node graph is determined. Every reference node is compared to every match node on the basis of the distribution of its neighbours. The combinations of reference and match nodes that have the most similar distribution of neighbours are selected as initial matches. These initial matches are written to the sInitPairFile.

If the bUseMarkerInfo-switch is in the "yes" position then before the moles in the reference and match image are matched, the marker information in the sMolePatFile is used to remove the scaling differences between the reference and match images.

REQUIRED ARGUMENTS

-sMolePatFile
  type: infile
  desc: Contains mole and marker position

-sInitPairsFile
  type: outfile
  desc: Contains, after execution, the initial mole pairs.

OPTIONAL ARGUMENTS

-bUseMarkerInfo
  type: integer
  desc: Whether or not use marker to remove ref/match scaling differences.
  default: 2
  allowed values:
  1 (No),
  or 2 (Yes)

EXAMPLES

SEE ALSO

RESTRICTIONS

REFERENCES

COPYRIGHT

Copyright (C) 1993, 1994, 1995, Khoral Research, Inc. All rights reserved.
A4.5 Man-page of the program PntRegistrationArtl

PROGRAM

PntRegistrationArtl – Registrates points using 1 initial point pair, based on article of Perednia and White.
.syntax "SCDS" "PntRegistrationArtl"

DESCRIPTION

PntRegistrationArtl is a registration program that searches in the reference image and its succeeding match image for moles that represent the same mole. The positions of the moles in the reference and match images are read from the mole pattern file (sMolePatFile). Reference moles and match moles that represent the same mole are defined as a mole pair and written to the found mole pair file (sFndPairFile). The moles for which no corresponding mole in the other image is found, are written to the unmapped file (sUnmappedFile). The program is based on an algorithm from an article of Perednia and White. It uses 1 initial mole pair to reduce the dimensionality of the matching problem. This initial mole pair, which the program finds at the first position of the initial mole pair file (sInitPairsFile), tells the program of one mole in the reference and one mole in the match image that they represent the same mole. The principle of the program is as follows. The mole pattern of the reference and match image are overlaid. After which the reference and match image mole patterns are translated in such a way that the reference and match mole of the initial pair end up in a common origin. Now the reference image is rotated around this origin in iNrSteps steps from tMinAngle radials to tMaxAngle radials. For each step a correlation value between the reference and match mole pattern is calculated and stored in an array. The correlation value is higher when the mole patterns are more similar. The array is filtered using a iFilterSize running average filter, after which the position in the array is determined where the maximum correlation value is stored. The reference image is then rotated around the common origin over an angle that corresponds to the position were the maximum array value is found. The moles in the reference and match image that are each others mutually closest neighbours are defined as mole pair and written to the found mole pair file (sFndPairFile). The mole that do not have a mutually closest neighbour are registered in the unmapped file (sUnmappedFile).

If the bUseMarkerInfo-switch is in the “yes” position then before the moles in the reference and match image are matched, the marker information in the sMolePatFile is used to remove the scaling differences between the reference and match images.

REQUIRED ARGUMENTS

-sMolePatFile
  type: infile
  desc: Contains positions of moles and markers

-sInitPairsFile
  type: infile
  desc: File of initial mole pairs

-sFndPairFile
  type: outfile
  desc: Contains, after execution, the found mole pairs.

-sUnmappedFile
  type: outfile
  desc: List of moles that could not be paired with other moles.

-tMinAngle
  type: float
  desc: Minimum rotation angle for calculating correlation.
  bounds: no range checking

-tMaxAngle
  type: float
  desc: Max rotation angle for calculating correlation.
  bounds: no range checking

-fEps
  type: float
  desc: Small number to prevent division by zero
  bounds: value > 0.0
-iNrSteps
  type: integer
  desc: Nr of rotation steps for which the correlation is calculated.
  bounds: value > 0

-iFilterSize
  type: integer
  desc: Size of filter for filtering correlation graph.
  bounds: value > 0

-bUseMarkerInfo
  type: integer
  desc: Use marker info to remove ref/match-image scaling diff.
  allowed values:
  1 (No),
  or 2 (Yes)

OPTIONAL ARGUMENTS
  none

EXAMPLES
SEE ALSO
RESTRICTIONS
REFERENCES
COPYRIGHT
  Copyright (C) 1993, 1994, 1995, Koral Research, Inc. All rights reserved.
A4.6 Man-page of the program PntRegistrationArt2

PROGRAM
PntRegistrationArt2 — Registrates points using 2 initial point pairs, based on article of Perednia and White.
.syntax “SCDS” “PntRegistrationArt2”

DESCRIPTION
PntRegistrationArt2 is a registration program that searches in the reference image and its succeeding
match image for moles that represent the same mole. The positions of the moles in the reference and match
images are read from the mole pattern file (sMolePatFile). Reference moles and match moles that
represent the same mole are defined as a mole pair and written to the found mole pair file (sFndPairFile).
The moles for which no corresponding mole in the other image is found, are written to the unmapped file
(sUnmappedFile). The program is based on an algorithm from an article of Perednia and White. It uses 2
initial mole pairs to reduce the dimensionality of the matching problem. These initial mole pairs, which
the program finds at the first two positions of the initial mole pair file (sInitPairsFile), tell the program of
two moles in the reference and two moles in the match image that they represent the same mole. The
principle of the program is as follows. The mole pattern of the reference and match image are overlaid.
The moles in the reference and match image are translated in such a way that the reference and match
mole of the first initial pair are both in a common origin. Then both mole patterns are rotated around the
common origin causing the reference and match mole of the second initial pair to end up on a common
axis, this program uses the x-axis as common axis. After this the reference and match mole patterns are
scaled such that the reference and match mole of the second initial pair coincide. Now the moles in the
reference and match image that are each others mutually closest neighbours are defined as mole pair and
written to the found mole pair file (sFndPairFile). The mole that do not have a mutually closest neighbour
are registered in the unmapped file (sUnmappedFile).

REQUIRED ARGUMENTS
-sMolePatFile
type: infile
desc: Contains positions of moles and markers

-sInitPairsFile
type: infile
desc: File of initial mole pairs

-sFndPairFile
type: outfile
desc: Contains, after execution, the found mole pairs.

-sUnmappedFile
type: outfile
desc: List of moles that could not be paired with other moles.

OPTIONAL ARGUMENTS
none

EXAMPLES
SEE ALSO
RESTRICTIONS
REFERENCES
COPYRIGHT
Copyright (C) 1993, 1994, 1995, Khoral Research, Inc. All rights reserved.
A4.7 Man-page of the program PntRegistrationArt3

PROGRAM

PntRegistrationArt3 -
    .syntax "SCDS" "PntRegistrationArt3"

DESCRIPTION

PntRegistrationArt3 is a registration program that searches in the reference image and its succeeding match image for moles that represent the same mole. The positions of the moles in the reference and match images are read from the mole pattern file (sMolePatFile). Reference moles and match moles that represent the same mole are defined as a mole pair and written to the found mole pair file (sFndPairFile). The moles for which no corresponding mole in the other image is found, are written to the unmapped file (sUnmappedFile). The program is based on an algorithm from an article of Perednia and White. It uses 3 initial mole pairs to reduce the dimensionality of the matching problem. These initial mole pairs, which the program finds at the first three positions of the initial mole pair file (sInitPairsFile), tell the program of three moles in the reference and two moles in the match image that they represent the same mole. The principle of the program is as follows. The mole pattern of the reference and match image are overlaid. The mole pattern of the reference and match image are overlaid, using the three initial mole pairs an affine transformation is calculated. This affine transformation transforms the mole pattern of the reference image in such away that the reference moles of the initial mole pairs coincide with their corresponding match moles. Now the moles in the reference and match image that are each others mutually closest neighbours are defined as mole pair and written to the found mole pair file (sFndPairFile). The mole that do not have a mutually closest neighbour are registered in the unmapped file (sUnmappedFile).

REQUIRED ARGUMENTS

-sMolePatFile
    type: infile
    desc: Contains positions of moles and markers

-sInitPairsFile
    type: infile
    desc: File of initial mole pairs

-sFndPairFile
    type: outfile
    desc: Contains, after execution, the found mole pairs.

-sUnmappedFile
    type: outfile
    desc: List of moles that could not be paired with other moles.

OPTIONAL ARGUMENTS

none

EXAMPLES

SEE ALSO

RESTRICTIONS

REFERENCES

COPYRIGHT

Copyright (C) 1993, 1994, 1995, Khoral Research, Inc. All rights reserved.
PROGRAM

PntRegistrationOwn - Registrates points using 2 initial point pairs, baseline algorithm.
.syntax "SCDS" "PntRegistrationOwn"

DESCRIPTION

PntRegistrationOwn is a registration program that searches in the reference image and its succeeding
match image for moles that represent the same mole. The positions of the moles in the reference and match
images are read from the mole pattern file (sMolePatFile). Reference moles and match moles/points that
represent the same mole are defined as a mole/point pair and written to the found mole pair file
(sFndPairFile). The moles for which no corresponding mole in the other image is found, are written to the
unmapped file (sUnmappedFile). It uses 2 initial mole pairs to solve of the matching problem. These
initial mole pairs, which the program finds at the first two positions of the initial mole pair file
(sInitPairsFile), tell the program of two moles in the reference and two moles in the match image that they
represent the same mole. The principle of the program is as follows. It starts by defining the line between
the two initial point pairs as a baseline, the line between the points in the reference images is called the
reference baseline, the one in the match image is called the match baseline. In the reference image the
point that is closest to the reference baseline is taken as third point of a triangle, which is formed the
reference point and the two points of the baseline. Of this triangle some geometrical properties are
calculated which are used for some kind of similarity metric. In the match image these geometric
properties are calculated for every point with the points of the match base forming the other corners of the
triangle. The point in the match image that has the most similar geometric properties when compared to
those of the point in the reference image, is said to the matched point. This reference and match point are
defined as a point pair. Together with the two points of the baseline two new baselines can be defined,
namely from one point pair of the current baseline to the newly found point pair and from the other point
pair of the current baseline to this new point pair. The reference point is labelled as "registered" and will
never again be selected a being closest to a baseline. Now the point in the reference image is taken that is
closest to one of the available baselines. For this reference point and the selected baseline the matching
point the matching point in the match image is determined on the basis of the similarity metric. This
results in a new point pair with which again two new baselines are formed. Again the reference point is
labelled as "registered" and will not take part in the process any more. This process goes on and on until
all reference points are matched (paired) to a point in the match image. In the end all the formed point pair
are written to the found mole pair file (sFndPairFile) and the points in the match image that are not paired
with a reference point are defined as unmapped and written to the unmapped file (sUnmappedFile).

REQUIRED ARGUMENTS

-sMolePatFile
type: infile
desc: Contains positions of moles and markers

-sInitPairsFile
type: infile
desc: File of initail mole pairs

-sFndPairFile
type: outfile
desc: Contains, after execution, the found mole pairs.

-sUnmappedFile
type: outfile
desc: List of moles that could not be paired with other moles.

OPTIONAL ARGUMENTS

-bUseMarkerInfo
type: integer
desc: Use marker info to remove ref/match scale diff.
default: 2
allowed values:
1 (No),
or 2 (Yes)
-iNrInitPairsGeom
  type: integer
  desc: Nr of initial pairs used to remove ref/match scale diff.
  default: 3
  bounds: value > 0

EXAMPLES
SEE ALSO
RESTRICTIONS
REFERENCES
COPYRIGHT
  Copyright (C) 1993, 1994, 1995, Khoral Research, Inc. All rights reserved.
A4.9 Man-page of the program Compare

PROGRAM

Compare - Compares the found mole pairs for changes in features.
.syntax "SCDS" "Compare"

DESCRIPTION

The program Compare checks if moles, that represent the same mole in reference and match image, have changed over the period of time between the two photo sessions at which the images were taken. From the found mole pair file (sMolePairFile) the program finds out which moles in the reference and match image represent the same mole. For every mole pair in this file it looks up the mole features in the so called feature files. The features of the reference moles it finds in the reference feature file (sRefFeatureFile) and the features of the match moles are in the match feature file (sMatchFeatureFile). The reference and match mole of each mole pair are compared to each other feature by feature. For every feature the relative change of the feature is calculated. If the relative change is larger than a certain, to this feature belonging threshold (one of the fRelTh_ parameters) then this feature is marked as changed. The mole itself is defined as changed if the number of changed features of that mole exceeds a user defined number (iMaxNrViolations). All information is gathered in the comparison process and written to the mole difference file (sMoleDiffFile). The Visualize program uses this information to indicate which moles have changed.

REQUIRED ARGUMENTS

-sRefFeatureFile
type: infile
desc: Features of moles from reference image.

-sMatchFeatureFile
type: infile
desc: Features of moles in the match image.

-sMolePairFile
type: infile
desc: File of registrated/found mole pairs.

-sMoleDiffFile
type: outfile
desc: Contains, after execution, differences between ref and match moles.

-iMaxNrViolations
type: integer
desc: Nr of diff above threshold are acceptable.
bounds: value > 0

OPTIONAL ARGUMENTS

-fRelTh_Area
type: float
desc: Maximum relative change in area between ref and match mole.
default: 0.1
bounds: value > 0.0

-fRelTh_Perimeter
type: float
desc: Maximum relative change in perimeter between ref and match mole.
default: 0.1
bounds: value > 0.0

-fRelTh_Compactness
type: float
desc: Maximum relative change in compactness between ref and match mole.
default: 0.1
bounds: value > 0.0
- fRelTh_AsymmetryIndex
  type: float
  desc: Max change in asymmetry index between ref and match moles.
  default: 0.1
  bounds: value > 0.0

- fRelTh_AvgPolar
  type: float
  desc: Max change in average polar distance between ref and match moles.
  default: 0.1
  bounds: value > 0.0

- fRelTh_StdPolar
  type: float
  desc: Max change in std dev polar distance between ref and match mole.
  default: 0.1
  bounds: value > 0.0

- fRelTh_MinPolar
  type: float
  desc: Max change in minimum polar distance between ref and match mole.
  default: 0.1
  bounds: value > 0.0

- fRelTh_MaxPolar
  type: float
  desc: Max change in maximum polar distance between ref and match mole.
  default: 0.1
  bounds: value > 0.0

- fRelTh_Eccen
  type: float
  desc: Max change in eccentricity between ref and match mole.
  default: 0.1
  bounds: value > 0.0

EXAMPLES
SEE ALSO
RESTRICTIONS
REFERENCES
COPYRIGHT
Copyright (C) 1993, 1994, 1995, Khoral Research, Inc. All rights reserved.
A4.10 Man-page of the program Visualize

PROGRAM

Visusalize - Visualizes which moles are new or changed.
.syntax "SCDS" "Visualize"

DESCRIPTION

The Visualize program forms the interface to the physician, it indicates at which moles the physician must have a closer look. Indicating is done by displaying the original reference and match images. In these images the changed moles are marked with a ubChanged_(R,B,G) coloured cross and the new (unmapped) moles are marked using a cross of the colour ubUnmapped_(R,G,B). The size of the cross is determined by iCrossRadius. The program finds out which moles have changed or are new by reading respectively the mole difference file (sMoleDiffFile) and the unmapped file (sUnmappedFile). The position of the mole, which is needed to draw the crosses on the location with the new or changed moles, are read from the mole pattern file (sMolePatFile).

REQUIRED ARGUMENTS

-`-sMoleDiffFile`
type: infile
desc: Contains differences between corresponding ref and match moles.

-`-sMolePatFile`
type: infile
desc: Positions of moles and markers.

-`-sUnmappedFile`
type: infile
desc: Moles of one image that couldn’t be found in the other.

-`-iCrossRadius`
type: integer
desc: Size of cross that marks new and changed moles.
    bounds: value > 0

-`-ubUnmapped_R`
type: integer
desc: Red-component of marker of new moles.
    bounds: value >= 0

-`-ubChanged_R`
type: integer
desc: Red-component of marker of changed moles.
    bounds: value >= 0

-`-ubUnmapped_G`
type: integer
desc: Blue-component of marker of new moles.
    bounds: value >= 0

-`-ubChanged_G`
type: integer
desc: Green-component of marker of changed moles.
    bounds: value >= 0

-`-ubUnmapped_B`
type: integer
desc: Green-component of marker of new moles.
    bounds: value >= 0

-`-ubChanged_B`
type: integer
desc: Blue-component of marker of changed moles.
    bounds: value >= 0
OPTIONAL ARGUMENTS
  none

EXAMPLES
SEE ALSO
RESTRICTIONS
REFERENCES
COPYRIGHT
  Copyright (C) 1993, 1994, 1995, Khoral Research, Inc. All rights reserved.
A4.11 Man-page of the program Randomize

PROGRAM
Randomize - Randomizes the mole positions for statistical analysis
.syntax "SCDS" "Randomize"

DESCRIPTION
The Randomize program generates "random" reference and match images pairs for the statistical analysis of the registration and initial mole pair selecting algorithms. It reads the image pairs in the mole pattern file (sMolePatFile) one by one, together with the information about which reference and match mole represent the same mole from the correct mole pair file (sCorrMolePairsFile). For every image pair in the mole pattern file it generates iNrTestPerImagePair random images pairs. For each of these image pairs there are at random iNrTestMoles correct mole pairs chosen from the correct mole pair file and stored in memory as the new correct mole pairs. The reference and match moles in these new correct mole pairs are used to compose the reference and match mole patterns. From these mole patterns iNrMisRefMoles and iNrMisMatchMoles are remove from respectively the reference and match mole pattern. Of the new correct mole pairs the ones are removed that consist out of one or two of the removed reference or match moles. The resulting reference and match mole patterns are written to the new mole pattern file (sTestMolePatFile) and the new correct mole pairs are written to the new correct mole pair file (sTestCorrMolePairsFile). Of the new correct mole pairs, three are chosen to serve as initial mole pairs and are written to the initial mole pair file (sTestInitMolePairsFile). This is done for every image pair in the original mole pattern file and the results are appended to the mentioned file above.

REQUIRED ARGUMENTS
- sMolePatFile
type: infile
desc: Mole and marker positions

-sCorrMolePairsFile
type: infile
desc: All correct mole pairs are listed in this file

-sTestMolePatFile
type: outfile
desc: After execution, randomized positions of the moles

-sTestCorrMolePairsFile
type: outfile
desc: After execution, correct mole pairs belonging to randomized mole patterns.

OPTIONAL ARGUMENTS
- sTestInitMolePairsFile
type: outfile
desc: After execution, initial mole pairs for testing registration routines.
default: {none}

-iNrTestMoles
type: integer
desc: Number of moles in ref- and match-image before mole removal.
default: 5
bounds: value > 0

-iNrMisRefMoles
type: integer
desc: Number of ref-moles that are removed
default: 0
bounds: value >= 0

-iNrMisMatchMoles
type: integer
desc: Number of match-moles that are removed
default: 0
bounds: value >= 0

-iNrTestPerImagePair
  type: integer
  desc: Number of image-pair that are randomly taken from 1 image-pair.
  default: 1
  bounds: value > 0

EXAMPLES
SEE ALSO
RESTRICTIONS
REFERENCES
COPYRIGHT

Copyright (C) 1993, 1994, 1995, Khoral Research, Inc. All rights reserved.
A4.12 Man-page of the program Check

PROGRAM
   Check — Checks found mole pairs for correctness.
   .syntax "SCDS" "Check"

DESCRIPTION
   The Check program compares the mole pairs in the found mole pair file (sFndPntPairFile), found by the registration or initial mole pair selecting programs, to the correct mole pairs in the correct mole pair file (sCorrPntPairFile) which are generated by the Extract or Randomize program. It counts how many found mole pairs were correct and incorrect. The number of correct mole pairs that were not found is counted and for the statistics of the initial mole pair selecting programs it counts how many mole pairs are correct in row. All this information is written to the evaluation file (sEvaluationFile) which is used by the Stat program to generate the histograms and other statistical graphs.

REQUIRED ARGUMENTS
   -sMolePatFile
      type: infile
desc: Contains mole and marker positions.

   -sCorrPntPairFile
      type: infile
desc: File containing the correct mole pairs.

   -sFndPntPairFile
      type: infile
desc: Mole pairs found by registration or initial pairs programs.

   -sEvaluationFile
      type: outfile
desc: After execution, info about correctness of found mole pairs.

OPTIONAL ARGUMENTS
   none

EXAMPLES

SEE ALSO

RESTRICTIONS

REFERENCES

COPYRIGHT
   Copyright (C) 1993, 1994, 1995, Khoral Research, Inc. All rights reserved.
A4.13 Man-page of the program Stat

PROGRAM

.syntax "SCDS" "Stat"

DESCRIPTION

The Stat program reads the evaluation file (sEvaluationFile) and the initial mole file (sInitPntPairFile) and generates on basis of this the following histograms:
1) Average and standard deviation of the number of correctly found, incorrectly found and not found mole pairs as function of the distance between the first two initial mole pairs in the initial mole pair file (sInitPntPairFile). It is used for the evaluation of PntRegistrationOwn.
2) Average and standard deviation of the number of correctly found, incorrectly found and not found mole pairs as function of the minimum angle of the triangle formed by the first three initial mole pairs in the initial mole pair file (sInitPntPairFile). It is used for the evaluation of PntRegistrationArt1 - PntRegistrationArt3.
3) Average and standard deviation of the correctness of the initial mole pairs as function of the rank of the mole pair. It is used for the evaluation of InitPntArt and InitPntOwn. All histograms are written to the statistics file (sStatisticsFile) and can be displayed using Matlab.

REQUIRED ARGUMENTS

-sEvaluationFile
type: infile
desc: File of evaluated mole pairs.

-sMolePatFile
type: infile
desc: Mole and marker positions

-sInitPntPairFile
type: infile
desc: File of initial mole pairs.

-sStatisticsFile
type: outfile
desc: Produced statistics

OPTIONAL ARGUMENTS

-iNrHistoBins
type: integer
desc: Nr bins Angle/Distance histogram.
default: 30
bounds: value > 0

EXAMPLES

SEE ALSO

RESTRICTIONS

REFERENCES

COPYRIGHT

Copyright (C) 1993, 1994, 1995, Khoral Research, Inc. All rights reserved.