Proceedings of the 7th WSEAS International Conference on Automation & Information, Cavtat, Croatia, June 13-15, 2006 (pp167-172)

Foot Study: A System for Measuring Foot Geometry

MIHA GRČAR[1], NEJC ŠARABON[2,3], JANKO STREL[3], GREGOR STARC[3], JURE LABROVIČ[3]

[1]Dept. of Knowledge Technologies
Jožef Stefan Institute
Jamova 39, 1000 Ljubljana
SLOVENIA

[2]Olympic Rehabilitation Center Zdravilišče Šmarješke Toplice

[3]Faculty of Sport, University of Ljubljana

Abstract: - In this paper we present a system for measuring foot geometry using a footprint. Such analyses are of great importance in medicine, footwear industry, and sport science. The presented automated system has several advantages over the manual analysis which is still quite common. The key advantage is in the fact that the system strictly follows mathematical and/or algorithmic definitions of quantities that need to be measured. This way the performed analyses are objective and always repeatable. The system also counters other shortcomings of the manual analysis such as: time consumption, susceptibility to errors, inability to build large repositories of performed analyses, and lack of skills needed for the preparation of clear and concise reports. It is wrapped into an intuitive graphical user interface which makes it easy to use. In this paper the system is presented from two different points of view. On one hand, we present some key algorithms that form the core of the system, and on the other hand, the graphical user interface as seen (or "experienced") by the end-user is described. We believe that we have correctly identified and addressed the issues related to the manual foot geometry analysis and that the Foot Study system will initiate a series of research work involving foot geometry analysis.

Key-Words: - foot geometry, footprint analysis, foot geometry quantities/measures definitions, software application, image processing, contour tracing, algorithms, graphical user interface

1 Introduction

The measuring of foot geometry is of great importance in medicine, footwear industry and sport science. Despite technological capabilities the analyses are predominantly still conducted manually. Ink is applied onto a foot which is then pressed against a piece of paper. Afterwards, pencils and rulers are used to visualize and measure anthropometric quantities that are required in the given context. Such an approach has many shortcomings, some of which are listed below:

• Analyses by hand are time-consuming. Even a versed expert needs a lot of time to perform them accurately.

• Susceptibility to errors. The human factor can cause errors in the performed analyses.

• Vaguely defined measures. Quantities that need to be measured are at most vaguely defined. An exact (i.e. mathematical or algorithmic) definition is theoretically possible but it is difficult for a human expert to follow it exactly. For example, it is difficult – if not impossible – to see exactly at which point "the tangent line of a footprint contour turns from 'vertical' to 'horizontal' orientation."

• The need to archive measurements and to maintain a central repository of measured subjects. In our opinion it is crucial for certain segments of science (let that be research in footwear industry, medicine or sport science) to build a large database of measurements. The shortcomings of the manual analysis act against such repository. The time consumption prevents a decent growth of the database. Furthermore, errors caused by the human factor and vaguely defined quantities that encourage subjectivity make it impossible to compare measurements from different sources.

• Clarity and aesthetic perfection of reports on performed analyses. Each report should contain an image of the given footprint visualizing all the measured quantities and a table with the actual values of the quantities. It is unrealistic to expect that the measurer – a scientist, a doctor or a footwear salesman – has the skills necessary for the preparation of a high-quality report using merely pencils and rulers.

These shortcomings initiated the development of a software application that eliminates the human factor, is fast and accurate and follows strict definitions of quantities to be measured. The application also maintains a central repository of measurements. Furthermore, it provides the user (i.e. the measurer) with an intuitive graphical user interface, clear visualization and strong support in creating standardized reports.

2 Literature Overview

Our work combines topics discussed in literature from various areas. Generally, this literature can be divided into: (i) literature on image processing, (ii) articles on contour tracing, and (iii) literature from the areas of kinesiotherapy, pediatrics, orthopedics, physiatry, sports science, and footwear industry.

From the latter segment, the study by Gros [1] dealing with the initial approach to formalization of quantities measured in footwear industry needs to be mentioned. Additionally, some quantities primarily related to the incorrect foot development are described in the book by Stanić [2]. Other resources from this segment need not to be mentioned as they are inconsistent with the focus of this paper.

When searching for image processing algorithms, the books by Glassner [3] and Russ [4] proved to be the most useful resources. ACM Digital Library <*http://portal.acm.org*> and CiteSeer <*http://citeseer.ist.psu.edu*> provided information on contour tracing algorithms. As it turned out, our contour tracing problem is rather specific and has not been addressed widely in literature. The problem of contour tracing is usually approached to as a problem of tracing the points which are 4- or 8-connected (see Toussaint [5]). Since our case is rather different, other works on contour tracing are not listed herein.

The remaining literature contains information on parametric curves (Anderson [6], Dodgson [7]) and standards for descriptive languages [8, 9, 10, 11]. These works are referred to in the corresponding parts of the text.

3 Description of the Algorithms

The majority of algorithms forming the core of the footprint analysis software presented in this paper were developed for the purpose of the project. The need for developing new algorithms arose mainly from specific procedures and problems experienced during the development of the application.

3.1 Pre-processing of the Digitized Footprint

First, the digitized footprint is transformed into a 1-bit (i.e. black and white) image. Afterwards, the size of this image is changed in order to achieve the required resolution (e.g. 100 x 100 DPI). In this way, digitized footprints are normalized up to a certain point and prepared for further processing. The filter used for transforming the color depth of the image into a 1-bit depth is very straightforward. The user adjusts the filter sensitivity ($t \in [0, \sqrt{3}]$) to set the boundary between black and white pixels of the target image. The pixel of the target image (P_O) is determined with the following formula:

$$P_O = \begin{cases} (1,1,1) & \text{if } d(P_I, (0,0,0)) \geq t \\ (0,0,0) & \text{otherwise} \end{cases}$$

where P_I is the pixel with the same position on the original image. In this formula, each pixel is assumed to have three components – red, green and blue – where the value (0,0,0) represents pure white color and the value (1,1,1) pure black color.

3.2 Footprint Contour Tracing

Despite the normalization, the quality of a 1-bit image of a footprint is still questionable, especially when considering the density of black pixels separating the footprint from the background. The density of black pixels depends on the sensitivity of the importing filter, the quality and quantity of used ink and also on the way the footprint was taken. This means that we can create a "strong" footprint in one extreme and a very "weak" one in the other extreme. Our contour tracing algorithm was developed in such a way that it can be adapted to the quality of the footprint.

The algorithm first sets the staring point of contour tracing somewhere along the edge of the footprint. This point is the starting origin of the direction vector s. The vector s then orientates itself automatically along the edge of the footprint. This is achieved with the use of a loop, which rotates s around the point of origin, gradually changing the angle for a relatively small $\Delta\varphi$. After each step, the algorithm evaluates the quality of the resulting vector position. Let us assume that the vector is gradually being rotated counter clockwise. In this case, the evaluation of the quality of the position (according to the labels in Fig. 1) is given by the following equation:

$$q = w_{reward} \times PixelCount(\Gamma) - w_{punish} \times PixelCount(\Phi),$$

where w_{reward} and w_{punish} are weights used to emphasize the rewarding of black pixels lying in the area Γ and/or punishing of black pixels lying in the area Φ. Function $PixelCount(\Theta)$ counts the black pixels in the area Θ.

It needs to be mentioned that the appropriate setting of the parameters enables us to adapt the algorithm to the footprint quality. If the footprint is (extremely) weak, the weighting parameters are set in such a way that w_{punish} is much higher than w_{reward} and vice versa. Other parameters also help us to adapt the algorithm to the footprint quality and the resolution (DPI) used in the procedures.

Once the direction vector is correctly orientated (i.e. when it is in the position with the highest q), its endpoint is set as the origin of the new direction vector and the orientation procedure is repeated. The whole procedure is finished when the endpoint of the last orientated direction vector is close enough to the starting point of contour tracing. The origins of direction vectors are used as control points for calculating the curve representing the footprint contour. The curve type used for modeling the footprint contour is the closed uniform 4th order B-spline curve [6,7].

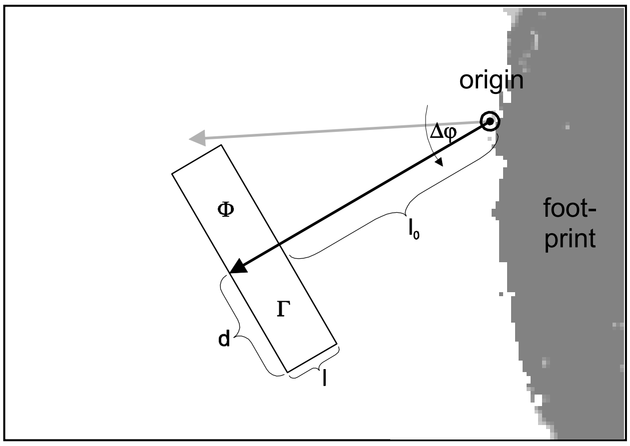

Fig. 1: The direction vector in the process of searching for the optimal position during contour tracing.

3.3 Setting the Coordinate System

The coordinate system is of key importance in measuring procedures. It determines the origin and axes for the positioning of points between which the distances, angles and other quantities are measured. For example, many quantities by their definition require to be measured parallel or orthogonal to one of the two coordinate axes. The incorrect positioning of the coordinate system therefore results in the incorrect calculation of such quantities. This has been avoided by positioning the coordinate system with an automatic procedure. This procedure is described below.

The Y axis of the coordinate system is defined as the mid-line of the foot. The mid-line of the foot is the line connecting the centers of circles drawn inside the heel and the 2nd toe (inscribed circles of the heel and the 2nd toe, respectively). The user determines the center of the 2nd toe manually, whereas the center of the circle inscribed within the heel is determined by the following algorithm:

The Y axis is defined by two points: $\mathbf{Y_1}$ is placed in the center of the 2nd toe and $\mathbf{Y_2}$ is placed somewhere within the boundaries of the footprint. The algorithm described below corrects the position of $\mathbf{Y_2}$ and sets it as the center of the circle inscribed within the heel.

1. First an orthogonal line is drawn onto the line $\mathbf{Y_1Y_2}$ through $\mathbf{Y_2}$. Then the leftmost and the rightmost intersections of the orthogonal line and the footprint contour are determined. The left intersection is marked $\mathbf{P_l}$ and the right intersection $\mathbf{P_r}$.
2. Point $\mathbf{Y_2}$ is moved to the middle of the line $\mathbf{P_lP_r}$.
3. Next, the following condition is tested: $|d(\mathbf{Y}_2,\mathbf{T}) - d(\mathbf{Y}_2,\mathbf{P}_r)| \le \varepsilon$, where $\mathbf{T}$ is the point in which the line through $\mathbf{Y_1}$ and $\mathbf{Y_2}$ intersects the footprint contour at the heel and ε is a relatively small positive number. If this condition is fulfilled, the algorithm is ended, otherwise it continues to step 4.
4. Point $\mathbf{Y_2}$ is moved along the vector $\mathbf{Y_1Y_2}$ for the distance $d(\mathbf{Y}_2,\mathbf{T}) - d(\mathbf{Y}_2,\mathbf{P}_r)$. If the result is a positive value, point $\mathbf{Y_2}$ is moved towards point $\mathbf{T}$, otherwise it is moved away from point $\mathbf{T}$.
5. Iteration is repeated from step 1.

The X axis is directed orthogonally onto the Y axis. Since we have a footprint contour which represents a closed curve, the origin of the coordinate system can be defined in relation to the contour. The X axis is placed onto the bottommost point of the footprint curve in such a way that it is clockwise-orthogonal to the Y axis. In this way the positioning of the coordinate system is uniquely defined.

4 Formalized Definitions of Quantities

The definitions of quantities calculated from a footprint by the experts require a mathematical or an algorithmic formalization which enables a standardized procedure for measuring a particular quantity. Consequently the measurements conducted by different experts or different applications for the purpose of analysis are comparable and repeatable.

Below is a list of certain quantities and their definitions:

• The mid-line of the foot determined from the footprint: see Section 3.

• The 1st and 5th metatarsal point determined from the footprint (**MTP_1** and **MTP_5**): when analyzing the left foot the position of **MTP_1** in the top right quarter of the footprint corresponds to its maximum distance from the Y axis; **MTP_5** is similarly placed in the top left quarter of the footprint. The user can move these two points, however, only along the footprint contour.

• The foot length determined from the footprint: the distance between the furthermost points of the footprint contour along the Y axis.

• The orthogonal foot width determined from the footprint: the distance between **MTP_1** and the point where the orthogonal line onto the Y axis running through **MTP_1** intersects the footprint contour on the side of the 5th toe.

• The anatomical foot width determined from the footprint: the distance between **MTP_1** and **MTP_5**.

• The foot length to **MTP_1** determined from the footprint: the distance between the point where the Y axis intersects the footprint contour at the heel and the orthogonal projection of **MTP_1** onto the Y axis.

• The inner anatomical foot width determined from the footprint: the distance between **MTP_1** and the point where the line going through **MTP_1** and **MTP_5** intersects the Y axis.

• The foot length to **MTP_5** determined from the footprint: the distance between the point where the Y axis intersects the footprint contour at the heel and the orthogonal projection of **MTP_5** onto the Y axis.

• The angle of the metatarsophalangeal joint line determined from the footprint: the angle between the mid-line and the line running through **MTP_1** and **MTP_5** (measured at the outer bottom or the inner top side of the footprint).

• Clark's angle: the angle with **MTP_1** as the vertex. The first ray runs through the most protruding point in the inner bottom quarter of the footprint contour (point **A**). The second ray runs through point **B**. Point **B** is defined as the "point where the footprint contour, when "traveling" from point **MTP_1** towards point **A**, changes from horizontal to vertical orientation". Or more formal, it is defined as the point on the footprint contour where the differential (i.e. curve's tangent line slope) reaches or exceeds value 1.

Other quantities are defined in a similar manner.

5 Introduction of the Graphical User Interface

5.1 The Footprint Acquisition Wizard

By choosing the option *Measurement/New...* from the main application window menu the user starts the new-project wizard. The first window is intended for entering information about the subject to be analyzed; nevertheless, the wizard also works without entering any information.

The following window enables the user to import images. If a special device for image acquisition is used (this device exceeds the framework of this paper), the footprint can be acquired in real time otherwise it can be imported as an image file. The application requires the resolution with which the foot was digitized – this information enables the normalization of the resolution. The resolution is usually specified in the metadata of the image file. However, if the application cannot determine this data correctly, it can also be entered manually. The user can also crop and scale the acquired or imported images – these two options correct the most obvious flaws in the currently used specialized digitizing equipment.

The next window requires the entering of the pre-processing parameters. The user has to set the sensitivity of the simple intensity filter by experience. Similarly, the "side" of the foot (i.e. left or right) and the color of the background (i.e. black or white) have to be set manually. The last two parameters are used in the normalization process; if necessary, horizontal mirroring and inverting of the image are performed.

5.2 Setting the Coordinate System and Footprint Contouring

When the work with the new-project wizard is finished, a pre-processed normalized image of the footprint appears in the main window of the application. In the next step, the user has to generate the footprint contour and position the coordinate system.

Around the footprint (without toes), a free path has to be ensured for the direction vector which iteratively places the control points of the contour curve. This path is usually obstructed in the area between the toeprints and the rest of the footprint. In other words, the toeprints need to be excluded from the process of contouring. For this purpose, two tools can be used: the rectangle which determines the active area of contour tracing and the eraser which enables the user to erase parts of the footprint. This tool can be accessed from the menu by selecting *Tools/Eraser*. The rectangular active area of contour tracing can always be visualized on the workspace. The user can set this rectangular area according to his needs. Fig. 2 shows the active area of contour tracing and the use of the eraser for erasing toeprints which due to their position still remain within the active area of contour tracing.

Once the conditions for contour tracing are fulfilled, all the user needs to do is to set the starting point of contour tracing anywhere on the edge of the footprint. The contour tracing algorithm described in Section 3.2 calculates the curve of the footprint contour (see Fig. 3). After contour tracing, the coordinate system can be positioned.

The coordinate system is positioned by selecting two reference points: the first one is placed in the center of the 2nd toe and the second one anywhere on the heel. The algorithm described in Section 3.3 moves the reference point on the heel to its appropriate position and in this way determines the position of the coordinate system. This means, the application has all the necessary data for calculating the quantities listed in Section 4.

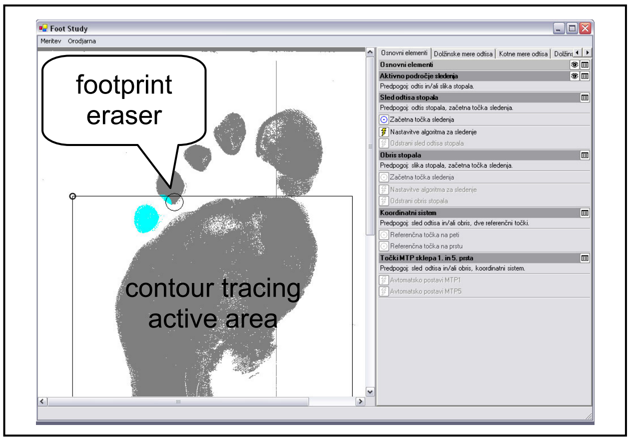

Fig. 2: The active area of contour tracing and the use of the eraser for removing the toeprints.

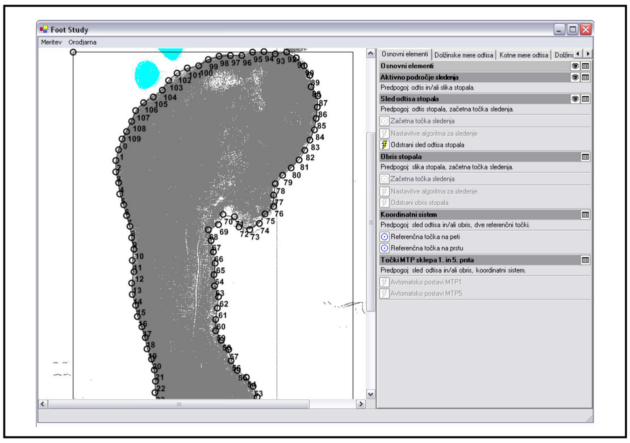

Fig. 3: Visualization of the curve after the contour tracing procedure.

5.3 Calculation of Quantities and their Visualization

After the coordinate system has been positioned, all the quantities which the application can calculate are automatically visualized (Fig. 4 shows such visualization). The quantities are divided into some basic categories (Basic elements, Lengths, Angles, Additional measures). Each category contains a list of all the measures with their corresponding control options and actual values in millimeters, degrees or square millimeters. The control options basically allow the user to hide and show the vector which visualizes a particular quantity (by clicking on the corresponding eye icon). Additional options are available for some quantities (e.g. automatic repositioning of the vector representing the quantity).

It has to be added that the user can interactively reposition the quantity visualizing vectors by repositioning the points visualized as circles (o). The points which the user cannot reposition are not visible or are visualized as crosses (x). When the user changes the position of one point, all the quantities related to this point are automatically recalculated.

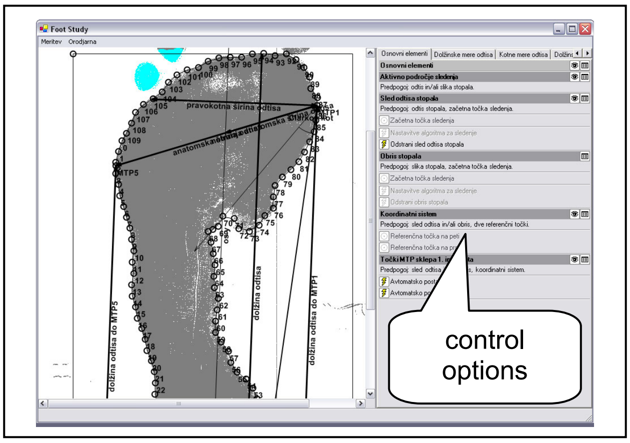

Fig. 4: Visualization of the calculated quantities.

5.4 Working with Data Files

The application allows the user to save (*Measurement/Save*, *Measurement/Save As…*) and reopen the projects (*Measurement/Open…*). These options are quite straightforward and are therefore not explained herein.

Additionally, the user can export (*Measurement/Export…*) the current visualization to an image file of type BMP or to a vector image format of type SVG (*Scalable Vector Graphics* [8]). The latter can be imported by a number of applications used for working with vector graphics (e.g. CorelDRAW). With these applications, SVG files can be converted to practically any vector format (e.g. EPS).

5.4 Preparation and Printing of Reports

This segment has received special attention as clear and concise reports are one of the most important contributions of the described software.

The reports can have a user-defined layout which needs to be formulated in the XSLT descriptive language (*eXtensible Stylesheet Language Transformation* [9]). The XSLT language is generally used for describing procedures which transform the source description in the XML language (*eXtensible Markup Language* [10]) into the target description in the XML language. In our case, the source description is clearly defined (it contains values of quantities, current visualizing features and a lot of other metadata), whereas the target description has to be in the HTML format (*HyperText Markup Language* [11]; format which is also used for creating websites).

Fig. 5 shows the print preview (*Measurement/Print Preview…*) of the report on the performed analysis.

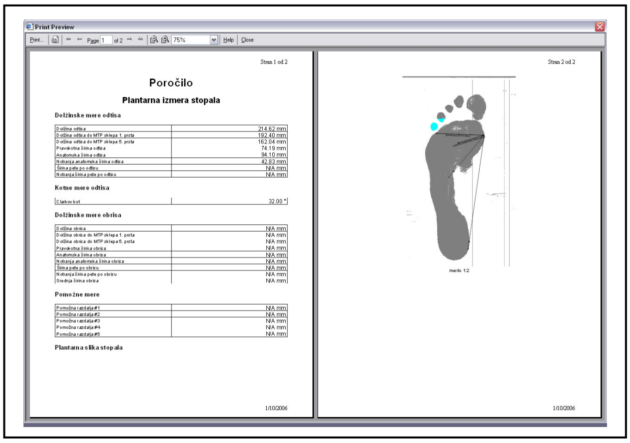

Fig. 5: The print preview of a report.

6 Conclusion

In this paper the described footprint analysis software is presented from both the algorithmic and the user interface point of view. We believe that the problems which occur in the manual analysis have been addressed correctly. The procedure has become faster and the human factor has been eliminated up to a certain point. Additionally, mathematical and/or algorithmic formalisms were used to define the measurements calculated by the application. The user can also create clear and concise reports about the performed measurements.

At the moment, correlations between the circumferences of certain foot sections and the plantar quantities measured by the software are being looked into; however, the results are not yet available.

In the framework of the software application development we would like to increase the level of automation and further eliminate the human factor. At the same time, the list of quantities which this software application can calculate will be extended. A larger number of angular and area quantities will be introduced. The whole process will be extended to a digitized foot image (as a complement to the digitized footprint) and also to a digitized image of the side of the foot.

Finally, we need to implement a system for maintaining a central repository of the performed analyses.

Acknowledgements:

This work was supported by Slovenian Research Agency, Alpina, d.d., and Foundation for financing sport organizations.

The authors would also like to thank Tanja Brajnik for her help.

References:

[1] Gros, J.: *Standardization of Parts of Footwear*. Research study material, 1990.

[2] Stanić, J.: *Physical Education of Pupils with Disabilities*. Školska knjiga, Zagreb, 1979.

[3] Glassner, A.S.: *Graphics Gems*. ISBN: 0122861663, Morgan Kaufman, 1990.

[4] Russ, J.C.: *The Image Processing Handbook*. ISBN: 0849325323, CRC Press, 1998.

[5] Toussaint, G.: Course Notes: Grids, Connectivity and Contour Tracing. At: *http://www.cs.mcgill.ca/~aghnei/contour_tracing.ps*, 2000.

[6] Anderson, F.: Bezier and B-spline Technology. At: *www.cs.umu.se/education/examina/Rapporter/461.pdf*, 2003.

[7] Dodgson, N.: Bezier Curves and B-splines. At: *http://www.cl.cam.ac.uk/Teaching/1999/AGraphHCI/SMAG/node3.html*, 1999.

[8] W3C: Scalable Vector Graphics Specification. At: *http://www.w3.org/TR/SVG/*, 2003.

[9] W3C: XSL Transformations. At: *http://www.w3.org/TR/xslt*, 1999.

[10] W3C: Extensible Markup Language 1.0. At: *http://www.w3.org/TR/REC-xml/*, 2004.

[11] W3C: HTML 4.01 Specification. At: *http://www.w3.org/TR/html401/*, 1999.