

## Improving the Modeling of the Plantar Surface via Bezier Networks and Laser Projection

J. Apolinar Muñoz Rodríguez

*Centro de Investigaciones en Optica A.C. Apartado  
Postal 1- 948, León, GTO, 37000, México  
e-mail: munoz@foton.cio.mx*

**crossref** <http://dx.doi.org/10.5755/j01.itc.43.4.5766>

**Abstract.** An automatic technique to improve the plantar surface model via Bezier networks is presented. This technique generates the foot sole model via Bezier networks based on surface points, which are retrieved via line projection. The surface model is defined by means of network weights and the control points, which are measured via line position by a Bezier network. Thus, the model represents the plantar surface with high accuracy. It is because the model passes through by all control points of the physical surface. Furthermore, the network reduces operations and memory size to calculate the surface. It is because the model is implemented with less mathematical terms than the traditional models. Also, the calibration of vision parameters is performed via laser line to avoid external calibration, which determines vision parameters outside of the vision system and increases the surface representation inaccuracy. Additionally, the shoe-last bottom is adjusted to plantar surface model. The viability of the proposed surface modeling is corroborated by an evaluation based on the speed, accuracy and memory size of the traditional surface models. Thus, the contribution of the proposed technique is elucidated.

**Keywords:** Foot sole model; laser line projection; Bezier networks.

### 1. Introduction

Nowadays, the foot sole data plays an important role in the foot care, gait characteristics and footwear manufacture [1], [17], [26]. Thus, computational models have been implemented to represent the plantar surface [2]. In this field, algorithms have been implemented to construct the plantar surface model [3], [7]. This model provides data to perform the diagnoses of the gait characteristics and the body functionality [28]. The standard method that constructs the surface model is the non-uniform rational B-Splines (NURBS) [19]. This method consumes long time and huge memory to represent the plantar surface. It is because the model is generated by a great amount of terms [25]. Moreover, this model does not interpolate all surface points. Also, this method performs the surface measurement via external calibration, which increases 0.3 % the surface representation [29]. The same criterion is corroborated by the models generated via B-Splines and least squared. Therefore, the improvement of plantar surface model still represents a challenge task. To improve the plantar surface modeling, it is necessary to develop a model that interpolates all surface points by using small memory size in short time.

The proposed technique constructs the plantar surface model via Bezier networks to improve the accuracy, speed and memory size of the traditional models. The surface measurement is performed by a Bezier network via line projection. Also, the vision parameters are calibrated inside of vision system via laser line to avoid physical measurements of external calibration. Thus, the surface points are measured with high accuracy to build the plantar surface model. This model is defined via network weights, which are computed by an equation system to interpolate all control points. Thus, the plantar surface is represented with high accuracy. Also, this model reduces operations and memory size to compute the surface. It is because the network is implemented with less mathematical terms than the traditional models. Additionally, the shoe-last bottom is adjusted to the plantar surface model. The viability of this model is elucidated by an evaluation based on model accuracy, representation accuracy, operations number and memory size of the traditional models. The paper describes the surface model in Section 2, the surface measurement in Section 3, the calibration in Section 4, the plantar surface modeling in Section 5 and the evaluation in Section 6.

## 2. Basic theory

The proposed Bezier network constructs the planar surface model via control points  $P_{00}, P_{01}, P_{02}, \dots, P_{33}$ , which are shown in Fig.1 as a 4x4 surface segment. The positions of the points are depicted by  $(u_{00}, v_{00}), (u_{00}, v_{01}), (u_{00}, v_{02}), \dots, (u_{33}, v_{33})$ , respectively. Thus, the surface model is constructed by the next network

$$S(u, v) = \sum_{j=0}^m \sum_{i=0}^n W_{ij} B_i(u) B_j(v) P_{ij},$$

$$0 \leq u \leq 1, 0 \leq v \leq 1, \quad (1)$$

$$B_i(u) = \binom{n}{i} u^i (1-u)^{n-i}, \quad \binom{n}{i} = \frac{n!}{i!(n-i)!}.$$

For this equation,  $W_{ij}$  are the weights and  $B_i(u)B_j(v)$  are the Bezier functions [16]. Thus, the network given in Eq.(1) generates each model by the next equation

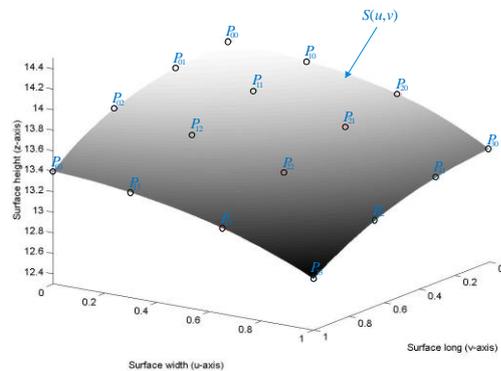
$$S(u, v) = (1-u)^3(1-v)^3 W_{00} P_{00} + 3(1-u)^2 u(1-v)^3 W_{10} P_{10} + \dots + u^3(1-v)^3 W_{30} P_{30} + 3(1-u)^3(1-v)^2 v W_{01} P_{01} + \dots + 3u^3(1-v)^2 v W_{31} P_{31} + \dots + u^3 v^3 W_{33} P_{33}. \quad (2)$$

This network provides continuity in the internal control point via Bezier continuity [20]. The continuity in the edge points is deduced based on the surfaces  $S(u, v)$  and  $Q(u, v)$ , which are shown in Fig.2.

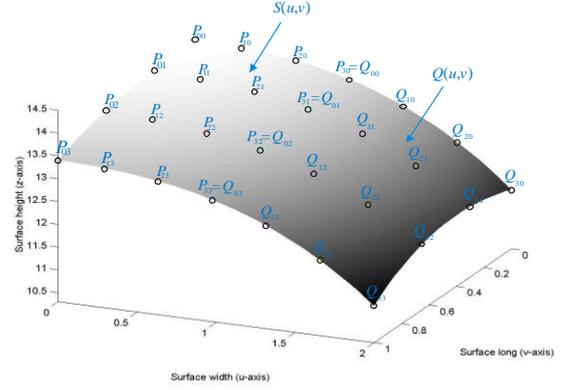
The continuity  $C^0$  is reached when  $S(1, v) = Q(0, v)$  for the interval  $0 \leq v \leq 1$ . Based on Eq. (2), the continuity  $C^0$  is deduced by the next equation of common edge

$$\sum_{j=0}^3 W_{3,j} B_j(v) P_{3,j} = \sum_{j=0}^3 W'_{0,j} B_j(v) Q_{0,j}. \quad (3)$$

This continuity is achieved by using the last points of  $S(u, v)$  as the first points of  $Q(u, v)$ :  $P_{30} = Q_{00}, P_{31} = Q_{01}, P_{32} = Q_{02}, P_{33} = Q_{03}$ . The continuity  $G^1$  is deduced based on the plane tangent to the surface via derivatives:  $\partial S(u, v) / \partial u$  and  $\partial Q(u, v) / \partial u$ . Thus, the continuity  $G^1$  is described by the next equation



**Figure 1.** Surface generated via Bezier network  $S(u, v)$  based on control points



**Figure 2.** Connection of the surfaces  $S(u, v)$  and  $Q(u, v)$  via  $G^1$

$$\left. \frac{\partial S(u, v)}{\partial u} \right|_{u=1} = \alpha \left. \frac{\partial Q(u, v)}{\partial u} \right|_{u=0}. \quad (4)$$

To achieve this continuity, a Bezier function  $B(u)B(v)F_{ij}$  is added in Eq.(2) before the edge points. Thus, a network of fifth degree is obtained and its derivatives are  $\partial S(u, v) / \partial u = 5P_{3j} - 5F_{3j}$  and  $\partial Q(u, v) / \partial u = 5F_{0j} - 5Q_{0j}$ .

To satisfy Eq.(4), the values  $Q_{0j} = F_{3j} = F_{0j} = P_{3j}$  are established. This means that  $F_{3j}, Q_{0j} = P_{3j}$  and  $F_{0j}$  are equally spaced in straight line and  $\partial S(u, v) / \partial u = \partial Q(u, v) / \partial u$ . Based on these continuity parameters, the following network is obtained

$$S(u, v) = (1+4u)(1-u)^4 [(1+4v)(1-v)^4 P_{00} + 10(1-v)^3 v W_{01}^2 P_{01} + 10(1-v)^2 v^2 W_{02} P_{02} + (5-4v)v^4 P_{03}] + 10(1-u)^3 u^2 [(1+4v)(1-v)^4 W_{10} P_{10} + 10(1-v)^3 v W_{11} P_{11} + 10(1-v)^2 v^2 W_{12} P_{12} + (5-4v)v^4 W_{13} P_{13}] + 10(1-u)^2 u^3 [(1+4v)(1-v)^4 W_{20} P_{20} + 10(1-v)^3 v^2 W_{21} P_{21} + 10(1-v)^2 v^3 W_{22} P_{22} + (5-4v)v^4 W_{23} P_{23}] + (5-4u)u^4 [(1+4v)(1-v)^4 P_{30} + 10(1-v)^3 v^2 W_{31} P_{31} + 10(1-v)^2 v^3 W_{32} P_{32} + (5-4v)v^4 P_{33}]. \quad (5)$$

Thus, the network of Eq.(5) preserves continuity  $G^1$ , which is enough to connect two surfaces in a good manner [21]. The interpolation is obtained via weights  $W_{ij}$ . The weights  $W_{00} = W_{30} = W_{03} = W_{33} = 1$  are initially established. Then, the point  $S(u, v) = P_{10}$  and its position  $(u_{10} = 1/3, v_{10} = 0)$  are replaced in Eq.(5) to obtain Eq.(6). Also,  $S(u, v) = P_{20}$  and  $(u, v) = (u_{20} = 2/3, v_{20} = 0)$  are replaced in Eq.(5) to obtain Eq.(7):

$$P_{10} = (1+4u_{10})(1-u_{10})^4 P_{00} + 10(1-u_{10})^3 u_{10}^2 W_{10} P_{10} + 10(1-u_{10})^2 u_{10}^3 W_{20} P_{20} + (5-4u_{10})u_{10}^4 P_{30}, \quad (6)$$

$$P_{20} = (1+4u_{20})(1-u_{20})^4 P_{00} + 10(1-u_{20})^3 u_{20}^2 W_{10} P_{10} + 10(1-u_{20})^2 u_{20}^3 W_{20} P_{20} + (5-4u_{20})u_{20}^4 P_{30}. \quad (7)$$

Solving these equations,  $W_{10}$  and  $W_{20}$  are obtained. In the same manner,  $W_{01}$  and  $W_{02}$  are computed by substituting the points  $\{(u, v) = (u_{01} = 0, v_{01} = 1/3), S(u, v) = P_{01}\}$  and  $\{(u, v) = (u_{02} = 0, v_{02} = 2/3), S(u, v) = P_{02}\}$  in Eq.(5). Also, this procedure calculates  $W_{1,3}, W_{2,3}, W_{31}$  and  $W_{32}$ . Then, the points  $\{(u_{11} = 1/3, v_{11} = 1/3),$

$S(u,v)=P_{11}$ },  $\{(u_{21}=2/3, v_{21}=1/3), S(u,v)=P_{21}\}$ ,  $\{(u_{12}=1/3, v_{12}=2/3), S(u,v)=P_{12}\}$  and  $\{(u_{22}=2/3, v_{22}=2/3), S(u, v) = P_{22}\}$  are substituted in Eq.(5) to obtain the next equation system

$$\begin{bmatrix} P_{11}-R_{11} \\ P_{21}-R_{21} \\ P_{12}-R_{12} \\ P_{22}-R_{22} \end{bmatrix} = \begin{bmatrix} B_2(u_{11})B_2(v_{11}) & B_3(u_{11})B_2(v_{11}) & B_2(u_{11})B_3(v_{11}) & B_3(u_{11})B_3(v_{11}) \\ B_2(u_{21})B_2(v_{21}) & B_3(u_{21})B_2(v_{21}) & B_2(u_{21})B_3(v_{21}) & B_3(u_{21})B_3(v_{21}) \\ B_2(u_{12})B_2(v_{12}) & B_3(u_{12})B_2(v_{12}) & B_2(u_{12})B_3(v_{12}) & B_3(u_{12})B_3(v_{12}) \\ B_2(u_{22})B_2(v_{22}) & B_3(u_{22})B_2(v_{22}) & B_2(u_{22})B_3(v_{22}) & B_3(u_{22})B_3(v_{22}) \end{bmatrix} \begin{bmatrix} W_{11}P_{11} \\ W_{12}P_{12} \\ W_{21}P_{21} \\ W_{22}P_{22} \end{bmatrix}. \quad (8)$$

For this equation system,  $R_{11}$  is obtained by the sum of terms of the right hand of Eq.(5) except the terms  $[10(1-v_{11})^3v_{11}^2W_{11}P_{11}+10(1-v_{11})^2v_{11}^3W_{12}P_{12}] + [10(1-v_{11})^3v_{11}^2W_{21}P_{21} + 10(1-v_{11})^2v_{11}^3W_{22}P_{22}]$ . In the same manner,  $R_{21}$ ,  $R_{12}$  and  $R_{22}$  are obtained by substituting  $(u=u_{21}, v=v_{21})$ ,  $(u=u_{12}, v=v_{12})$ , and  $(u=u_{22}, v=v_{22})$ , in Eq.(5), respectively. By solving Eq.(8),  $W_{11}$ ,  $W_{21}$ ,  $W_{12}$ ,  $W_{22}$  are determined. Thus, the model has been completed by solving a 4x4 matrix and four 2x2 matrices. The model accuracy is elucidated by computing surface points via position  $(u, v)$ . For instance, the positions  $(u_{00}, v_{00})$ ,  $(u_{10}, v_{10})$ ,  $\dots$ ,  $(u_{33}, v_{33})$  are substituted in the network (5) and the results are  $P_{00}$ ,  $P_{10}$ ,  $\dots$ ,  $P_{33}$ , respectively. Thus, all control points have been interpolated. This means that the network passes through by all control points. Thus, the Bezier network (5) provides interpolation and preserves continuity. The model accuracy is calculated by means of relative error, which is computed by  $rms(\%)=rms \times 100/h_m$ . Here,  $rms$  is the root mean squared Eq.(17) and  $h_m$  is the mean of the surface  $P_{ij}$ . The surface representation accuracy is given by the sum of model error and measurement error. Thus, the surface shown in Fig.2 is represented with a relative error of 0.0017 %. The operations to compute the surface  $S(u,v)$  are deduced via network (5). For instance, the term  $(1+4v)(1-v)^4W_{10}P_{10}$  is calculated by one sum, one rest and seven multiplications. Thus, the sixteen terms of Eq.(5) are determined by 200 operations: 160 multiplications, 20 sums and 20 rests. This model is defined by the terms  $W_{00}P_{00}$ ,  $W_{01}P_{01}$ ,  $\dots$ ,  $W_{33}P_{33}$ , and the functions  $B_i(u)B_j(v)$  are the same for all segments. Thus, the network uses sixteen memory localities to save each surface model. Typically, the viability of a surface model is deduced via accuracy, speed and memory size [6], [24], [20]. To elucidate the viability of the Bezier network, it is examined based on the standard model NURBS. This method is described by means of the following expression

$$S(u,v) = \frac{N_0^3(u)N_0^3(v)W_{00}P_{00} + N_0^3(u)N_1^3(v)W_{01}P_{01} + \dots + N_3^3(u)N_3^3(v)W_{33}P_{33}}{N_0^3(u)N_0^3(v)W_{00} + N_0^3(u)N_1^3(v)W_{01} + \dots + N_3^3(u)N_3^3(v)W_{33}}. \quad (9)$$

For this equation,  $W_{ij}$  are the weights and  $N_i^p(u)$   $N_j^q(v)$  are the B-spline basis functions [4]. The weights  $W_{ij}$  are calculated by means of the knot vector, which adds points to move the surface toward the control points [8]. For a surface with degree  $(p, q)$ , there are at most  $(p+1) \times (q+1)$  non-zero functions to be computed

[15]. The operations to compute a surface point by NURBS are deduced via Eq.(9). Each B-spline produces an expression such as  $N_i^3(u)=(au^3+bu^2+cu+d)$ , which is calculated by 3 sums and 8 multiplications. Thus, the sixteen terms  $N_i^3(u)N_j^3(v)W_{ij}P_{ij}$  are determined by 415 operations: 111 sums and 304 multiplications. This result corresponds only to the numerator of Eq.(9), which is bigger than the 200 operations of the Bezier network. The NURBS accuracy indicates that the model does not passes through by all control points [25]. The accuracy reported by this model is a relative  $rms$  around 0.0030% [20]. The NURBS saves the model by means of sixteen memory localities for the terms  $W_{ij}P_{ij}$ , sixteen localities for  $W_{ij}$ , sixteen localities for the values  $u$  that produce  $N_i^p(u)$  and sixteen localities for the values  $v$  that produce  $N_j^q(v)$ . The above results indicate that the Bezier network improves the traditional surface models. For instance, the network improves the model accuracy. It is because the network passes through by all control and preserves continuity. Thus, the surface representation is achieved with a high accuracy. It is an improvement respect to NURBS, which do not interpolate all surface points. Also, the Bezier network generates the model in fast form by solving a 4x4 matrix and four 2x2 matrices. In the NURBS method, the model is generated via knot vector, which adds points to move the surface toward the control points. Furthermore, the network computes each surface point via 200 operations. Also, the Bezier network saves each segment model via sixteen memory localities whereas NURBS uses sixty four memory localities. Thus, the proposed method improves the accuracy, speed and memory size of the traditional surface model. The optical setup to perform the surface measurement is described in Section 3.

### 3. Optical setup for surface measurement

The surface representation includes two stages: surface measurement and model implementation. Therefore, the accuracy of the plantar surface representation is obtained by the sum of measurement error and model error. The surface points for the surface model are measured via line scanning. This procedure avoids external measurements to the vision system. Thus, the measurements  $(x, y, z)$  lead to obtain a high accuracy for the surface representation. It is because the measurement error is included in the accuracy of the plantar surface representation. The optical setup to measure the plantar surface is shown in Fig. 3. This arrangement includes a laser line, a CCD camera, a glass platform, an electromechanical device and a computer. In this setup, the  $x$ -axis and  $y$ -axis are located in the glass platform, which is perpendicular to  $z$ -axis. The geometry of line projection in  $x$ -axis is shown in Fig. 4(a), where  $f$  is the focal length,  $x_c$  is the image center,  $L$  is the distance between the laser line and the optical axis,  $D$  is the distance from the lens to the glass platform and the surface is indicated by  $h_{ij}$ .

The scanning of plantar surface is performed by moving the laser line in  $x$ -axis via electromechanical device, which provides the coordinate  $x$ . The coordinate  $y$  is deduced via line projection in  $y$ -axis, which is shown in Fig. 4(b), where the coordinate  $y$  is calculated by the equation  $y=(D+h_{ij})f/(y_j-y_c)\eta$ . In this expression  $\eta$  is the pixel size. The vision parameters  $D, f, L, x_c, y_c, \eta$  are calibrated in Section 4.

The surface  $z$  is determined via Fig.4(a) by  $h_{ij}=[fL/\eta(x_{ij}-x_c)]-D$ . This equation indicates that the surface  $h_{ij}$  is directly proportional to the line shifting  $s_{ij}$ , which is calculated by

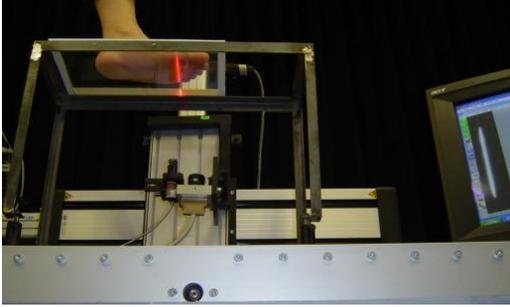


Figure 3. Optical setup to perform the scanning of foot sole

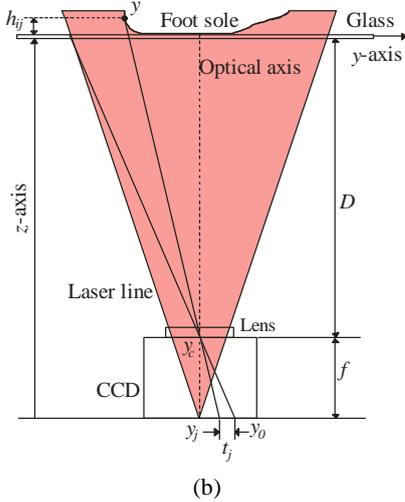
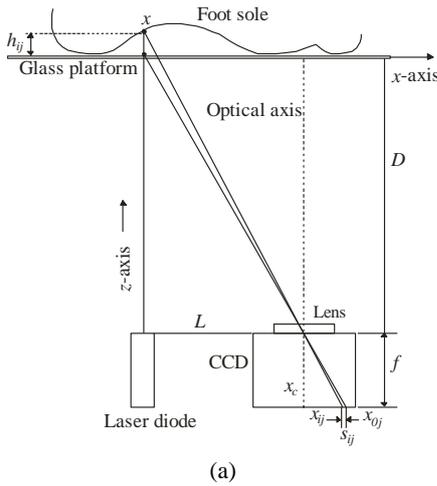


Figure 4. (a) Geometry of the line projection in  $x$ -axis.  
(b) Geometry of the line projection in  $y$ -axis

$$s_{ij} = x_{0j} - x_{ij}. \quad (10)$$

The line position  $x_{ij}$  is determined based on maximum intensity via Bezier curves [11]. To carry it out, a Bezier curve is built via pixel position  $x_i$  and pixel intensity  $I_i$  by the next two equations

$$x(u) = (1-u)^n x_0 + \binom{n}{1} (1-u)^{n-1} u x_1 + \binom{n}{2} (1-u)^{n-2} u^2 x_2 + \dots + u^n x_n, \quad 0 \leq u \leq 1, \quad (11)$$

$$I(u) = (1-u)^n I_0 + \binom{n}{1} (1-u)^{n-1} u I_1 + \binom{n}{2} (1-u)^{n-2} u^2 I_2 + \dots + u^n I_n, \quad 0 \leq u \leq 1. \quad (12)$$

By substituting the pixel position  $x_i$  in Eq.(11) and the pixel intensity  $I_i$  in Eq.(12), a concave curve is obtained. Thus, the line position is computed based on first derivative  $I'(u)=0$  via bisection method [9]. Then, value  $u$  where  $I'(u)=0$  is replaced in Eq.(11) to obtain the position  $x(u)$ . And the position  $x_{ij}=x(u)$  is replaced in Eq.(10) to compute the line shifting  $s_{ij}$ .

The surface  $h_{ij}$  is computed by a Bezier network [10], which is described by the next expression

$$\mathcal{C}(u, v) = \sum_{i=0}^n \sum_{j=0}^m w_{ij} h_{ij} B_i(u) B_j(v), \quad (13)$$

$$0 \leq u \leq 1, \quad 0 \leq v \leq 1.$$

This equation is applied to compute the control points, which are used to construct the surface model. Therefore, the notation of Eq.(13) is different to the notation of Eq.(1). In this case,  $B_i(u)$  and  $B_j(v)$  are related with  $s_{ij}$  and  $y_j$ , respectively. Thus, the line shifting is converted to a value  $u$  by means of the expression

$$u = a_0 + a_1 s_{ij}. \quad (14)$$

For this equation, the values  $a_0$  and  $a_1$  are obtained by solving the equations  $a_0 + a_1 s_{00} = 0$  and  $a_0 + a_1 s_{nm} = 1$ . The coordinate  $y_j$  is converted to a value  $v$  by

$$v = b_0 + b_1 y_j. \quad (15)$$

For this equation, the values  $b_0$  and  $b_1$  are obtained by solving the equations  $b_0 + b_1 y_0 = 0$  and  $b_0 + b_1 y_n = 1$ . Then, the values  $h_{ij}$  are substituted in Eq.(13) to obtain the next network

$$\mathcal{C}(u, v) = w_{00} h_{00} B_0(u) B_0(v) + w_{01} h_{01} B_0(u) B_1(v) + \dots + w_{1m} h_{1m} B_1(u) B_m(v) + \dots + w_{nm} h_{nm} B_n(u) B_m(v). \quad (16)$$

For this equation, the weights  $w_{ij}$  are computed by substituting the values  $(u, v)$  of each  $h_{ij}$  in Eq.(16) to obtain an equation system [22]. Here, the procedure to obtain Eq.(6) to Eq.(8) is applied. By solving the equation system, the weights  $w_{ij}$  are obtained to complete the network  $\mathcal{C}(u, v)$ . This network is applied to compute the surface from the line shown in Fig. 5(a). In this procedure, the line shifting  $s_{ij}$  is detected in each coordinate  $y_j$ . Then, the values  $(u, v)$  of the shifting are replaced in Eq.(16) to calculate the

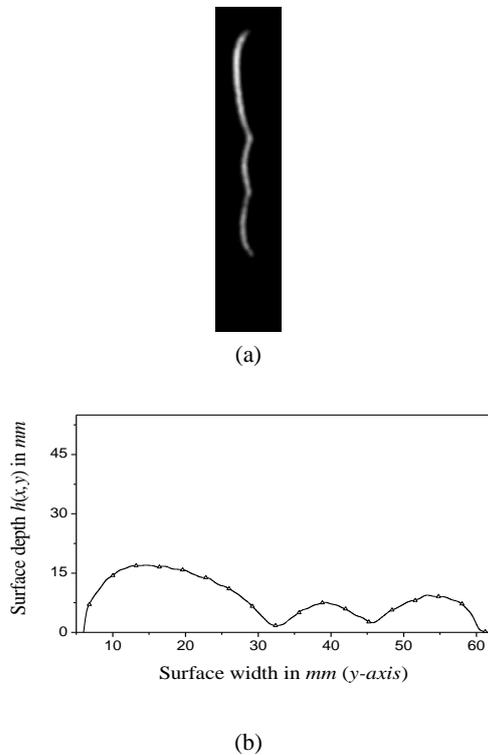
surface shown in Fig. 5(b), where the symbol " $\Delta$ " represents the data measured by a coordinate measure machine (CMM). The measurement accuracy provided by the network (16) is determined via root mean squared. This *rms* value [23] is computed by the next expression

$$rms = \sqrt{\frac{1}{n} \sum_{i=1}^n (ho_i - hc_i)^2}, \quad (17)$$

For this equation,  $ho_i$  is the surface data  $z_{ij}$  measured by a coordinate measure machine (CMM),  $hc_i$  is the surface data  $z_{ij} = \mathcal{C}(u, v)$  calculated by the network (16) and  $n$  is the number of points  $z_{ij}$ . The error of the slider position is  $0.0014 \text{ mm}$ , which is added to Eq.(17) to obtain a  $rms = 0.1024 \text{ mm}$  for the profile shown in Fig. 5(b). Thus, the network provides the measurement of plantar surface as  $z_{ij} = \mathcal{C}(u, v)$ . The calibration to determine the coordinate  $y$  is described in Section 4.

#### 4. Calibration of vision parameters

The calibration of vision parameters is performed inside of the vision system to complete the coordinates  $(x, y, z)$  to construct the surface model. Thus, errors of external measurements to the vision system are avoided. The traditional models perform an external calibration to determine the vision parameters. The external calibration is performed outside of the vision system via measurement of external references [12]. This external calibration transforms the world coordinates



**Figure 5.** (a) Laser line projected onto the plantar surface. (b) Surface depth computed by the network from the laser line

$P_w = (x_w, y_w, z_w)$  to the image coordinates  $(X_u, Y_u)$  by means of the expression  $P_c = \mathbf{R} \cdot P_w + \mathbf{t}$ , where  $P_c = (x_c, y_c, z_c)$  are the camera coordinates,  $\mathbf{R}$  is the rotation matrix and  $\mathbf{t}$  is the translation vector. This calibration is suitable for a static setup. But, the calibration is not achieved when the setup is modified in the vision process. It is due to the lack of references  $(x_w, y_w, z_w)$  during the vision task. In the proposed setup, the vision parameters are modified when the vision system is moved in  $z$ -axis. Therefore, the calibration is performed via mobile setup, where the coordinate  $x$  is provided by the electromechanical device and the coordinate  $z$  is given by Eq.(16). The coordinate  $y = f(D + h_{ij}) / (y_j - y_c) \eta$  is determined via calibration of the parameters  $D, f, y_c$  and  $\eta$ . Based on Fig.6(a), the radial distortion is deduced via line position  $x_{ij} = (X_{ij} + \delta x_{ij})$ , where  $X_{ij}$  is the distorted position and  $\delta x_{ij}$  is the distortion. Thus, the undistorted shifting is given by the equation  $s_{ij} = (X_{0j} + \delta x_{0j}) - (X_{ij} + \delta x_{ij})$  and the distorted shifting is given by the equation  $S_{ij} = (X_{0j} - X_{ij})$ . The distortion is deduced by the equation  $\delta x_{ij} = (X_{0j} - X_{ij}) + \delta x_{0j} - s_{ij} = S_{ij} - s_{ij} + \delta x_{0j}$ . In this case, the first line shifting is defined without distortion, therefore,  $\delta x_{0j} = 0$ ,  $\delta x_{1j} = 0$  and  $s_{1j} = S_{1j}$ . Therefore, the undistorted shifting  $s_{ij}$  is represented based on the first distorted shifting  $S_{1j}$  by means of the expression  $s_{ij} = i * S_{1j}$ . Thus, the distortion in  $x$ -axis is calculated by  $\delta x_{ij} = i * S_{1j} - (X_{0j} - X_{ij})$  for  $i = 1, 2, 3, \dots, n$  and  $j = 1, 2, 3, \dots, m$ . The same procedure is performed to determine the distortion  $\delta y_j$  based on Fig. 6(b) via equations  $t_j = (y_0 - y_j)$  and  $T_j = (Y_0 - Y_j)$ . Thus, the distortion in  $y$ -axis is computed via equation  $\delta y_j = j * T_j - (Y_0 - Y_j)$ . The image plane of the camera is placed parallel to the glass platform shown in Fig. 3. To corroborate that the image plane is parallel to the glass window in  $x$ -axis, the projection  $k_i$  of Fig. 6(a) is calculated. This projection is computed by the next expression

$$k_i = \frac{\eta(x_{ij} - x_c)h_{ij}}{f} = \frac{\eta[(x_{0j} - x_c) - s_{ij}]h_{ij}}{f}. \quad (18)$$

For this equation,  $f, x_c, \eta$  are constants and  $h_{ij}$  is computed via  $x_{ij}$ . Thus, the expression  $k_i$  given by Eq.(18) is a function of first order and the derivative  $dk/ds$  is a constant. To corroborate that the image plane is parallel to the glass platform in  $y$ -axis, the projection  $q_j$  of Fig. 6(b) is calculated. This projection is computed based on  $t_j = \eta(y_0 - y_c) - \eta(y_j - y_c)$  by means of the next expression

$$q_j = \frac{\eta(y_j - y_c)h_{ij}}{f} = \frac{\eta[(y_0 - y_c) - t_j]h_{ij}}{f}. \quad (19)$$

For this equation  $y_c, y_0$  are constants and Eq.(19) is a function of first order. Therefore, the derivative  $dq/dt$  is a constant. Thus, the image plane is parallel to glass platform when the derivatives  $dk/ds$  and  $dq/dt$  are a constant.

The image centre is deduced via Fig. 6(a) by the equation  $L_j = \eta(x_{ij} - x_c)(D + h_{ij})$ , which generates the sequence  $\eta(x_{1j} - x_c)(D + h_{1j}) = \eta(x_{2j} - x_c)(D + h_{2j}) = \eta(x_{3j} - x_c)(D + h_{3j})$  and the expression  $[h_{2j}(x_{2j} - x_c) - h_{1j}(x_{1j} - x_c)]$

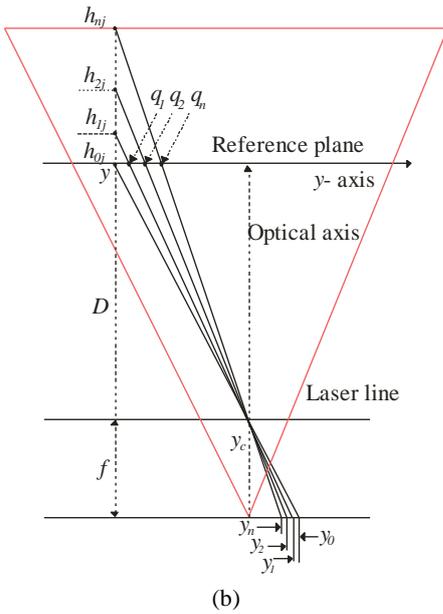
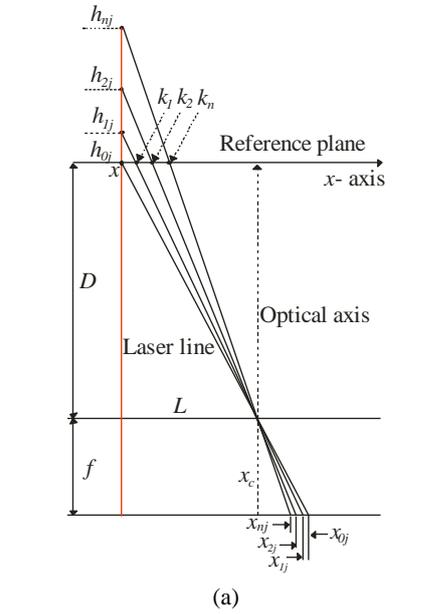
$x_c)(x_{1j}-x_{3j})=[h_{3j}(x_{3j}-x_c)-h_{1j}(x_{1j}-x_c)](x_{1j}-x_{2j})$ . Thus, the image centre is defined by

$$x_c = \frac{(h_{1j}x_{1j} - h_{2j}x_{2j})(x_{1j} - x_{3j}) + (h_{3j}x_{3j} - h_{1j}x_{1j})(x_{1j} - x_{2j})}{(h_{1j} - h_{2j})(x_{1j} - x_{3j}) + (h_{3j} - h_{1j})(x_{1j} - x_{2j})} \quad (20)$$

For this equation,  $h_{ij}$  is calculated by the network via  $s_{ij}$ . Then, the distance  $D$  is determined by

$$D = \frac{h_{2j}(x_{2j} - x_c) - h_{1j}(x_{1j} - x_c)}{(x_{1j} - x_{2j})} \quad (21)$$

The focal length is deduced via Fig. 7 by moving the camera lens from  $f$  to  $f_1$ . Thus, the line is moved from  $x_{ij}$  to  $\alpha x_{ij}$  and the distance  $D_1$  is calculated by



**Figure 6.** (a) Geometry of the optical setup in x-axis. (b) Geometry of the optical setup in y-axis

$D_1=h_{ij}(\alpha x_{ij}-x_c)/(\alpha x_{ij}-\alpha x_{0j})$ . Then,  $fL=D(x_{0j}-x_c)\eta$  and  $f_1L=D_1(\alpha x_{0j}-x_c)\eta$  are deduced from Fig. 6(a) and Fig. 7, respectively. From these terms and the equation  $D+f=D_1+f_1$ , the focal length is computed by

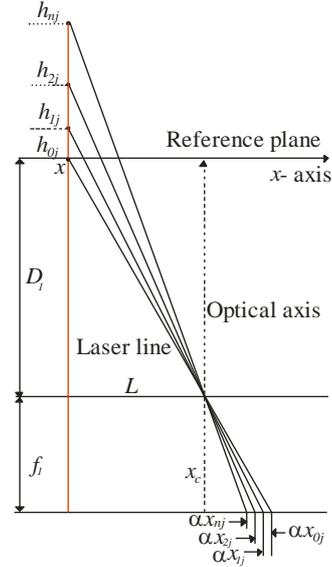
$$f = \frac{(D_1 - D)}{1 - D_1(\alpha x_{0j} - x_c) / D (x_{0j} - x_c)} \quad (22)$$

$$f_1 = D + f - D_1 \quad (23)$$

Thus, the focal length  $f$  and its modification  $f_1$  are obtained. From Fig. 6(a), the pixel size  $\eta$  is deduced via distances  $a^2=(x_{0j}-x_c)^2\eta^2+f^2$ ,  $b^2=[D(x_{0j}-x_c)\eta/f]^2+D^2$  and  $c^2=[D(x_{0j}-x_c)\eta/f_0+(x_{0j}-x_c)\eta]^2+(D+f)^2$ . From these distances and the relationship  $c=a+b$ , the next equation is obtained

$$D(x_{0j} - x_c)^2 \eta^2 + Df^2 - \sqrt{\frac{D^2}{f^2} (x_{0j} - x_c)^2 + D^2} = 0 \quad (24)$$

This equation is solved by the bisection method via interval  $0 \leq \eta \leq 1$  to obtain  $\eta$ . Then, the coordinate  $y_c$  is deduced based on Fig. 6(b) by solving the equations:  $t_1 = \eta(y_0 - y_c) - \eta(y_1 - y_c)$  and  $t_2 = \eta(y_0 - y_c) - \eta(y_2 - y_c)$ . From this calibration, the coordinate  $y = (D + h_{ij})f / (y_j - y_c)\eta$  is computed to complete the coordinates  $(x, y, z)$  of the plantar surface. The surface modeling is described in Section 5.



**Figure 7.** Geometry of the optical setup at  $f_1$  in x-axis

### 5. Plantar surface modeling

The plantar surface model is constructed via coordinates  $(x, y, z)$ . The slider machine provides the coordinate  $x$  where the line is captured. From the line shifting values  $(u, v)$ , the network in Eq.(16) calculates the surface  $z = \mathcal{S}(u, v)$ . The coordinate  $y = (D + h_{ij})f / (y_j - y_c)\eta$  is determined based on line position  $y_j$ . From all images, the network  $\mathcal{S}(u, v)$  computes the plantar surface shown in Fig. 8(a). The accuracy of this measurement is a relative  $rms(\%) = 0.482 \%$ . This

values is calculated by the expression  $rms(\%) = rms \times 100 / h_m$ , where the  $rms = 0.1014 \text{ mm}$  and  $n = 400$ . Then, the surface is divided in  $4 \times 4$  segments to construct the model. This procedure is performed via points shown in Fig. 8(b), which represent the heel area. Thus, the surface positions  $(x, y)$  of each segment are converted to values  $(u, v)$  by the following equations

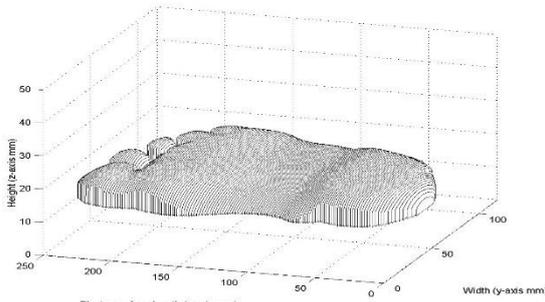
$$u = (x - p) / T, \quad p = \text{int}(x / T) * T, \quad 0 \leq u \leq 1, \quad (25)$$

$$v = (y - q) / T, \quad q = \text{int}(y / T) * T, \quad 0 \leq v \leq 1. \quad (26)$$

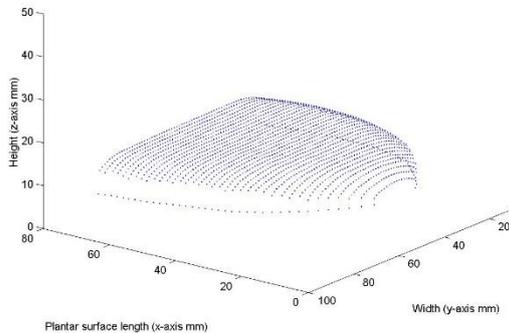
For these equations,  $T = 3$  is the period of each segment and the values  $(p, q)$  are the period number. Then, the control points  $P_{ij} = h_{ij}$  are replaced in Eq.(5) to build the model

$$S(u, v) = (1 + 4u)(1 - u)^4(1 + 4v)(1 - v)^4 W_{00} P_{00} + \dots + 10(1 - u)^3 u^2(1 + 4v)(1 - v)^4 W_{10} P_{10} + \dots + (5 - 4u)u^4(5 - 4v)v^4 W_{33} P_{33}. \quad (27)$$

For this equation, the weights  $W_{01}, W_{02}, W_{10}, W_{20}, W_{13}, W_{23}, W_{31}, W_{32}$  are computed based on the procedure described in Eqs. (6) and (7). These weights are put in Eq.(27) to obtain Eq.(8), which is solved to obtain  $W_{11}, W_{21}, W_{12}, W_{22}$ . Thus, the model of each segment is generated via sixteen terms  $W_{ij} h_{ij}$ . Therefore, plantar surface model is a matrix, which contains the terms  $W_{ij} h_{ij}$  of all surface segments. The steps to construct the surface model are indicated in



(a)



(b)

**Figure 8.** (a) Plantar surface retrieved via network  $\mathcal{L}(u, v)$ . (b) Surface points to collect  $4 \times 4$  segments

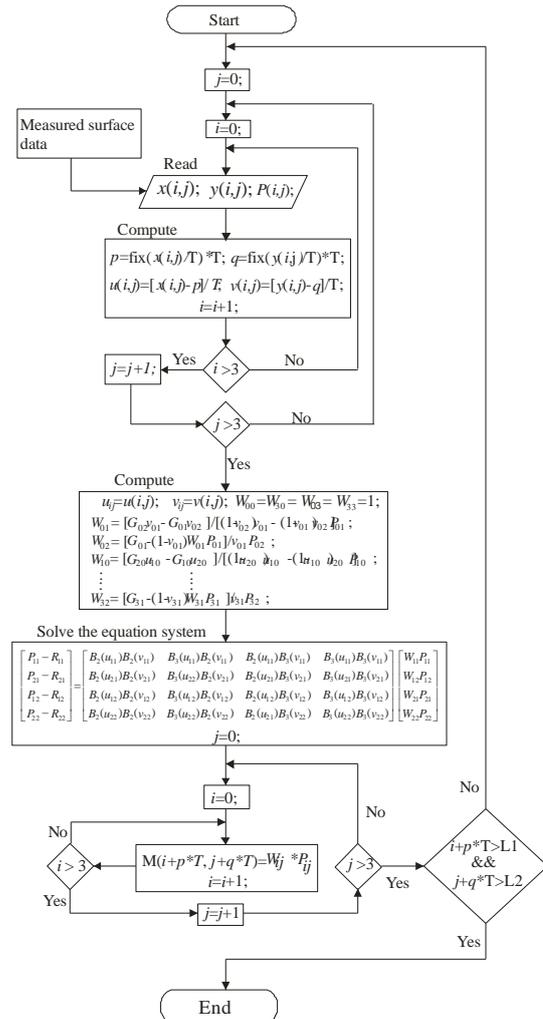
the flow chart shown in Fig. 9. In this flow chart, the values  $L_1$  and  $L_2$  are the length and the width of the measured surface data, respectively. The function  $\text{fix}()$  provides the integer part of a division.

The plantar surface model is defined by the matrix  $M(i, j)$ , which stores all terms  $W_{ij} * P_{ij}$ . The terms  $G_{ij}$  in the flow chart correspond to  $G_{01} = [P_{01} - (1 + 4v_{01}) (1 - v_{01})^4 P_{00} - (5 - 4v_{01}) v_{01}^4 P_{03}] / 10(1 - v_{01})^2 v_{01}^2$ ,  $G_{10} = [P_{10} - (1 + 4u_{10})(1 - u_{10})^4 P_{00} - (5 - 4u_{10})u_{10}^4 P_{30}] / 10(1 - u_{10})^2 u_{10}^2$ , ...,  $G_{32} = [P_{32} - (1 + 4v_{32})(1 - v_{32})^4 P_{03} - (5 - 4v_{32})v_{32}^4 P_{33}] / 10(1 - v_{32})^2 u_{32}^2$ . The surface model determines the surface  $z = S(u, v)$  via position  $(x, y)$ . Also, the network provides equations for the surface position  $(x, y)$  via next expressions

$$x(u, v) = \mathcal{W}_{00} x_{00} B_0(u) B_0(v) + \mathcal{W}_{01} x_{01} B_0(u) B_1(v) + \dots + \mathcal{W}_{03} x_{03} B_0(u) B_3(v) + \dots + \mathcal{W}_{33} x_{33} B_3(u) B_3(v), \quad (28)$$

$$y(u, v) = \mathcal{W}_{00} y_{00} B_0(u) B_0(v) + \mathcal{W}_{01} y_{01} B_0(u) B_1(v) + \dots + \mathcal{W}_{03} y_{03} B_0(u) B_3(v) + \dots + \mathcal{W}_{33} y_{33} B_3(u) B_3(v). \quad (29)$$

For these equations, the procedure to determine the weights  $W_{ij}$  is applied to compute the weights  $\mathcal{W}_{ij}$  and  $\mathcal{W}_{ij}$ . Thus, Eq.(28) and Eq.(29) calculate the position  $x = x(u, v)$  and  $y = y(u, v)$ .



**Figure 9.** Flow chart to construct the surface model

To represent the surface  $(x, y, z)$ , the networks Eq.(27), Eq.(28) and Eq.(29) are evaluated for each segment in the intervals  $0 \leq u \leq 1$  and  $0 \leq v \leq 1$ . In the surface edge, the period  $T$  varies in  $x$ -axis and  $y$ -axis according to the segment points. Thus, plantar surface shown in Fig. 10 is computed by  $z=S(u, v)$ ,  $x=\mathcal{A}(u, v)$  and  $y = \mathcal{C}(u, v)$ . The accuracy of this surface is determined via  $rms$  Eq.(17), where  $ho_i$  is provided by  $\mathcal{A}(u, v)$ ,  $hc_i$  is the surface computed by  $S(u, v)$  and  $n= 400$ . Thus, an  $rms= 0.000478 \text{ mm}$  is obtained and the relative error is  $rms(\%)=0.0022\%$  based on  $h_m=22.08 \text{ mm}$ . The procedure to compute object surface via network (5) is described by the algorithm shown in Fig. 11. The algorithm to construct the surface model and the algorithm to compute the object surface are implemented in C programming language.

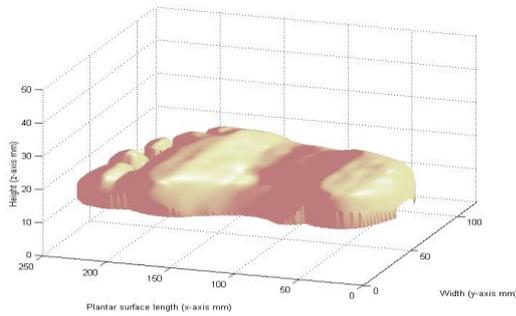


Figure 10. Plantar surface generated by the network  $S(u, v)$

The surface model provides the data to adjust the shoe-last bottom shown to the plantar surface. To carry it out, the shoe-last bottom shown in Fig. 12(a) is scanned via setup shown in Fig. 4. In this procedure, the line shifting  $s_{ij}$  is detected via Eq.(10), Eq.(11) and Eq.(12). Then, the shifting is converted to values  $(u, v)$ , which are replaced in the network Eq.(16) to compute the surface  $z=\mathcal{C}(u, v)$ .

The coordinate  $x$  is provided by the slider device,

```

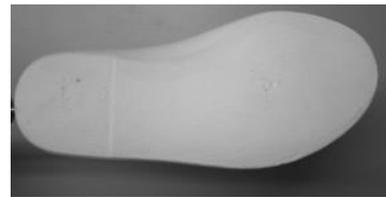
Define T=3, L1, L2, r, s, t, E[1]=E[2]=A[1]=A[2]=10;
for p=0 to L1
  for q=0 to L2
    u=0;
    for k=0 to r
      v=0;
      for t=0 to s
        Suv=(k+r*p,t+s*q)=0;
        E[0]=1+4*v; E[3]=5-4*v; A[0]=1+4*u; A[3]=5-4*u;
        for i=0 to 3
          SUM=0;
          for j=0 to 3
            SUM=SUM+E[j]*pow(1-v,4-fix(j/3))*
              pow(v,2*j+fix(2*j/3))* W(i+q*T,j+p*T)* F(i+q*T,j+p*T);
          end
          SUM=A[j]*pow(1-u,4-i-fix(i/3))*pow(u,2*i+fix(2*i/3))*SUM;
          Suv(k+r*p,t+s*q)=Suv(k+r*p,t+s*q)+SUM;
        end
      end
      v=v+1/s;
    end
    u=u+1/r;
  end
end
end
end

```

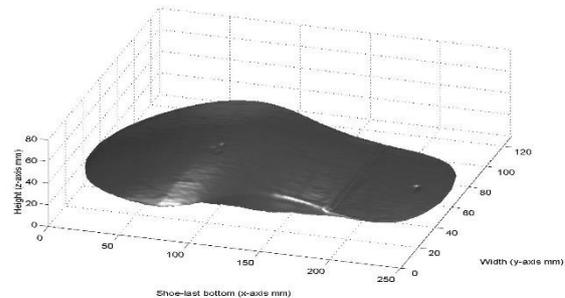
Figure 11. Algorithm to compute the object surface  $S(u, v)$

whereas the coordinate  $y$  is computed via equation  $y=(D+h_{ij})f(y_j-y_c)\eta$ . Then, the model of the shoe-last bottom is constructed via Eq.(27), Eq.(28), Eq.(29). These equations are evaluated in the intervals  $0 \leq u \leq 1$  and  $0 \leq v \leq 1$  to obtain each surface segment of shoe-last bottom shown in Fig. 12(b). Then, the morphological parameters are deduced from the diagram shown in Fig. 13(a) [11]. The foot length is the distance between the heel point  $A$  and the maximum point  $B$  in  $x$ -axis.

The center line is the line passing the center heel  $A$  and the second metatarsal  $C$ . The foot width is the distance between the points  $E$  and  $D$  in  $y$ -axis. The mid foot width is the breadth at  $1/2$  of the foot length in  $x$ -axis. The heel width is the maximum width in the heel area. The positions of first and fifth metatarsal heads are pointed by  $D$  and  $J$ , respectively. Thus, the surface is moved toward the plantar surface in  $x$ -axis and  $y$ -axis. To carry it out, the maximum plantar surface  $d_y$  outside of shoe-last bottom is detected via equation  $d_y=y_s-y_f$ , where  $y_f$  is the contour position of the plantar surface in  $y$ -axis and  $y_s$  is the contour position of the shoe-last bottom. Then, the contour of the shoe-last bottom is magnified based on the scale factor  $\varepsilon=(y_s+d_y)/y_s$ . Thus, the surface position  $(x, y)$  is re-computed via equations  $x=x\varepsilon$  and  $y=y\varepsilon$ . Then, the position of the morphological parameters of the shoe-last bottom is moved toward the morphological parameters of the foot sole. These two steps are shown in Fig. 13(b) by the green dash line. Then, the contour is smoothed to obtain continuity  $G^1$  via derivatives of the neighbor points. The contour is smoothed in  $y$ -axis by the expression  $y_{ij}=(y_{i-1,j}+y_{i+1,j})/2$ , where the indexes  $(i, j)$  correspond to the row and column number, respectively. The smoothed contour is shown in Fig. 13(b) by the black continuous line. Based on these steps, the

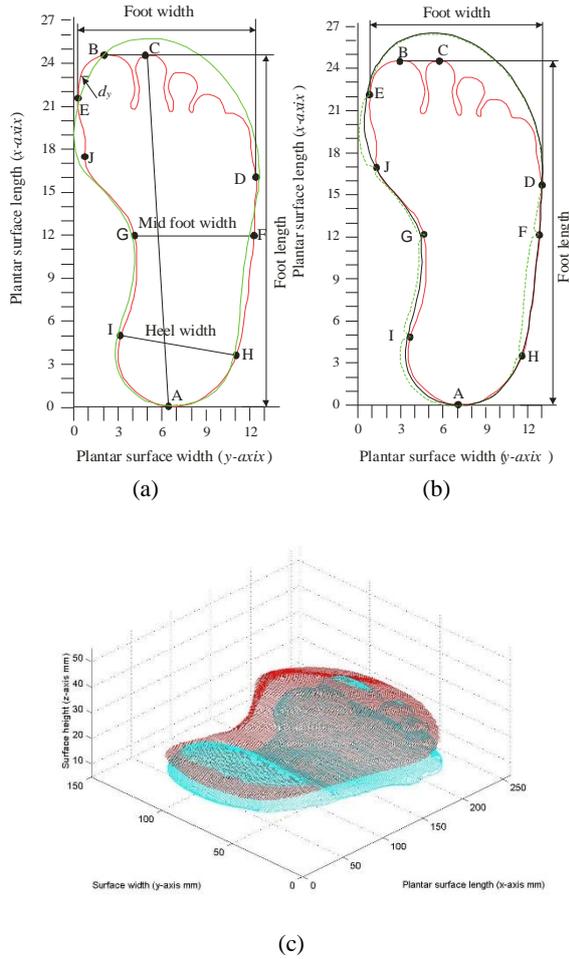


(a)



(b)

Figure 12. (a) Shoe-last bottom to be adjusted to the plantar surface. (b) Surface of the shoe-last bottom represented by the network  $S(u, v)$

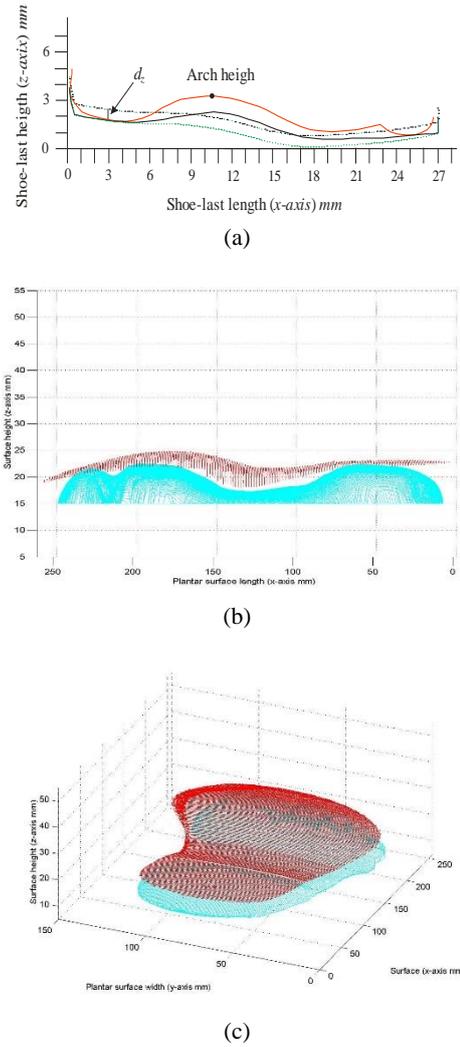


**Figure 13.** (a) Morphological parameters of the foot sole. (b) Matching of the shoe-last bottom with the morphological parameters. (c) Adjusted shoe-last bottom in  $x$ -axis and  $y$ -axis

shoe-last bottom is adjusted in  $x$ -axis and  $y$ -axis. In this procedure, the position  $(x, y)$  is magnified by the equations  $x=x\epsilon$  and  $y=y\epsilon$ . Then, the contour is matched and smoothed with the morphological parameters. The adjustment of the shoe-last bottom in  $x$ -axis and  $y$ -axis is shown in Fig. 13(c).

The adjustment in  $z$ -axis is performed based on the diagram shown in Fig. 14(a), where the surface  $d_z$  outside of the contour is determined by  $d_z=z_f-z_s$ . For this equation  $z_f$  is the plantar surface position and  $z_s$  is the position of the shoe-last bottom. Then, the contour of the shoe-last bottom is magnified in  $z$ -axis via scale factor  $\zeta=(z_s+d_z)/z_s$ . Thus, the surface is re-computed by  $h_{ij}=h_{ij}\zeta$ . The result of this process is shown in Fig. 14(a) by the green dash line. Then, the arch height is replaced in the surface of the shoe-last bottom, which is smoothed by means of the equation  $h_{ij}=(h_{ij}+h_{i+1,j}+h_{i,j+1}+h_{i+1,j+1})/4$ . This adjustment is shown by the continuous black line in Fig. 14(a). Thus, the adjustment in  $z$ -axis is performed by detecting  $d_z$  to magnify the surface via expression  $h_{ij}=h_{ij}\zeta$ .

Then, the surface of the shoe-last bottom is match-



**Figure 14.** (a) Diagram to adjust the plantar surface in  $z$ -axis. (b) Adjusted surface in  $z$ -axis. (c) Adjusted shoe-last bottom in  $x$ -axis,  $y$ -axis and  $z$ -axis

ed with arch height. Thus, the shoe-last bottom is smoothed based on neighbor points and the result is shown in Fig. 14(b). Thus, the adjustment of the shoe-last bottom has been completed and it is shown in Fig. 14(c). The evaluation of the plantar surface model is described in Section 6.

## 6. Evaluation of the plantar surface model

The plantar surface model has been constructed via Bezier networks. This model interpolates all control points and preserves continuity. Moreover, the calibration does not add errors to the surface representation. It is because the vision parameters are determined based on the laser line. Thus, surface representation error is minor to 1%.

The surface representation uncertainty  $u$  is deduced by the measurement uncertainty  $u_n$  and the model uncertainty  $u_M$ . Based on the propagation law, the measurement uncertainty is deduced via line position  $(u, v)$  and the focal length  $f_l$  by the following expression

$$\mathbf{u}_h^2 = \left( \frac{\partial h}{\partial u} \right)^2 \mathbf{u}_u^2 + \left( \frac{\partial h}{\partial v} \right)^2 \mathbf{u}_v^2 + \left( \frac{\partial h}{\partial f} \right)^2 \mathbf{u}_{f1}^2, \quad (30)$$

where  $\partial h/\partial u$ ,  $\partial h/\partial v$ ,  $\partial h/\partial f$  are the sensitivity coefficients. The uncertainty  $\mathbf{u}_v^2 = (\partial v/\partial y_j)^2 (\mathbf{u}_y)^2$  is deduced via Eq.(15) and  $\mathbf{u}_y = 0.2570$  pixels is computed by the standard deviation of sixteen measurements of the laser line in  $y$ -axis. Thus, the sensitivity coefficient is  $\partial v/\partial y_j = b_f = 0.2282$   $1/\text{pixel}$  and the uncertainty is  $\mathbf{u}_v = 0.0585$ . The uncertainty  $\mathbf{u}_s^2 = (\partial u/\partial s)^2 (\mathbf{u}_s)^2$  is deduced via Eq.(14), where  $\mathbf{u}_s = 0.2688$  pixels is the standard deviation of sixteen measurements of the line shifting in  $x$ -axis. Thus, the sensitivity coefficient is  $\partial u/\partial s = 0.1714$   $1/\text{pixel}$  and  $\mathbf{u}_s = 0.0461$ . The uncertainty  $\mathbf{u}_{f1}^2 = (\partial f/\partial f)^2 (\mathbf{u}_f)^2$  is determined via Eq.(23), where  $(\mathbf{u}_f)^2 = (\partial f/\partial x_i)^2 (\mathbf{u}_{xi})^2$  and  $\mathbf{u}_{xi} = 0.328$  pixels is the standard deviation of sixteen measurements  $x_{ij}$ . The sensitivity coefficients are  $\partial f/\partial x_{ij} = 0.1032$   $1/\text{pixel}$ ,  $\partial f/\partial f = 0.3442$   $1/\text{pixel}$ . Thus, the uncertainties are  $\mathbf{u}_f = 0.0432$  and  $\mathbf{u}_{f1} = 0.0142$ . Then, the sensitivity coefficients  $\partial h/\partial u = 0.5253 \text{ mm}$ ,  $\partial h/\partial v = 0.5394 \text{ mm}$  and  $\partial h/\partial f_1 = 0.5234 \text{ mm}$  are computed based on the network Eq.(16). These values are substituted in Eq.(30) to calculate the measurement uncertainty  $\mathbf{u}_h = 0.04065 \text{ mm}$ . Then, the model uncertainty  $\mathbf{u}_M$  is determined via surface position  $(u, v)$  by the following expression

$$\mathbf{u}_M^2 = \left( \frac{\partial S}{\partial u} \right)^2 \mathbf{u}_u^2 + \left( \frac{\partial S}{\partial v} \right)^2 \mathbf{u}_v^2. \quad (31)$$

The uncertainty  $\mathbf{u}_v^2 = (\partial v/\partial y_j)^2 (\mathbf{u}_y)^2$  is deduced via Eq.(26), where  $\mathbf{u}_y = 0.2570$  pixels is the standard deviation of sixteen measurements of the laser line in  $y$ -axis. Thus, the sensitivity coefficient is  $\partial v/\partial y_j = b_f = 0.3142$   $1/\text{pixel}$  and the uncertainty is  $\mathbf{u}_v = 0.0807$ . The uncertainty  $\mathbf{u}_u$  is deduced by moving the laser line via slider device from a reference  $x_0$ . Based on the displacement  $s = \eta(x - x_0)$ , the uncertainty is defined by  $\mathbf{u}_s^2 = (\partial s/\partial x)^2 (\mathbf{u}_x)^2$ , where  $\mathbf{u}_x = 0.2868$  pixels is the standard deviation of sixteen displacements. Thus, sensitivity coefficient is  $\partial s/\partial x = 0.1063 \text{ mm/pixel}$  and  $\mathbf{u}_s = 0.03056 \text{ mm}$ . The sensitivity coefficients  $\partial S/\partial u = 0.4321 \text{ mm}$  and  $\partial S/\partial v = 0.4897 \text{ mm}$  are calculated via network Eq.(27). These values are substituted in Eq.(31) to compute the model uncertainty  $\mathbf{u}_M = 0.04166 \text{ mm}$ . Then, the uncertainty  $\mathbf{u}$  is determined by  $\mathbf{u} = (\mathbf{u}_h^2 + \mathbf{u}_M^2)^{1/2}$  [18]. Thus, the uncertainty of the surface representation is  $\mathbf{u} = 0.0624 \text{ mm}$ .

The proposed model preserves both interpolation and continuity. Thus, a contribution is achieved to improve accuracy, speed and memory size of the plantar surface representation. Typically, B-splines and NURBS do not interpolate to preserve continuity. Moreover, the interpolation provided by these methods introduces discontinuities near of the interpolated points [5], [14]. The interpolation and continuity has a great influence in the surface representation accuracy. For instance, the network interpolates all control points by solving a small equation system.

Thus, a high accuracy for surface representation is achieved. Also, the model improves the speed to compute each surface point via 200 operations. This number of operations is minor than the number of operations performed by the traditional methods. Moreover, the method saves the terms  $W_{ij}P_{ij}$  of each surface segment by means of via sixteen memory localities. The contribution of the proposed surface modeling is elucidated by an evaluation based on model accuracy, representation accuracy, number of operations, memory size and consumed time. The evaluated methods are Least squared, B-Splines and NURBS, which are indicated in the first column of Table 1. These methods construct the plantar surface model. The accuracy of each model is determined via relative *rms* and it is shown in the second column in millimeters and in percentage. The model accuracy of the traditional methods is a relative *rms* over 0.031%. This accuracy is reported by the traditional methods in the references [13], [24], [27]. The surface representation accuracy is calculated by the sum of the model error and the measurement error. The representation accuracy of the traditional methods is a relative *rms* over 0.86%, which is shown in the third column. To represent the plantar surface shown in Fig.10, 27840 points are computed. To calculate these points, the Bezier networks perform  $(27840 \times 200) = 5568000$  operations and B-splines perform  $(27840 \times 415) = 11553600$  operations. The fourth column shows the number of operations to represent the plantar surface. The Bezier network uses 160 bytes to save each segment model via sixteen localities. The plantar surface model is composed by 870 segments, which lead to use 139200 memory bytes. The B-splines use 480 bytes to save each segment model by means of forty-eight memory localities for the terms  $N_i^p(u)$ ,  $N_j^q(v)$  and  $W_{ij}P_{ij}$ . Thus, the B-splines consume 556800 memory bytes to save the plantar surface model. The memory size used to save the plantar surface model is shown in the fifth column. The time taken to represent the plantar surface includes surface measurement, model implementation and surface computation. The time is shown in the sixth column. The results presented in Table 1 indicate that the Bezier network improves the model accuracy and surface representation accuracy. It is because the model interpolates all surface points and preserves continuity. Also, the speed and memory size have been improved via Bezier network. These results provide a contribution of the proposed method in the plantar surface modeling. For instance, the Bezier networks represent the plantar surface with a relative *rms* = 0.4842% and reduce the number of operations and memory size. Thus, that the model of the plantar surface of the traditional methods has been improved. The improvement is corroborated by the results presented in Table 1.

The computer used to perform the surface modeling is a PC of 1.8 GHz. In this computer, the surface modeling is implemented in C programming language. The slider moves the vision system at 34 steps per

**Table 1.** Evaluation of the plantar surface model.

Method	Model accuracy. Relative rms %	Accuracy of the plantar surface representation. Relative rms %	Operations to represent the plantar surface.	Memory size to save the plantar surface model. Bytes	Consumed time to represent the plantar surface. Seconds
Least squared	0.325mm (1.46 %)	1.135 mm (5.09%)	16213512	421472	480 s
B-Splines	0.007mm (0.031%)	0.193 mm (0.86%)	11553600	380640	41.5 s
NURBS	0.009mm (0.042%)	0.218mm (0.97%)	24053760	486400	72.8 s
Bezier Networks	0.0004mm 0.0022%	0.106 mm (0.464%)	5568000	139200	14.51 s

second. The frame rate of the camera is 34 fps. Each image of laser line is captured via language C based on XCLIB software library of EPIX. Each laser line image is processed by the network in 0.010 sec. The step resolution of the slider machine is 0.024 mm. This machine has a position error of 0.0014 mm. This error is added to the network  $\mathcal{C}(u,v)$ , which provides the surface measurement. Thus, a high accuracy for surface representation is achieved via Bezier networks. Thus, the plantar surface modeling is performed in good manner.

## 7. Conclusions

A technique to construct the plantar surface model via Bezier networks has been presented. The proposed technique provides a valuable tool for the examination of the plantar surface via computational models. This technique avoids measurements on the setup, as is common in the optical methods. The vision parameters are obtained automatically by computational process to improve representation accuracy. It is because the errors of external measurement to the vision system are not added to the surface representation. Also, the proposed model reduces the number of operations and memory size to represent the plantar surface. Thus, computational-optical setup constructs the model of the plantar surface, which provides morphological parameters to perform the adjustment of the shoe-last bottom. Therefore, the modeling of the plantar surface is carried out in good manner.

## Acknowledgments

The author would like to thank the financial support by CONACYT Mexico.

## References

- [1] **A. Descatoire, A. Thévenon, P. Moretto.** Baropodometric information return device for foot unloading. *Medical Engineering & Physics*, 2009, Vol. 31, 607-613.
- [2] **A. Agić.** Foot structure descriptors, *Ergonomia IJE&HF*, 2006, Vol. 28, 237-248.
- [3] **A. S. Rodrigo, R. S. Goonetilleke, C. P. Witana.** Model based foot shape classification using 2D foot outlines. *Computer-Aided Design*, 2012, Vol. 44, 48-55.
- [4] **A. Krishnamurthy, R. Khardekar, S. McMains.** Optimized GPU evaluation of arbitrary degree NURBS curves and surfaces. *Computer-Aided Design*, 2009, Vol. 41, 971-980.
- [5] **C. G. Provatidis, S. K. Isidorou.** Solution of one-dimensional hyperbolic problems using cubic B-Splines collocation. *International Journal of Computer Science and Application*, 2012, Vol. 1, 12-18.
- [6] **C. M. Chuang, H. T. Yua.** A new approach to z-level contour machining of triangulated surface models using fillet end mills. *Computer-Aided Design*, 2005, Vol. 37, 1039-1051.
- [7] **D. Cotoros, M. Baritz, A. Stanciu.** Conceptual analysis of correspondence between plantar pressure and corrective Insoles. *World Academy of Science, Engineering and Technology*, 2011, Vol. 59, 155-158.
- [8] **G. V. Ravi Kumar, P. Srinivasan, K. G. Shastry, B. G. Prakash.** Geometry based triangulation of multiple trimmed NURBS surfaces. *Computer-Aided Design*, 2001, Vol. 33, 439-454.
- [9] **H. Frederick, G. J. Lieberman.** *Introduction to operations research.* McGraw-Hill, USA, 1982, p. 754-758.
- [10] **J. A. Muñoz Rodriguez, F. Chaves.** Foot sole scanning by laser metrology and computer algorithms. *Laser Scanner Technology*, Intech, 2012, p. 1-14.
- [11] **J. A. Muñoz-Rodriguez.** Computer vision of the foot sole based on laser metrology and algorithms of artificial intelligence. *Optical Engineering*, 2009, Vol. 48, 123604-1-123604-13.
- [12] **J. L. Vilaca, J. C. Fonseca, A. M. Pinho.** Calibration procedure for 3D measurement system using two cameras and a laser line. *Optics and laser Technology*, 2009, Vol. 41, 112-119.
- [13] **J. Wang, H. Zhang, G. Lu, Z. Liu.** Rapid parametric design methods for shoe-last customization. *International Journal of Advanced Manufacturing Technology*, 2011, Vol. 54, 173-186.
- [14] **L. J. Ferrer-Arnau, R. Reig-Bolaño, P. Marti-Puig, A. Manjabacas, V. Parisi-Baradad.** Efficient cubic spline interpolation implemented with FIR filters. *International Journal of Computer Information Systems and Industrial Management Application*, 2013, Vol. 5, 98-105.

- [15] **L. Piegl, W. Tiller.** *The NURBS Book*, Springer, 2<sup>nd</sup> Ed., U.S.A. 1997, p. 50-58.
- [16] **M. E. Mortenson.** *Geometric Modeling*, Willey, 2<sup>nd</sup> Ed.; U.S.A. 1997, p. 83-105.
- [17] **M. Ihnatouski, A. Sviridenok, V. Lashkovski and B. Krupicz.** Biomechanical analysis of anthropometric and functional zones on human plantar at walking. *Acta mechanica et automatica*, 2008, Vol. 2, 19-23.
- [18] **M. Mahmud, D. Joannic, M. Roy, A. Isheil, J. F. Fontaine.** 3D part inspection path planning of a laser scanner with control on the uncertainty. *Computer-Aided Design*, 2011, Vol. 43, 345–355.
- [19] **M. Mochimaru, M. Kouchi, M. Dohi.** Analysis of 3-D human foot forms using the Free Form Deformation method and its application in grading shoe lasts. *Ergonomics*, 2000, Vol. 43, 1301-1313.
- [20] **N. E. Leal, O. Ortega Lobo, J. W. Branch.** Improving NURBS Surface Sharp Feature Representation. *International Journal of Computational Intelligence Research*, 2007, Vol. 3, No. 2, 131-138.
- [21] **O. Czarny, G. Huysmans.** Bezier surfaces and finite elements for MHD simulations. *Journal of Computational Physics*, 2008, Vol. 227, p. 7423–7445.
- [22] **P. H. Winston.** *Artificial Intelligence*, Addison-Wesley, Third Ed, U.S.A., p. 527-540, 1992.
- [23] **R. C. Gonzalez, P. Wintz.** *Digital image processing*, Addison-Wesley, 2<sup>nd</sup> Ed; U.S.A., 1987, p. 672-675.
- [24] **S. H. Kim, K. H. Shin.** A method for deforming a surface model using guide surfaces in shoe design. *International Journal of Advanced Manufacturing Technology*, 2009, Vol. 42, 60-72.
- [25] **S. Yuwen, G. Dongming, J. Zhenyuan, L. Weijun,** B-spline surface reconstruction and direct slicing from point clouds. *International Journal of Advanced Manufacturing Technology*, 2006, Vol. 27, 918–924.
- [26] **T. W. Weerasinghe, R. S. Goonetilleke.** Getting to the bottom of footwear customization. *Journal of Systems Science and Systems Engineering*, 2011, Vol. 20, 310-322.
- [27] **W. Chung, S. H. Kim, K. H. Shin.** A Method for planar development of 3D surfaces in shoe pattern design. *Journal of Mechanical Science and Technology*, 2008, Vol. 22, 1510-1519.
- [28] **Y. Cong, W. Lee, M. Zhang.** Regional plantar foot pressure distributions on high-heeled shoes-shank curve effects. *Acta Mechanica Sinica*, 2011, Vol. 27, 1091–1097.
- [29] **Y. F. Li, S. Y. Chen.** Automatic recalibration of an active structured light vision system. *IEEE Transactions on Robotics and Automation*, 2003, Vol. 19, 259-268.

Received June 2013.