

An Efficient Approach for Automatic Cloth Panel Extraction from Pattern Images

KhaledF. Hussain
Faculty of Computers and
Information, Assiut University,
Assiut, Egypt

Samia A. Ali
Department of Electrical
Engineering,
Faculty of Engineering,
AssiutUniversity,
Assiut, Egypt

SaherM.Malek
Faculty of Engineering,
Assiut University,
Assiut, Egypt

ABSTRACT

Producing 3D cloths for different body sizes from 2D patterns of any size has recently gained attention in both academic and industry fields. Most of the classical patterns in old books and magazines are available only in hardcopy forms. Thus, there is a need to produce softcopies of those patterns in the old books and magazines. In this research work, efficient solutions for these two issues are provided. First, a specific pattern size given any size pattern is automatically generated. This greatly facilitates the generation process of different size patterns in the apparel industry. Second, the design of customized apparel products provided in hardcopy images is fully automated. This automation process can greatly improve the efficiency of cloth production in the apparel industry.

General Terms

Pattern recognition, Virtual reality

Keywords

Garment industry, 3D clothes, Cloth modeling

1. INTRODUCTION

The desire of getting softcopies of old patterns found in hardcopy forms in books and magazines has been considered a critical issue and need to be solved efficiently. Furthermore, the problem of obtaining 3D cloths for different body sizes from 2D patterns of a specific size requires fast and accurate solutions. Moreover, the pattern generation process is considered to be a bottleneck for the garment manufacturing, especially for producing different size garments for the same style patterns [1].

In the literature, three methods are available for modeling of virtual cloths: The first method [2] is using the traditional approach to produce real cloth through drawing the 2D panels. This approach requires a skillful pattern-maker for the 2D panel drawing. The second method [3] is producing the 2D sketch based on 3D features of the character model. This approach is efficient for simple clothes, but it is hard to use for complex ones. The third method [4] is designing clothes in 3D by modeling 2D mesh cloth around the character model. This approach requires an intensive labor to modify the size of the cloth model to fit different character models.

There are many challenges in the garment production industry that are related to the different features of customers in order to produce maximally conformable garments. Thus, it is very important for the clothdesigners and manufacturers to know information about human body outer shape description, sizing and measurements. Currently the most important challenge for the clothing industry is the improvement on the basic stage of the designing process [5]. Also, it is important to know the 3-

D tools that might be used in the garment design to improve the efficiency of pattern generation and producing a more attractive design presentation.

The classical patterns in old books and magazines are available only in hardcopy forms. Therefore, it is desirable in some instances to produce softcopies of those patterns in the old books and magazines. It is of some practical interest to produce 3D cloths for different body sizes from 2D patterns of any given size. In this work, efficient solutions to solve these two problems are introduced. More specifically, external boundary detection and dart detection techniques are used to automatically extract panels from pattern images. Furthermore, for a specific given panel, a cubic Bezier curve representing the panel is generated. Then, alinear conformal transformation and/or affine transformation are used to generate the panel with the desired size. Thus, the two proposed techniques can greatly improve the efficiency of cloth production in the apparel industry.

The rest of this paper is organized as follows. Section 2 summarizes the related work to the proposed research. Section 3 introduces the proposed reliable approach for automatic panel extraction. Resizing panel is presented in Section 4. Developed software and the results are reported in Section 5. Finally, conclusions of this research work are shown in Section 6.

2.RELATED WORK

There is a number of techniques that can be used for modeling of virtual cloths. One approach is based on 2D patterns [2]. It is mainly, using the traditional approach of making real cloth through drawing 2D panels, fitting them to a character model, creating seams for sewing the panels, and running a cloth simulation to assist the deformation of the 2D panels into the character model. The main disadvantage of this approach is that it needs a skilled pattern-maker for the drawing of the 2D panels. Another is a 2D sketch approach based on 3D features of the character model [3]. It consists of three main parts; construction of garment feature templates, 3D profile specification using 2D strokes, and construction of garment surfaces interpolating the specified 3D profiles. This 2D sketch-based approach is efficient for modeling simple clothes, but it is difficult to use in modeling complex clothes. A third approach is designing clothes in 3D by modeling 2D mesh cloth around the character model [4]. The output 3D garment models can be fit on different character models by deforming the garment mesh to match the different shapes of the character models [6]. Normally, the cloth model is tailored on a specific character model; therefore it is necessary to modify the size of the cloth model to fit different character models which is not a straight forward task and needs an

intensive labor. Modifying the size of the cloth is required in many applications such as online clothing shops.

The current proposed research work has two main contributions; one is the automatic extraction of cloth panels from pattern images, while the other is the generation of different cloth sizes for different character models from a single given size pattern. The proposed research work relied on three image processing techniques: snakes, curve representation, and transformations. Active contours, or snakes, are used widely in the literature to find object boundaries. Curve representations are used to draw panels which consist mainly of curves. Transformation techniques have become very popular and important in image processing. Transformation is used in deforming the panels.

Active contours, or snakes, are curves defined within an image domain that can move under the influence of internal forces coming from within the curve itself and external forces computed from the image data [7]. They are often used in computer vision and image processing techniques to detect and locate objects, and to describe their shapes. For example, a snake might be used to automatically find a manufactured part on an assembly line, the outline of an organ in a medical image, and the outline of a pattern in this work. There are two well-known types of snakes, the traditional snake and the gradient vector flow (GVF) snake. In this paper, the gradient vector flow (GVF) snake is used. The advantages of the GVF snake over the traditional one are its insensitivity to initialization and its ability to move into boundary concavities. Its initializations can be inside, outside, or across the object's boundary. The gradient vector flow (GVF) snake begins with the calculation of field of forces, called the GVF forces, over the image domain. The GVF forces are used to drive the snake, modeled as a physical object having a resistance to both stretching and bending, towards the boundaries of the object. The GVF external forces make the GVF snake inherently different from the previous snakes. The GVF forces are derived from a diffusion operation, thus they tend to extend very far away from the pattern.

Curves are normally used to draw panels. A curve can be as simple as a line or as complex as a B-Spline. In general, Curves can be classified as follows:

I- Analytical curves: This type of curves can be represented by a simple mathematical equation, such as, a circle or an ellipse. They cannot be used to represent curves of patterns because of their complex shapes.

II- Interpolated curves: They are drawn by interpolating the given data points and have fixed forms which are dictated by the given data points. These curves cannot be used to represent curves of the pattern because they are hard to edit, have limited flexibility, and needing a large number of data points.

III- Approximated curves: These curves provide the most flexibility in the drawing of complex shapes. There are many types of approximated curves such as Bezier and B-spline. These curves can be joined together to form a smooth curve, and local changes in the shape do not affect the entire shape of the curve [8].

Bezier curves are most often used in a number of applications due to their flexibility in forming curves. Bezier curves differ from other types of parametric curves by the type of basis polynomials used to form them [9]. Bezier curves are now widely used in many fields such as industrial and computer-aided design, vector-based drawing, font design and 3D modeling. The Bezier curve of the third order is the most commonly used and it is fully defined by four points: two endpoints (P_0, P_3) and two control points (P_1, P_2). The control points do not lie on the curve itself, but they define its shape.

There are many properties of Bezier Curves such as endpoint interpolation, convex hull property, and prescribed tangent lines at the endpoints.

Transformation is mainly a function that creates images which are congruent to the original image. There are many types of transformations in the literature such as linear conformal transform, box transformation, composite transformation, projective transformation, polynomial transformation, LWM transformation (local weighted mean), piecewise transformation, custom transformation, and affine transformation [10]. The affine, linear conformal, projective, and polynomial transformations are global transformations. In these types of transformations, a single mathematical expression applies to an entire image. However, the piecewise, and LWM (local weighted mean) transformations are local transformations. In these transformations, different mathematical expressions apply to different regions within an image. In this research work, linear conformal transformation and affine transformation are used. The affine transformation is a mixture of scaling, rotation, shearing, and translation processes. Straight lines remain straight and parallel lines remain parallel to avoid distortion of the main features of the pattern. The affine transform requires at least three pairs of control points. On the other hand, linear conformal transform is a mixture of scaling, rotation, and translation processes. Thus linear conformal transform requires at least two pairs of control points.

Due to the importance of virtual clothing for the cloth design industry, computer generated movies and games, thus cloth simulation is needed. Cloth simulation is one of the main subjects in computer animation that recently gained attention in the literature [11-17]. Early cloth simulations focused on generating the geometrical features of clothes [18]. With the fast advance in technology of graphic cards and computers' processing power, an accurate generation of movement of clothes has been achieved, using particle systems [15, 16] or finite elements [19]. The modeling of virtual cloths requires special tailoring skills.

3. AUTOMATIC PANEL EXTRACTION

In this section, a novel reliable approach for automatic panel detection based on image processing techniques from pattern images is presented. The proposed approach consists of two main stages: automatic external boundary detection and automatic dart detection.

To detect external boundaries of panels from pattern images, the next steps are followed:

- 1- Given an input gray pattern image $I(x,y)$, the input image is converted to gray level, if the input image is a color image.
- 2- The grayscale image is converted into a binary image and the connected components (C_1, C_2, \dots, C_n) are computed for the binary image.
- 3- For each connected component, a panel or a non-panel (e.g., arrows and words) test is performed. Then the width, height, and area of the connected component are computed. If the width is less than a threshold value T_w , the height is less than a threshold value T_H , or the area is less than a threshold value T_A , the connected component is considered a non-panel.
- 4- The canny edge detection is applied to the image $I(x,y)$ followed by the boundary trace algorithm [20]; the result of the external boundary trace is sometimes incorrect, as shown in Figure 1(c). This problem is a common problem, especially for old pattern books. To overcome this problem, the Euler number is computed of the binary images. If the Euler

number is equal to zero, this means that applying the boundary trace algorithm produces an incorrect result. Thus, the snake technique is used in this case.

- 5- The snake based on Gradient Vector Flow (GVF) [7] is applied to extract the correct boundary of the panel. The snake is used to fit closed curves which is usually the case in cloth panels. The input to the snake is the boundary that is computed in the previous step. Then the snake is attracted towards the boundary of the panel by forces called gradient vector flow (GVF).

The gradient vector flow field is defined as $V(x, y) = [u(x, y), v(x, y)]$ which minimizes the energy ϵ . ϵ is defined as follows:

$$\epsilon = \iint \mu(u_x^2 + u_y^2 + v_x^2 + v_y^2) + |\nabla f|^2 |v - \nabla f|^2 dx dy, \quad (1)$$

where the coefficient μ specifies the relative importance of the first term versus the second term and $f(x, y) = |\nabla I(x, y)|^2$.

Equation (1) can be rewritten in the following general form:

$$\epsilon = \iint F(u, u_x, u_y, v, v_x, v_y, x, y) dx dy, \quad (2)$$

which has the following general solution that is represented by the two Euler equations:

$$\begin{aligned} \frac{\partial u}{\partial t} &= - \left[\frac{\partial F}{\partial u} - \frac{\partial}{\partial x} \left(\frac{\partial F}{\partial u_x} \right) - \frac{\partial}{\partial y} \left(\frac{\partial F}{\partial u_y} \right) \right] \\ \frac{\partial v}{\partial t} &= - \left[\frac{\partial F}{\partial v} - \frac{\partial}{\partial x} \left(\frac{\partial F}{\partial v_x} \right) - \frac{\partial}{\partial y} \left(\frac{\partial F}{\partial v_y} \right) \right]. \end{aligned} \quad (3)$$

The general solution in (3) can be applied to the specific problem in (1) as follows:

$$\begin{aligned} \mu \nabla^2 u - (u - f_x)(f_x^2 + f_y^2) &= 0 \\ \mu \nabla^2 v - (v - f_y)(f_x^2 + f_y^2) &= 0. \end{aligned} \quad (4)$$

The iterative solution to the equations in (4) is:

$$\begin{aligned} u_{i,j}^{n+1} &= (1 - b_{i,j} \Delta t) u_{i,j}^n \\ &\quad + r(u_{i+1,j}^n + u_{i,j+1}^n + u_{i-1,j}^n + u_{i,j-1}^n \\ &\quad - 4u_{i,j}^n) + c_{i,j}^1 \Delta t \end{aligned}$$

$$v_{i,j}^{n+1} = (1 - b_{i,j} \Delta t) v_{i,j}^n + r(v_{i+1,j}^n + v_{i,j+1}^n + v_{i-1,j}^n + v_{i,j-1}^n - 4v_{i,j}^n) + c_{i,j}^2 \Delta t \quad (5)$$

Where $b(x, y) = f_x(x, y)^2 + f_y(x, y)^2$, $c^1 = b(x, y) f_x(x, y)$, $c^2 = b(x, y) f_y(x, y)$, and $r = \frac{\mu \Delta t}{\Delta x \Delta y}$.

The boundary is defined as a parametric curve $c(s) = (x(s), y(s))$, $s \in [0, 1]$, where s is the arc length. The final position of the boundary minimizes the energy, E_b :

$$E_b = E_c + E_s + E_{ext} \quad (6)$$

where $E_c = \frac{1}{2} \int_0^1 \alpha |c'(s)|^2 ds$, $E_s = \frac{1}{2} \int_0^1 \beta |c''(s)|^2 ds$, and

$$E_{ext} = \int_0^1 E_{img}(c(s)) ds. \text{ Thus, } E_b = \int_0^1 \frac{1}{2} [\alpha |c'(s)|^2 + \beta |c''(s)|^2] + E_{img}(c(s)) ds \quad (7)$$

The energy E_b is defined as the sum of three energy terms: (1) E_c is the continuity term which is responsible for the shrinking of the contour. (2) E_s is the smoothness term which is used to smooth the curve. (3) E_{ext} is an edge attraction term to attract the contour towards the boundary of the panel.

The solution of equation (7) that minimizes the snake's energy E_b must satisfy the Euler equation

$$\alpha c''(s) - \beta c''''(s) - \nabla E_{img} = 0 \quad (8)$$

Where $\nabla E_{img} = -v$, is calculated in equation (5). To solve equation (8), the contour $c(s)$ is consider as a function of time, i.e., $c(s, t)$

$$\frac{\partial c(s, t)}{\partial t} = \alpha c''(s, t) - \beta c''''(s, t) + v. \quad (9)$$

Equation (9) can be solved iteratively until $\frac{\partial c(s, t)}{\partial t} = 0$. The result is the boundary points $z = \{z_1, z_2, \dots, z_n\}$. Figure 1(d) shows the boundary after applying the snake technique.

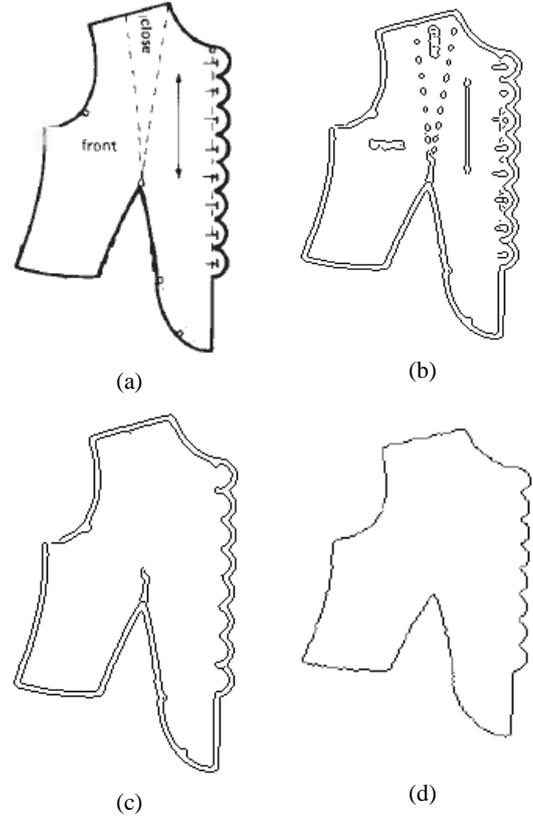


Fig. 1: (a) The original scanned pattern. (b) The image after applying the canny edge detection to the panel in image (a). (c) Incorrect boundary trace. (d) The boundary after applying the snake technique.

Applying the previous steps to a panel having darts will produce an incorrect boundary as shown in Figure 2. To remedy this problem, the darts are automatically detected and inserted into the detected boundary in the previous steps. More specifically, the exterior boundary B_E is traced for each panel, as well as the boundaries of the interior inside this panel B_I , in the binary image, as shown in Figure 3. For each interior boundary B_I , let p be the set of the points of the B_I . For each point $p_i \in p$, a test is performed to check whether it is a dart's corner or not as follows.

- 1- Let $j = T_1$
- 2- Calculate the vectors $v_1 = p_{i-j} - p_i$ and $v_2 = p_{i+j} - p_i$.
- 3- Normalize the two vectors v_1 and v_2 .
- 4- Calculate the angle between the two vectors $\varphi = \cos^{-1}(v_1 \cdot v_2)$.
- 5- If $\varphi \leq T_{\varphi_1}$ or $\varphi \geq T_{\varphi_2}$ exit the loop because the point p_i is not a dart's corner.
- 6- Increment j by 2
- 7- If $j = T_2$ the point p_i is consider as a dart's corner; so add the point p_i to the set S and exit the loop.
- 8- Loop through steps 2-7.

For each dart's corner p_k in the set S , the two intersection points (IP_k^1, IP_k^2) between the boundary of the corresponding dart and the exterior boundary B_E are calculated as follows.

- I. Calculate the first intersection point IP_k^1 by performing the following steps:

- 1- Let $j = T_3$
- 2- $d = \min_{v \in B_E} \left(\sqrt{((p_{k-j}^x - h^x)^2 + (p_{k-j}^y - h^y)^2)} \right)$

- 3- If $d \leq T_4$ $IP_k^1 = \arg \min_{h \in B_E} \left(\sqrt{((p_{k-j}^x - h^x)^2 + (p_{k-j}^y - h^y)^2)} \right)$, set $j_k^1 = j$, and exit the loop.
 - 4- Increment j
 - 5- If $j = T_4$ remove the dart's corner p_k from the set S and exit the loop.
 - 6- Loop through steps 2-5.
- II- Calculate the second intersection point IP_k^2
- 1- Let $j = T_3$
 - 2- $d = \min_{v \in B_E} \left(\sqrt{((p_{k+j}^x - h^x)^2 + (p_{k+j}^y - h^y)^2)} \right)$
 - 3- If $d \leq T_4$, $IP_k^2 = \arg \min_{h \in B_E} \left(\sqrt{((p_{k+j}^x - h^x)^2 + (p_{k+j}^y - h^y)^2)} \right)$, set $j_k^2 = j$, and exit the loop.
 - 4- Increment j
 - 5- If $j = T_4$ remove the dart's corner p_k from the set S and exit the loop.
 - 6- Loop through steps 2-5.

III- Replace the (IP_k^1, IP_k^2) segment of the exterior boundary B_E with the (j_k^1, j_k^2) segment of the dart k .

The cubic Bezier curve fitting is applied to the boundary B_E , for several reasons: (1) smoothing the data and improving the appearance of the boundary, (2) the problem of changing the size of the panel becomes a matter of changing the position of the control points and a panel can be redrawn again without losing a single detail; (3) the control points allow the user to easily change the shape of the panel;

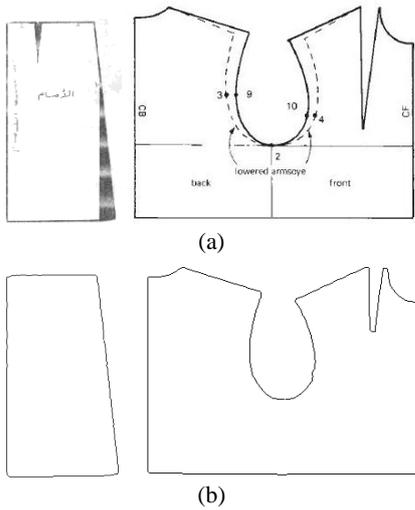


Fig. 2: (a) The original scanned patterns. (b) Incorrect boundaries.

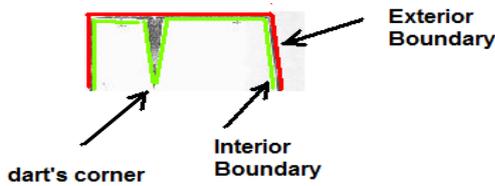


Fig. 3: An example illustrating the exterior boundary, interior boundary, and dart's corner

Cubic Bezier curves are defined using four control points (P_0, P_1, P_2, P_3) , known as knots. P_0 and P_3 are the end

points of the curve, while P_1 and P_2 define the slope at the end points. The Bezier curve is given by:

$$B(t_i) = \sum_{k=0}^3 B_k(t_i) P_k, 0 \leq t_i \leq 1, \quad (10)$$

Where $B_k(t_i) = \binom{3}{k} t_i^k (1 - t_i)^{3-k}$

The two points P_0 and P_3 are known and are equal to the end points of the curve. While the two points P_1 and P_2 are unknown. They are computed based on the least square method by minimizing the squared distance between the boundary z and the Bezier curve as follows:

$$S = \sum_{i=1}^n (z_i - B(t_i))^2, \quad (11)$$

Where $z_i \in z$ is a boundary point and n is the number of boundary points.

Substitute equation (10) into equation (11) yields:

$$S = \sum_{i=1}^n (z_i - B_0(t_i)P_0 - B_1(t_i)P_1 - B_2(t_i)P_2 - B_3(t_i)P_3)^2 \quad (12)$$

P_1 and P_2 are calculated by solving the following two equations:

$$\frac{\partial S}{\partial P_1} = 0, \quad \frac{\partial S}{\partial P_2} = 0 \quad (13)$$

The two equations in (13) can be written as follows:

$$a_1 P_1 + a_2 P_2 = b_1, \quad a_2 P_1 + a_3 P_2 = b_2 \quad (14)$$

Where

$$a_1 = 2 \sum_{i=1}^n (B_1(t_i))^2, \quad a_2 = 2 \sum_{i=1}^n B_1(t_i)B_2(t_i), \quad a_3 = 2 \sum_{i=1}^n (B_2(t_i))^2$$

$$b_1 = \sum_{i=1}^n 2B_1(t_i)(z_i - B_0(t_i)P_0 - B_3(t_i)P_3), \quad b_2 = \sum_{i=1}^n 2B_2(t_i)(z_i - B_0(t_i)P_0 - B_3(t_i)P_3)$$

The solution of the two equations in (14) is:

$$P_1 = \frac{a_3 b_1 - a_2 b_2}{a_1 a_3 - a_2^2}, \quad P_2 = \frac{a_2 b_1 - a_1 b_2}{a_2^2 - a_1 a_3} \quad (15)$$

The cubic Bezier curve fitting is applied to the boundary z as follows:

- 1- Let Err_{max} be the maximum allowed fitting error.
- 2- Let $y = \{z_1, z_n\}$ be the set of breakpoints, where z_1 is the first point of z and z_n is the last point of z .
- 3- Set cubic Bezier knots $k = (P_0^0, P_1^0, P_2^0, P_3^0)$, where $P_0^0 = z_1, P_3^0 = z_n$, and P_1^0 and P_2^0 are computed using equation (15).
- 4- Use cubic Bezier interpolation to generate n points (b_1, b_2, \dots, b_n) at $t = (\frac{0}{n}, \frac{1}{n}, \frac{2}{n}, \dots, \frac{n}{n})$ based on the knots k .
- 5- Calculate the fitting error Err_f as follows:

$$Err_f = \max_{i=1,2,\dots,n} |b_i - z_i|$$

6- If Err_f is greater than Err_{max} , insert a new breakpoint into the set $y = \{z_1, z_k, z_n\}$, where z_k is the boundary point that has the error Err_f .

7- Set cubic Bezier knots $k = (P_0^0, P_1^0, P_2^0, P_3^0, P_1^1, P_2^1, P_3^1)$, where $P_0^0 = z_1, P_3^0 = z_k, P_1^0 = z_k$, and $P_3^1 = z_n$, then calculate P_1^1, P_2^1, P_1^1 , and P_2^1 using equation (15).

8- Repeat the steps 4-7 until the Err_f is less than or equal to Err_{max} .

4. RESIZING OF PANELS

The problem of producing panels of different sizes from a given panel size is similar to image registration techniques. The proposed method gives the user two options: a linear conformal transformation and affine transformation. A linear conformal transformation is easy to use; it only requires two pair of points to be calculated. It allows rotation, translation, and scaling of both axes with the same amount, i.e., it will lock the aspect ratio. The main advantage of the linear conformal transformation is the ease of use. However, linear conformal transformation is not well applicable in the cases where a change in the aspect ratio is required. In such cases the affine transformation must be used. It allows for rotation,

translation, and scaling of both axes with different amount. The affine transformation needs three pairs of points to be calculated.

4.1 Linear Conformal Transformation

For each panel, the user creates two tapes (the tape is defined as a line segment with a start point sp and an end point ep): one tape $t_1 = (sp_1, ep_1)$ on the panel and the other tape $t_2 = (sp_2, ep_2)$ on the cross section image which is generated from a given 3D character model as shown in Figure 3.

The Linear conformal transformation is defined as follows:

$$H_s x = \hat{x},$$

$$\begin{bmatrix} h_1 & h_2 & t_x \\ -h_2 & h_1 & t_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ 1 \end{bmatrix} = \begin{bmatrix} \hat{x}_1 \\ \hat{x}_2 \\ 1 \end{bmatrix},$$

where H_s is the 3x3 coordinate transformation matrix, x denotes the panel homogeneous coordinates and \hat{x} is a new position in the cross section image. The Linear conformal transformation requires four parameters (h_1, h_2, t_x, t_y). The four parameters can be calculated by solving the following four equations:

$$\begin{aligned} sp_{1x}h_1 + sp_{1y}h_2 + t_x &= sp_{2x} \\ ep_{1x}h_1 + ep_{1y}h_2 + t_x &= ep_{2x} \\ -sp_{1x}h_2 + sp_{1y}h_1 + t_y &= sp_{2y} \\ -ep_{1x}h_2 + ep_{1y}h_1 + t_y &= ep_{2y} \end{aligned}$$

Where $sp_1 = (sp_{1x}, sp_{1y})$, $sp_2 = (sp_{2x}, sp_{2y})$, $ep_1 = (ep_{1x}, ep_{1y})$, and $ep_2 = (ep_{2x}, ep_{2y})$

The solution of the previous equations is:

$$\begin{bmatrix} h_1 \\ h_2 \\ t_x \\ t_y \end{bmatrix} = \begin{bmatrix} sp_{1x} & sp_{1y} & 1 & 0 \\ ep_{1x} & ep_{1y} & 1 & 0 \\ sp_{1y} & -sp_{1x} & 0 & 1 \\ ep_{1y} & -ep_{1x} & 0 & 1 \end{bmatrix}^{-1} \begin{bmatrix} sp_{2x} \\ ep_{2x} \\ sp_{2y} \\ ep_{2y} \end{bmatrix}$$

4.2 Affine Transformation

For each panel, the user creates three pairs of points: three points on the panel (pp_1, pp_2, pp_3) and the three corresponding points (cp_1, cp_2, cp_3) on the cross section image.

The affine transformation is defined as follows:

$H_a x = \hat{x}$,

$$\begin{bmatrix} a_1 & a_2 & t_x \\ a_3 & a_4 & t_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ 1 \end{bmatrix} = \begin{bmatrix} \hat{x}_1 \\ \hat{x}_2 \\ 1 \end{bmatrix}$$

where H_a is the 3x3 coordinate transformation matrix. The affine transformation requires six parameters ($a_1, a_2, a_3, a_4, t_x, t_y$). The six parameters can be calculated by solving the following six equations:

$$\begin{aligned} pp_{1x}a_1 + pp_{1y}a_2 + t_x &= cp_{1x} \\ pp_{2x}a_1 + pp_{2y}a_2 + t_x &= cp_{2x} \\ pp_{3x}a_1 + pp_{3y}a_2 + t_x &= cp_{3x} \\ pp_{1x}a_3 + pp_{1y}a_4 + t_y &= cp_{1y} \\ pp_{2x}a_3 + pp_{2y}a_4 + t_y &= cp_{2y} \\ pp_{3x}a_3 + pp_{3y}a_4 + t_y &= cp_{3y} \end{aligned}$$

where $pp_1 = (pp_{1x}, pp_{1y})$, $pp_2 = (pp_{2x}, pp_{2y})$, $pp_3 = (pp_{3x}, pp_{3y})$, $cp_1 = (cp_{1x}, cp_{1y})$, $cp_2 = (cp_{2x}, cp_{2y})$, and $cp_3 = (cp_{3x}, cp_{3y})$

The solution of the previous equations is:

$$\begin{bmatrix} a_1 \\ a_2 \\ t_x \end{bmatrix} = \begin{bmatrix} pp_{1x} & pp_{1y} & 1 \\ pp_{2x} & pp_{2y} & 1 \\ pp_{3x} & pp_{3y} & 1 \end{bmatrix}^{-1} \begin{bmatrix} cp_{1x} \\ cp_{2x} \\ cp_{3x} \end{bmatrix}$$

$$\begin{bmatrix} a_3 \\ a_4 \\ t_y \end{bmatrix} = \begin{bmatrix} pp_{1x} & pp_{1y} & 1 \\ pp_{2x} & pp_{2y} & 1 \\ pp_{3x} & pp_{3y} & 1 \end{bmatrix}^{-1} \begin{bmatrix} cp_{1y} \\ cp_{2y} \\ cp_{3y} \end{bmatrix}$$

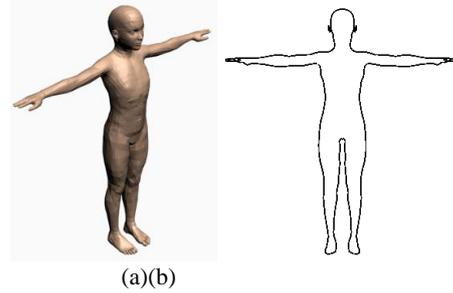


Fig.4: (a) 3D model of a human. (b) Cross-section of the 3D model in (a).

5. DEVELOPED SOFTWARE AND RESULTS

To test the effectiveness of the proposed approach in this research work, it was necessary to design a user friendly program to facilitate obtaining the results and visually inspecting them. The developed software package has been written in Matlab [20] and Maxscript [4]. The developed software takes as an input the 3D character models. Then it extracts the cross-section and saves it in a file, see Figure 4.

Once the developed software is started, it reads two files: the cross-section of the 3D character model and the intensity image of the cloth pattern. The main window of the developed software is shown in Figure 5. The developed algorithm described in Section 2 will automatically extract the panel. The developed software gives the user two options: "Create Tape for Registration" and "Create Points for Registration". The Create Tape for Registration option applies a linear conformal transformation which is easy to use since it only requires two tapes to be calculated per panel. One tape is drawn on the intensity image of the cloth pattern and the other tape is drawn on the cross-section of the 3D character model. The developed software allows the user to interactively modify the position of the tape. The main drawback of the linear conformal transformation is that it does not work in the cases where a change in the aspect ratio is required. In these cases the "Create Points for Registration" option should be used. It applies the affine transformation which requires three pairs of points to be calculated. The three points are placed on the intensity image of the cloth pattern and the other three points are placed on the cross-section of the 3D character model. Next, the user should press the Registration button. The developed program calculates the transformation parameters and transforms the panel from the coordinate of the cloth pattern's image to the coordinate of the 3D character model's cross-section. After the user has done fitting all the panels; the user creates seams for sewing the panels together. Then the user runs the cloth simulation to assist the deformation of the panels into the character model.

The developed approach of this research work is tested with several cloth pattern images and different character sizes. The test is done to demonstrate the accuracy achieved using the developed approach in extracting boundaries of the panel from cloth pattern images, changing their sizes to fit different character models, and sewing the panels together. Figures

6(a), 7(a), 8(a), 9(a), 10(a), and 11(a) show scanned cloth pattern images. Figures 6(b), 7(b), 8(b), 9(b), 10(b), and 11(b) show the boundary of the panels extracted using the proposed approach.

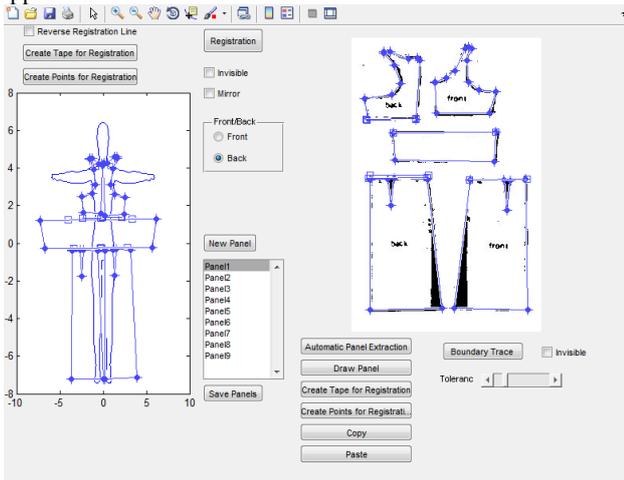


Fig. 5: Main window of the developed software

The proposed approach produces incorrect boundary of the panels extracted with some patterns as shown in Figure 12 and Figure 13. Figures 12(a) and 13(a) show a scanned cloth pattern image. Figures 12(b) and 13(b) show the incorrect boundary of the panels extracted using the proposed approach. The problem with the two patterns is that there are words or drawing inside the panel that touches or very close to the boundary of the panel.

Figure 14 and Figure 15 show the fitting of the cloth pattern in Figures 8 and 9 respectively to different human models using the proposed panel resizing approach to fit all the panels. Then, the panels are sewn together to construct the 3D garments.

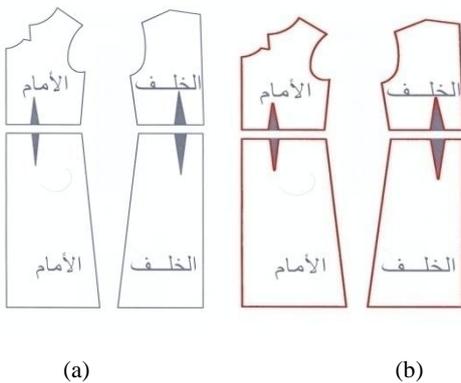


Fig. 6: (a) The original scanned patterns. (b) The boundary of the panels extracted using the proposed approach

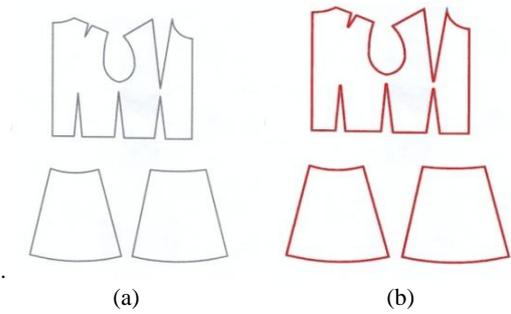


Fig. 7: (a) The original scanned patterns. (b) The boundary of the panels extracted using the proposed approach.

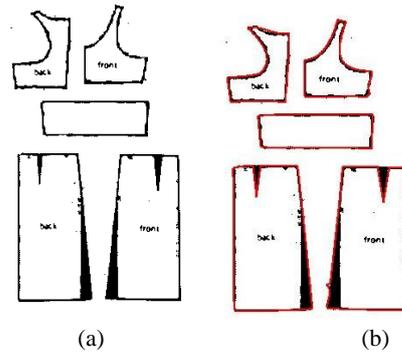


Fig. 8: (a) The original scanned patterns. (b) The boundary of the panels extracted using the proposed approach.

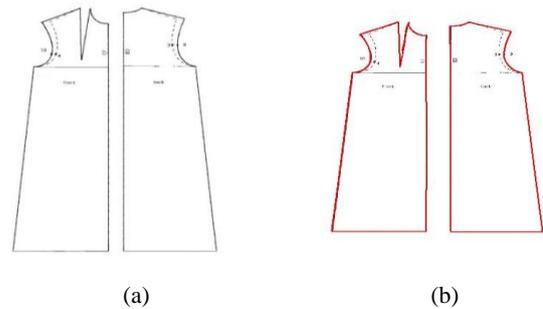


Fig. 9: (a) the original scanned patterns. (b) The boundary of the panels extracted using the proposed approach.

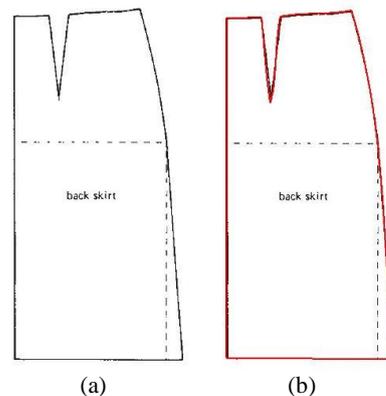


Fig. 10: (a) the original scanned patterns. (b) The boundary of the panels extracted using the proposed approach

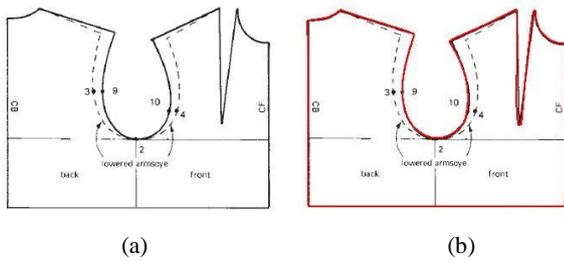


Fig. 11: (a) The original scanned patterns. (b) The boundary of the panels extracted using the proposed approach.

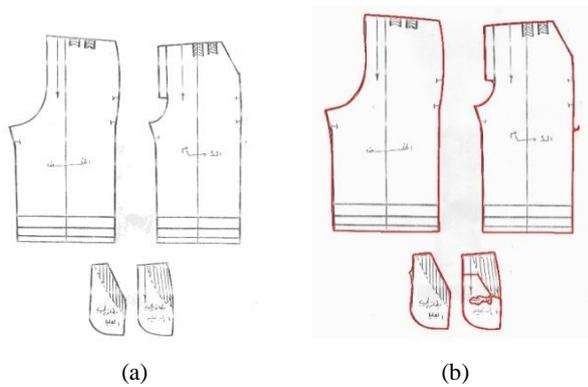


Fig.12: (a) The original scanned patterns. (b) Incorrect boundary of the panels extracted using the proposed approach.

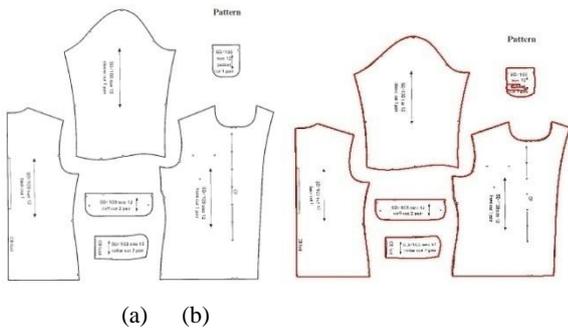


Fig.13: (a) The original scanned patterns. (b) Incorrect boundary of the panels extracted using the proposed approach.

6. CONCLUSIONS

Classical patterns in old books and magazines are available only in hardcopy forms; to reproduce such patterns it is more convenient to obtain softcopies of those patterns. In this paper, a novel approach is introduced for fully automated extraction of panels from cloth pattern images and producing different size patterns from a given pattern size. A software package has been developed, and tested it on many cloth pattern images and different 3D models. The proposed approach accurately extracts the panels from cloth pattern images and perfectly fits them to different human models. Thus, the proposed approach is consequently very helpful for computer generated movies, computer games, and cloth design industry.

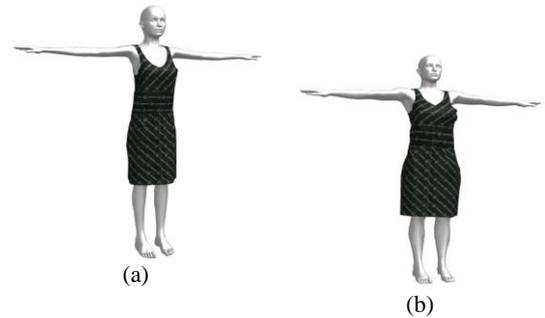


Fig. 14: The fitting of the cloth pattern in Figure8 to different human models.

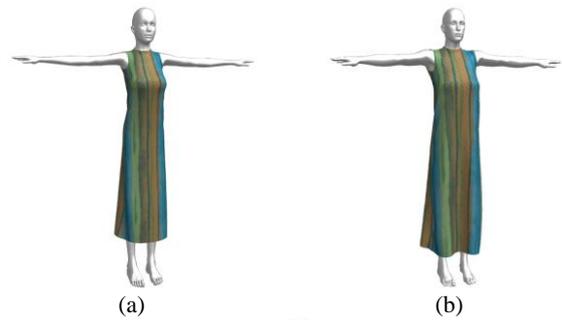


Fig. 15: The fitting of the cloth pattern in Figure9 to different human models.

7. REFERENCES

- [1] Charlie C. L. Wang.2004. CAD Tools in Fashion/Garment Design. Computer-Aided Design and Applications, Vol. 1, No.1, pp. 53-62.
- [2] J. Vince and R. Earnshaw, Protopsaltou, D., et al. 2002. A Body and Garment Creation Method for an Internet-Based Virtual Fitting Room. Advances in Modeling, Animation and Rendering, Springer-Verlag, 105-122.
- [3] Wang, C., Wang, Y., and Yuen, M.2003. Feature based 3D garment design through 2D sketches.Computer-AidedDesign, Volume 35, Issue 7, pp. 659–672

- [4] Autodesk 3ds Max. <http://usa.autodesk.com/3ds-max/>. Last accessed on July 2014.
- [5] Agnieszka Cichocka, Pascal Bruniaux. 2009. Comparison of Traditional 2D and Virtual Patterns Design in 3D. *Journal of Advanced Computational Intelligence and Intelligent Informatics*, Vol.13, No.5 pp. 542-549.
- [6] Li, J., Ye, J., Wang, Y., Bai, L. 2010. Fitting 3d garment models onto individual human models. *Computers & Graphics*, Vol. 34, No.6, pp. 742–755.
- [7] Xu, C., and Prince, J. L. 1998. Snakes, shapes, and gradient vector flow. *IEEE Transactions on Image Processing*, VOL. 7, NO. 3, pp. 359 – 36.
- [8] Saxena, Anupam, Sahay, Birendra. 2005. *Computer Aided Engineering Design*. Springer.
- [9] O. Ismail. 2010. Degree elevation of interval Bezier curves using Legendre-Bernstein basis transformations. *International Journal of Video & Image Processing and Network Security (IJVIPNS)*, Vol. 10, No.6.
- [10] Gonzalez Woods and Eddins. 2009. *Digital Image Processing Using Matlab*. Gatesmark Publishing; 2nd edition.
- [11] Jonathan M. Kaldor, Doug L. Jame. 2010. Efficient Yarn-based Cloth with Adaptive Contact Linearization. *ACM Trans. Graph*, Vol. 29, No. 4, Article No. 105.
- [12] Baraff, D., and Witkin, A. 1998. Large steps in cloth Simulation. *Proceedings of the SIGGRAPH '98*. pp.23-34, Orlando, FL, USA.
- [13] Kwang-Jin, Hyeong-Seok. 2002. Stable but responsive cloth. *SIGGRAPH'02*, pp. 604-611, San Antonio, TX, USA.
- [14] Carignan M, Yang Y. 1992. Dressing animated synthetic actors with complex deformable clothes. In *Proc. of SIGGRAPH'92*. pp. 99–104, Chicago, Illinois, USA.
- [15] Breen, D., House, D., and Wozny, M. 1994. Predicting the drape of woven cloth using interacting particles. In *Proc. of SIGGRAPH'94*, pp 365–372. Orlando, FL, USA.
- [16] Eberhardt B, Weber A. 1996. A fast, flexible, particle-system model for cloth draping. *Computer Graphics and Applications*, IEEE, Vol. 16, No. 5, pp. 52 - 59.
- [17] Pascal Vollino, Martin Courchesne. 1995. Versatile and efficient techniques for simulating cloth and other deformable objects. In *Proc. of SIGGRAPH'95*, Los Angeles, USA, pp.37–144.
- [18] Jerry Wef. 1986. The Synthesis of Cloth Objects. In *Proc. of SIGGRAPH'86*, Dallas, USA, Vol. 20, No.4.
- [19] Eischen, J.W. 1996. Finite-Element Modeling and Control of Flexible Fabric Parts. *Computer Graphics and Applications*, IEEE, Vol. 16, No. 5, pp. 71 - 80.
- [20] MATLAB. <http://www.mathworks.com>. Last accessed on July 2014.