# Automatic Cloth Panels Extraction and Resizing

Khaled F. Hussain
Faculty of Computers and Information, Assiut University, Assiut, Egypt

Samia A. Ali
Department of Electrical Engineering, Computer Section, Faculty of Engineering, Assiut University, Assiut, Egypt

## ABSTRACT

It is important to produce 3D cloths for different body sizes from 2D patterns of any size. Moreover, classical patterns in old books and magazines are available only in hardcopy forms. Thus it is imperative to produce softcopies of those patterns in old books and magazines. Solutions for these two issues are provided in this research work. First automating the design of customized apparel products from pattern images is provided which can greatly improves the efficiency of cloth production in the apparel industry. Second generating different sizes patterns from a given pattern size. This greatly facilitate the generation of pattern design in the apparel industry.

## General Terms

Pattern recognition, Virtual reality

## Keywords

Garment industry, 3D clothes, Cloth modeling

## 1. INTRODUCTION AND BACKGROUND

Producing convincing virtual cloth is important for enhancing the realism of computer generated movies and computer games. It is important as well for the cloth design industry. Existing cloth simulations can generate virtual garments with very realistic appearance. Cloth simulation is one of the main subjects in computer animation and a great deal of research is available in the literature [1-7]. Early cloth simulations focused on generating the geometrical features of clothes [8].With the great leap forward in technology and enhancements in graphic cards, coupled with decreasing the cost and increasing computers' processing power, an accurate generation of mechanical movement of clothes has been achieved, using particle systems [5, 6] or finite elements [9].

However, the modeling of virtual cloths is labor intensive and requires tailoring skills. Three methods are available and can be used for modeling of virtual cloths: The first method is based on 2D patterns [10]. It is 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 drawing of the 2D panels. One contribution of our research work is the automatic extraction of cloth panels from pattern images. The second method is a 2D sketches approach based on 3D feature of the character model [11]. 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. Sketch-based approach is efficient in modeling simple clothes but it is hard to use in modeling of complex clothes. The third method is designing clothes in 3D by modeling 2D mesh cloth around the character model [12]. The output 3D garment models can be fit on different character models by deforming the garment mesh to match the shapes of the character models [13].

The cloth model is tailored on a specific character model. However, it is not a straight forward task to modify the size of the cloth model to fit different character models. Thus it is also labor intensive and requires tailoring skills to reuse cloth models. Modifying the size of cloth is necessary in many applications such as online clothing shops. However, web shops that only provide simple apparel viewing websites do not provide consumers enough information to evaluate many cloth properties such as fitting. This leads to a high product return rate [14]. The other contribution of this research work is the generation of different cloth sizes for different character models from a given size pattern.

## 2. 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.

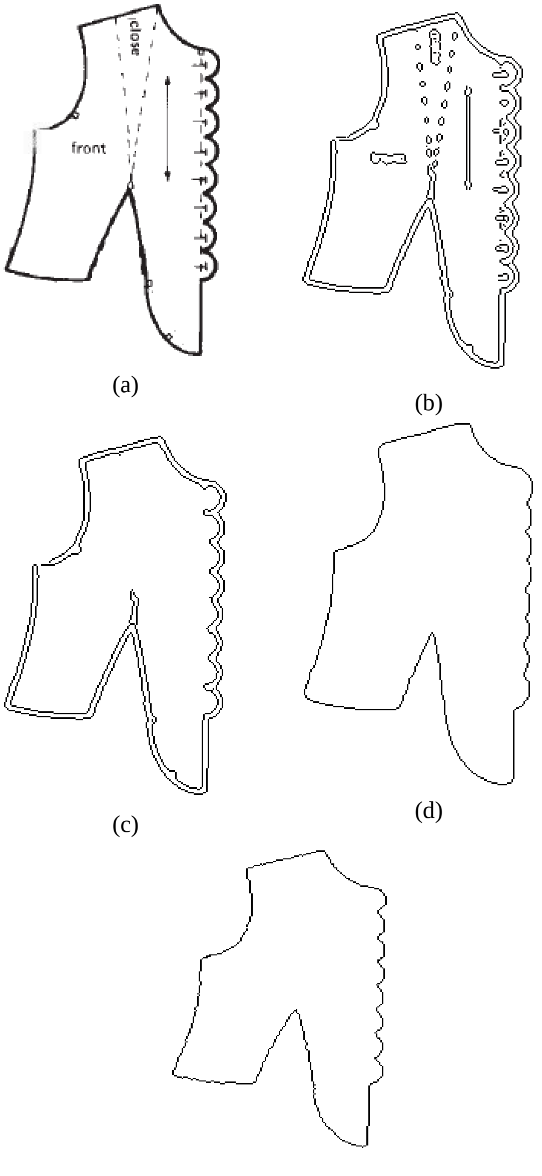
To detect external boundaries of panels from pattern images, the next steps are followed:

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 canny edge detection is applied to the image $I(x,y)$ followed by the application of the boundary trace algorithm [15], the result of the external boundary trace is incorrect as shown in Figure 1(c). This problem is a command problem especially for old pattern books. To overcome this problem, $I_{gg}(x,y)$ is computed as follows:

$$I_{gg}(x,y) = G_\sigma(x,y) * |\nabla I(x,y)| \qquad (1)$$

where $\nabla$ is the gradient operator and $G_\sigma(x,y)$ is the Gaussian lowpass filter with standard deviation σ. The value of σ should be large enough to cause the boundaries to become blurry. Such large $\sigma$'s is necessary in order to cover fuzzy parts due to the scanning process or old pattern papers. $I_c(x,y)$ is computed by applying the canny edge detector to $I_{gg}(x,y)$.



(a)

(b)

(c)

(d)

(e)

**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 output of applying the boundary trace algorithm to $I_c(x,y)$. (e) The boundary after applying the snake technique.**

3- The user clicks in any position $P_u$ on or near the boundary of the panel that he/she wants to extract. Then the nearest edge point $P_e$ in $I_c(x,y)$ which has the shortest Euclidean distance to $\underline{P_u}$ is found. The boundary trace algorithm [9] is then applied with inputs: $I_c(x,y)$ and $P_e$, as shown in Figure 1(d). The output boundary fits the panel roughly but it is not acceptable.

4- To make the boundary fit accurately the boundary of the panel, we apply the snake based on Gradient Vector Flow (GVF) [16]. The snake is used to fit closed curves which is usually the case in the 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 $\mathbf{v}(x,y) = [\,u(x,y), v(x,y)]$ which minimizes the energy ε. ε is defined as follows:

$$\varepsilon = \iint \mu\left(u_x^2+u_y^2+v_x^2+v_y^2\right) + |\nabla f|^2|\mathbf{v} - \nabla f|^2 dxdy, \qquad (2)$$

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

Equation (2) can be rewritten in the following general form:

$$\varepsilon = \iint F(u, u_x, u_y, v, v_x, v_y, x, y)dxdy, \qquad (3)$$

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

$$\frac{\partial u}{\partial t} = -\left[\frac{\partial F}{\partial u} - \frac{\partial}{\partial x}\left(\frac{\partial F}{\partial u_x}\right) - \frac{\partial}{\partial y}\left(\frac{\partial F}{\partial u_y}\right)\right]$$
$$\frac{\partial v}{\partial t} = -\left[\frac{\partial F}{\partial v} - \frac{\partial}{\partial x}\left(\frac{\partial F}{\partial v_x}\right) - \frac{\partial}{\partial y}\left(\frac{\partial F}{\partial v_y}\right)\right]. \qquad (4)$$

The general solution in (4) can be applied to our specific problem in (2) as follows:

$$\mu\nabla^2 u - (u - f_x)\left(f_x^2 + f_y^2\right) = 0$$

$$\mu\nabla^2 v - \left(v - f_y\right)\left(f_x^2 + f_y^2\right) = 0. \qquad (5)$$

The iterative solution to the equations in (5) is:

$$u_{i,j}^{n+1} = \left(1 - b_{i,j}\Delta t\right)u_{i,j}^n + r\left(u_{i+1,j}^n + u_{i,j+1}^n + u_{i-1,j}^n + u_{i,j-1}^n - 4u_{i,j}^n\right) + c_{i,j}^1\Delta t$$
$$v_{i,j}^{n+1} = \left(1 - b_{i,j}\Delta t\right)v_{i,j}^n + r\left(v_{i+1,j}^n + v_{i,j+1}^n + v_{i-1,j}^n + v_{i,j-1}^n - 4v_{i,j}^n\right) + c_{i,j}^2\Delta t, \qquad (6)$$

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 $\mathbf{c}(s) = \left(x(s), y(s)\right), s \in [0,1]$ , where $s$ is the arc length. The final position of the boundary minimizes the energy, $E_b$:

$$E_b = E_c + E_s + E_{ext},$$

where $E_c = \frac{1}{2}\int_0^1 \alpha\left|\mathbf{c}'(s)\right|^2 ds$ , $E_s = \frac{1}{2}\int_0^1 \beta\left|\mathbf{c}''(s)\right|^2 ds$, and $E_{ext} = \int_0^1 E_{img}\left(\mathbf{c}(s)\right)ds$. Thus,

$$E_b = \int_0^1 \frac{1}{2}\left[\alpha\left|\mathbf{c}'(s)\right|^2 + \beta\left|\mathbf{c}''(s)\right|^2\right] + E_{img}\left(\mathbf{c}(s)\right)ds \tag{7}$$

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 Eular equation

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

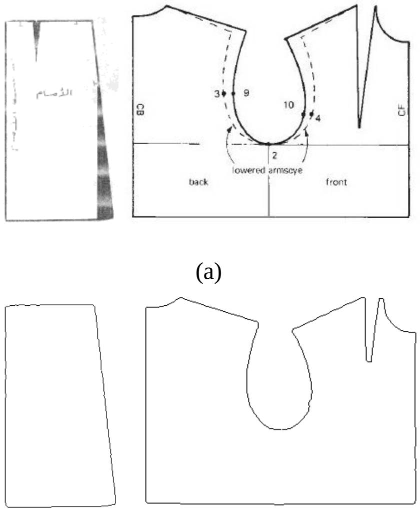
Where $\nabla E_{img} = -\mathbf{v}$ **,** is calculated in equation (6). To solve equation (8), we consider the contour **c**(*s*) to be a function of time, i.e., **c**(*s*,*t*)

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

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

Applying the previous steps to a panel contains 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 as follows:

1. Apply canny edge detector to the pattern image *I*(*x*, *y*).
2. Apply the Hough transform to the edge points inside the boundary **z**. Let $x\,cos(\theta_1) + y\,sin(\theta_1) = \rho_1$ , $x\,cos(\theta_2) + y\,sin(\theta_2) = \rho_2$ ,… $x\,cos(\theta_n) + y\,sin(\theta_n) = \rho_n$ be the output lines of the Hough transform.
3. Let ***p*** be the set of the intersection points $(x_{ij}, y_{ij})$ of each pair of lines $i,j \in \{1,2,\dots n\}, i \neq j$ and $\left|\theta_i - \theta_j\right| < T_1$, where $x_{ij} = \frac{\rho_i sin(\theta_j) - \rho_j sin(\theta_i)}{cos(\theta_i)sin(\theta_j) - cos(\theta_j)sin(\theta_i)}$, $y_{ij} = \frac{\rho_i cos(\theta_j) - \rho_j cos(\theta_i)}{sin(\theta_i)cos(\theta_j) - sin(\theta_j)cos(\theta_i)}$, and $T_1$ is a threshold.
4. Let ***h*** be the set of corner points found by applying the Harris corner detector [17] to the region in the pattern image *I*(*x*, *y*) that lies inside the boundary **z**.
5. Calculate the set of intersection points that have corner points near them as follows:
   $\mathbf{o} = \left\{p | p \in \mathbf{p} \text{ and } d < T_2 \text{where } d = \min_{\forall h \in \mathbf{h}}\left(\sqrt{\left((p_\mathbf{x} - h_x)^2 + \left(p_\mathbf{y} - h_y\right)^2\right)}\right)\right\}$ **,** where $T_2$ is a threshold.
6. Calculate the set of corner points that correspond to the set of intersection points as follows:
   $\mathbf{q} = \left\{q_p, \forall p \in \mathbf{p}\right\}$, where
   $q_p = arg_{h\in\mathbf{h}} \min \left(\sqrt{\left((p_\mathbf{x} - h_x)^2 + \left(p_\mathbf{y} - h_y\right)^2\right)}\right)$
7. For each point $q_p$ in **q** do the followings:
   i. Let $p = (x_{ij}, y_{ij})$ be the corresponding intersection point.
   ii. Let $l_i$ and $l_j$ be the two lines that intersect at intersection point $p$.
   iii. Calculate the intersection points ($b_i$ and $b_j$) between the two lines ($l_i$ and $l_j$) and the boundary **z,** respectively.
   iv. Divide the boundary **z** into two segments: $b_i b_j$ and $b_j b_i$, where a segment is defined as the set of all points between two consecutive points. Set **z** to be the longest segment.
   v. Add the points of the lines $b_i q_p$ and $q_p b_j$ to the segment **z.**



(a)

(b)

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

The cubic Bezier curve fitting is applied to the boundary **z**, for several reasons. They are (1) smoothing the data and improving the appearance of the boundary, (2) the problem of changing the size of the panel become 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;

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:

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

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

$P_0$ and $P_3$ are known and are equal to the end points of the curve. While $P_1$ 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:

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

where $z_i \in \mathbf{z}$ is a boundary point and *n* is the number of boundary points.

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

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

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

$$\frac{\partial S}{\partial P_1} = 0, \ \frac{\partial S}{\partial P_2} = 0 \tag{13}$$

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

$$a_1 P_1 + a_2 P_2 = b_1, a_2 P_1 + a_3 P_2 = b_2 \tag{14}$$

Where

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

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

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

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

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

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

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

f. If $Err_f$ is greater than $Err_{max}$ , insert a new breakpoint into the set **y**. Thus $\mathbf{y} = \{z_1, z_k, z_n\}$, where $z_k$ is the boundary point that has the error $Err_f$.
g. Set cubic Bezier knots $\mathbf{k} = (P_0^0,\ P_1^0, P_2^0, P_3^0, P_0^1, P_1^1, P_2^1, P_3^1)$, where $P_0^0 = z_1$, $P_3^0 = z_k$, $P_0^1 = z_k$, and $P_3^1 = z_n$, then calculate $P_1^0$,$P_2^0$, $P_1^1$, and $P_2^1$ using equation (15).
h. Repeat the steps d-g until the $Err_f$ is less than or equal to $Err_{max}$ .

# 3. RESIZING OF PANELS

The problem of producing panels of different sizes from a given panel size is similar to image registration techniques. The proposed software gives the user two options: nonreflective similarity transformation and affine transformation. The nonreflective similarity 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 nonreflective similarity transformation is the easy of usage. However, the nonreflective similarity transformation is not applicable for the cases that require a change in the aspect ratio. For 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 pair of points to be calculated.

## 3.1 Nonreflective Similarity 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 nonreflective similarity transformation is defined as follows:

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

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

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

$$sp_{1x} h_1 + sp_{1y} h_2 + t_x = sp_{2x}$$
$$ep_{1x} h_1 + ep_{1y} h_2 + t_x = ep_{2x}$$
$$-sp_{1x} h_2 + sp_{1y} h_1 + t_y = sp_{2y}$$

$$-ep_{1x}h_2 + ep_{1y}h_1 + t_y = ep_{2y}$$

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:

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

### 3.2 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:

$$H_a\mathbf{x} = \hat{\mathbf{x}},$$

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

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:

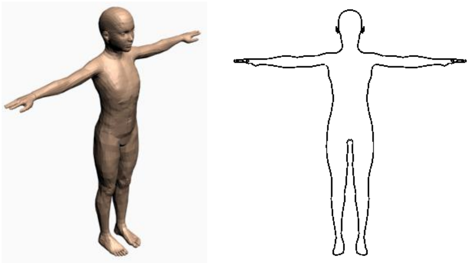
$$\begin{aligned} pp_{1x}a_1 + pp_{1y}a_2 + t_x &= cp_{1x} \\ pp_{2x}a_1 + pp_{2y}a_2 + t_x &= cp_{2x} \\ pp_{3x}a_1 + pp_{3y}a_2 + t_x &= cp_{3x} \\ pp_{1x}a_3 + pp_{1y}a_4 + t_y &= cp_{1y} \\ pp_{2x}a_3 + pp_{2y}a_4 + t_y &= cp_{2y} \\ pp_{3x}a_3 + pp_{3y}a_4 + t_y &= cp_{3y} \end{aligned}$$

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:

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

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



(a) (b)

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

## 4. DEVELOPED SOFTWARE AND RESULTS

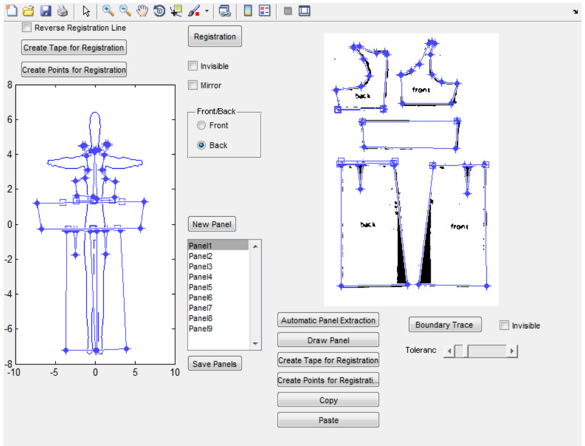
To test the effectiveness of the developed proposed approach in this research work, it was necessary to design a user friendly program to facilitate obtaining the results and visually inspecting them. The developed software package has been written in Matlab [15] and Maxscript [12]. The developed software takes as an input the 3D character models. Then it extracts the cross-section and saves it in a file, see Figure 3.

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 4. The user presses the New Panel button to start creating a new panel. Then, the user presses the Automatic Panel Extraction button and clicks in any position on or near the boundary of the panel that he/she wants to extract. 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 the nonreflective similarity 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 nonreflective similarity 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 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 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. Figure 5(a) shows a scanned cloth pattern image. For each panel, the user clicks in any position on or near the boundary of the panel that he/she wants to extract it. Then, the proposed approach automatically extracts it, as shown in Figure 5(b). A comparison is held between the proposed approach and the manual approach. Figure 5(c) shows the boundary of the panels generated manually and Figure 5(d) shows the overlap between the proposed approach and the manual approach. The difference between the proposed approach and the manual approach is computed through calculating the average of the Euclidean distance between points of the manual generated curve and the curve generated applying the proposed method. Table 1 presents the calculated difference for each panel in Figure 5. It is clear from Table 1 and Figure 5(c) that the difference between the panels generated manually and the panels generated using the proposed approach is very small. Figure 6 shows the fitting of the cloth pattern in Figure 5 to different human models using the proposed panel resizing approach to fit all the panels. To construct the 3D garments, the panels are sewed together and cloth simulation is executed to assist the deformation of the panels into the character model. One more cloth pattern is given in Figure 7(a). Figure 7(b) and 7(c) to show the boundary of the panels which is extracted using the proposed approach and the boundary of the panels which is generated using the manual approach, respectively. Figure 7(d) indicates the overlap between the proposed approach and the manual approach. Table 2 presents the calculated difference for each panel in Figure 7. Figure 8 shows the fitting of the cloth pattern in Figure 7 to different human models using the proposed panel resizing approach to fit all the panels. Then, the panels are sewed together to construct the 3D garments.

## 5. CONCLUSIONS

In this paper, we presented a novel approach for automatic extraction of panels from cloth pattern images and producing different size patterns from a given pattern size. A software package has been developed, and we have 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. Moreover, the proposed approach can be used for producing softcopies of patterns in old books and magazines. Thus, the proposed approach is consequently very helpful for computer generated movies, computer games, and cloth design industry.



☐ Reverse Registration Line

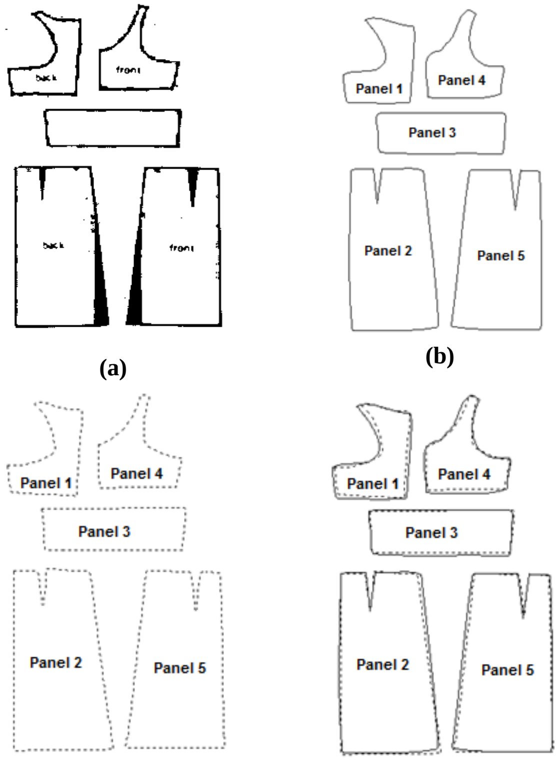
Front/Back: ○ Front ● Back

☐ Invisible

☐ Mirror

☐ Invisible

**Fig. 4: Main window of the developed software.**



(a) (b)

(c) (d)

**Fig. 5: (a) The original scanned patterns. (b) The boundary of the panels extracted using our proposed approach. (c) The boundary of the panels generated manually. (d) Overlap between (b) and (c).**

**Table 1. Difference between the proposed approach and the manual approach for each panel in Figure 5.**

| Panel No. | The difference |
|---|---|
| 1 | 2.2962 |
| 2 | 2.4279 |
| 3 | 1.7239 |
| 4 | 1.8367 |
| 5 | 2.7130 |

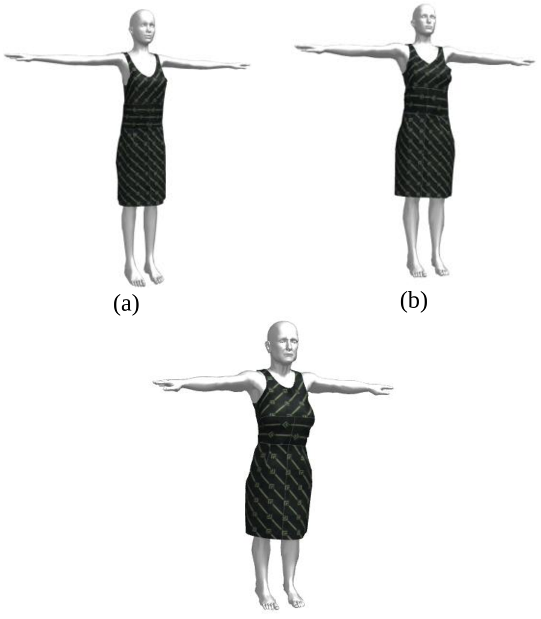


**Fig. 6: The fitting of the cloth pattern in Figure 5 to different human models.**

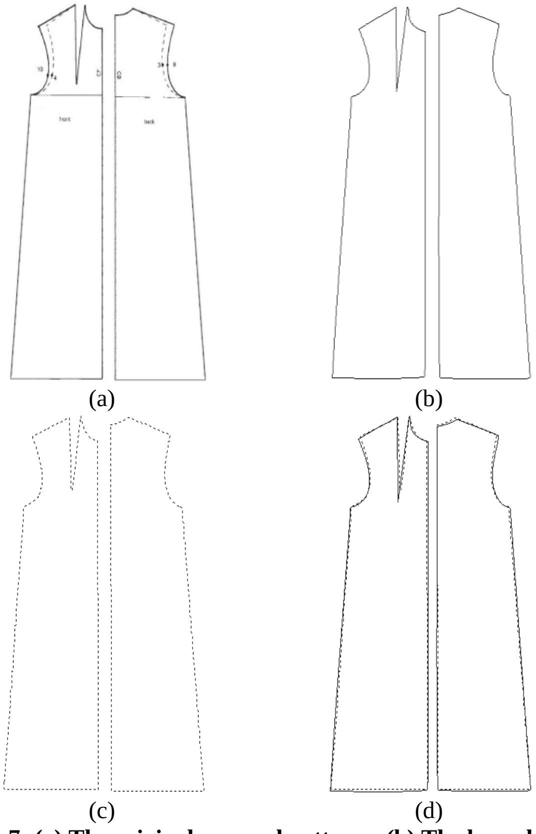


**Fig. 7: (a) The original scanned patterns. (b) The boundary of the panels extracted using our proposed approach. (c) The boundary of the panels generated manually. (d) Overlap between (b) and (c).**

**Table 2. Measuring the difference between our approach and the manual approach for each panel in Figure 7.**

| Panel No. | The difference |
|---|---|
| 1 | 2.2310 |
| 2 | 1.8870 |

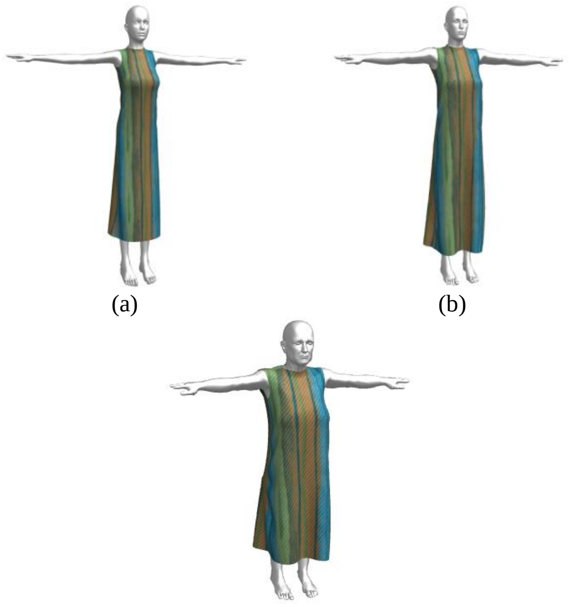


**Fig. 8: The fitting of the cloth pattern in Figure 7 to different human models.**

## 6. REFERENCES

[1] Kaldor, J., James, L., and Marschner, S. 2010. Efficient Yarn-based Cloth with Adaptive Contact Linearization. ACM Transactions on Graphics, 29(3), 205:1-105:10.

[2] Baraff , D., and Witkin, A. 1998. Large steps in cloth simulation. SIGGRAPH '98 Proceedings of the 25th annual conference on Computer graphics and interactive techniques.

[3] Choi, K. J., and Ko, H. S. 2002. Stable but responsive cloth. ACM Trans. Graph. (SIGGRAPH Proc.), 21:604-611.

[4] Carignan, M., Yang, Y., Magenenat-Thalmann, N., and Thalmann, D. 1992. Dressing animated synthetic actors with complex deformable clothes. In Proc. of SIGGRAPH’92. ACM, New York, NY, USA, 99–104.

[5] Breen, D., House, D., and Wozny, M. 1994. Predicting the drape of woven cloth using interacting particles. In Proc. of SIGGRAPH’94. ACM, New York, NY, USA, 365–372.

[6] Eberhardt, B., Weber, A., and Strasser, W. 1996. A fast, flexible, particle-system model for cloth draping. In IEEE Computer Graphics and Applications’96. IEEE, 52–59.

[7] Volino, P., Courchesne, M., and Thalmann, N. M. 1995. Versatile and efficient techniques for simulating cloth and other deformable objects. In Proc. of SIGGRAPH'95. ACM, New York, NY, USA, 137–144.

[8] Weil, J. 1986. The Synthesis of Cloth Objects. In Proc. of SIGGRAPH'86. ACM, New York, NY, USA, 24, 243-252.

[9] Eischen, J.W., Deng, S., Clapp, T.G. 1996. Finite-Element Modeling and Control of Flexible Fabric Parts. Computer Graphics in Textiles and Apparel (IEEE Computer Graphics and Applications), 71-80.

[10] Protopsaltou , D., et al. 2002. A Body and Garment Creation Method for an Internet-Based Virtual Fitting Room. Advances in Modeling, Animation and Rendering, J. Vince and R. Earnshaw, eds., Springer-Verlag, 105-122.

[11] Wang, C., Wang, Y., and Yuen, M. 2003. Feature based 3D garment design through 2D sketches. Comput Aid Des., 35(7), 659–72.

[12] Autodesk 3ds Max. http://usa.autodesk.com/3ds-max/, [Accessed Nov. 2011].

[13] Li, J., Ye, J., Wang, Y., Bai, L., and Lu, G. 2010. Technical section: Fitting 3d garment models onto individual human models. Comput. Graph. 34 (December), 742–755.

[14] Cordier, F., Seo, H., and Magnenat-Thalmann, N. 2003. Made-to-measure technologies for an online clothing store. IEEE Computer Graphics and Applications 23 (Jan), 38–48.

[15] MATLAB. http://www.mathworks.com, [Accessed Nov. 2011].

[16] Xu, C., and Prince, J. L. 1998. Snakes, shapes, and gradient vector flow. IEEE Transactions on Image Processing, VOL. 7, NO. 3, 359 – 369.

[17] Harris, C., and Stephens, M. J. 1988. A combined corner and edge detector. In Alvey Vision Conference, 147-152.