A New Image Warping Technique using Mesh Patterned Textile

Yong-Bin Kang, Young-Jung Yu, Hwan-Gue Cho*

Abstract

In this paper we propose an efficient image warping technique to simulate arbitrary texture images onto real clothings. Till now lots of apparel modeling tools were announced to help clothing designers. Previous works were done mainly on modeling for the fabric dynamics to simulate a virtual clothing to a virtual model. However their systems require too much computations and additional interventions of human designer.

Our system can warp a given textile image onto arbitrarily wrinkled clothes on which mesh patterns are printed. The proposed warping algorithm utilizes those mesh points to compute the wrinkles of cloths. For this work, one graph searching algorithm is applied to decide the topology of the mesh points appeared in cloth. Our experimental results show that this technique is highly effective to warp textile images to an arbitrary clothing without any manual works and 3-D information for the cloth surface.

Keywords: geometric modeling, texture warping, cloth model

1 Introduction

1.1 Previous works

Several models were already proposed to simulate a real cloth [2]. All problems in 3-D cloth modeling are reduced to ask how to find 3-D information from 2-D image or range data. Lots of works were already published on this problem[7, 8, 9, 10]. Till now a few static cloth simulation tools were announced, e.g., Computer Designer's U4ia, ModaCAD, Infografica's meta-reyes, BRAIN's TEX-SIM, Lumena, and MicroMap[1]. TEX-SIM and ModaCAD are intended to create fabric design and generate its realistic images easily. Comparing with ModaCAD, Super TEX-SIM can generate pre-dyeing weaves in an interactive manner. Also Super TEX-SIM creates fabric images themselves using computer graphics, whereas ModaCAD do not define how to generate such images used for texture mapping on a human figure[3]. And P-CAD of Shima seiki's are used to design and manipulate texture images.

One of common features in TEX-SIM, ModaCAD and P-CAD[6] is that all functions are performed with some manual works. Due to this constraint, if clothes have lots of

*This paper was supported by Korea Research Foundation(1996). Graphics Application Laboratory, Department of Computer Science, Pusan National University, Kum-Jung-Ku, Pusan 609-735, Korea. E-mail: {ybkang,yjyu,hgcho}@pearl.cs.pusan.ac.kr

wrinkles, then they cannot map easily on that clothes. MicroMap decomposes the image into only two basic components, plane and cylinder. And that system blends those basic geometric shapes to approximate given cloth surface. Recently others announced U4ia to visualize fabric for various clothing and home furnishing model[1]. U4ia needs lots of small polygon patches to construct NURBS modeling with the help of designer's hand work. Though this modeling method gives a good performance, but that is believed to be time-consuming and costs very expensive.

This paper considers the static fabric simulation using 2-D image warping only. In order to overcome disadvantages of those previous fabric/clothes simulation tools, we newly propose an automated mapping technique to give a realistic look without any manual interventions.

1.2 System overview

Our system will warp a given textile pattern onto a real cloth model, which is given in the form of photographic data. At first, we draw a mesh pattern on a clothing(white T-shirt in our experiment) to simulate a mesh pattern textile. Points and edges of the mesh are colored red and blue, respectively, which helps to isolate points and edges from the photo image.

Input data to this warping system are two entities. One is a photo showing a human model wearing clothes of mesh patterns. Another is a patterned textile image to be warped onto the cloth. Output will be the warped image where the given textile pattern is mapped into the clothes as if a human model wears that clothes with a given pattern. Our system consists of five major steps.

Step 1: (a) Read a photo P into an image file I;
(b) Extract the mesh image(points and edges) I_M and shadow image I_P from I;

Step 2: (a) Find the incomplete mesh image graph(for short, image graph) G_I by scanning I_M and separating points from edges;
(b) Make graph G_I to be a connected graph G_C by adding some extra edges;
(c) Determine mesh topology G_P by assigning index $[i, j]$ of each mesh point of G_C;

Step 3: Add hidden points and edges of G_P, which are not appeared in the original photo P;
/* This enables G_P to be a complete mesh topology isomorphic to those mesh(original mesh) drawn on the textile(e.g., mesh on T-shirt in our experiment). Let G_F denote this fully connected mesh graph */

Step 4: Finally we warp the textile pattern image into the G_F by mapping each corresponding sub-mesh region using bilinear image transformation;
/* Let P_W denote this mapped image */

Step 5: Blend the shadow image I_P with the warped image P_W on G_F to give realistic 3-D effects;

2 Image Graph Construction from Mesh Image

Let us introduce some preliminaries for the following sections. *Mesh point(edge)* is a solid point(edge) appeared in the original mesh pattern on textile. *Pseudo edge* is an edge which looks like a connected edge in wrinkled cloth, but this is not an edge of original mesh. *Supplement point(edge)* is a point(edge) added to make a complete mesh topology. *Hidden point(edge)* is a point(edge) which is not visible point(edge) since it is hidden in the folded textile. *Corner points* are four corner of the original mesh textile. See Fig.1(a) for these terms.

It is easy to see that every *hidden point(edge)* in photo image should be replaced with a *supplement point(edge)* to make a complete mesh topology. If clothes have wrinkles, there would be some hidden points and edges in G_I. So G_I does not represent the complete mesh structure on photo image. Note that G_I consists of mesh points, edges and has a few *pseudo edges* due to wrinkles of textile.

Fig.1(b) shows G_I obtained from a photo image given in Fig.1(a). As was shown in Fig.1(b), a line segment (e, f) is identified as a solid edge in the photo image file since e and f mapped to be closely located with edge connection. So (e, f) is one of pseudo edges. In G_I, verifying pseudo edges is very important because pseudo edge gives information about the folding structure of textile. Therefore we can estimate foldings by investigating hidden points and pseudo edges.

For constructing G_I, we have to decide mesh index $[i, j]$ for all points in G_I. In order to determine the mesh topology of G_I, at first, we have to isolate mesh points and edges from an input photo image. Mesh points can be easily located since they are "red" pixels. If visible points are identified by image processing, it is easy to identify mesh edges by visiting the mesh point and looking around its neighborhood pixels to find four connecting edges. However we can not detect all visible points on photo image. because a photo has noises due to the distortion of scanning process and degeneration of colors for lighting or shading effects.

In this process, if mesh index is not matched properly, we encounter *conflict* in mesh index assigning. With the correctly identified mesh points and pseudo edges, we can complete the mesh topology including pseudo edges. Conflict edge means an edge which is identified by two points with distance more than two. Let $e(a, b)$ be an edge of G_I, and $x[i, j]$ denote index $[i, j]$ of points x. If these two adjacent points are indexed as $a[i, j]$, $b[i+2, j+1]$, then this point a and b have a conflict index, since two adjacent points of mesh could not be separated in distance 2. Index conflict happens when some points of mesh are hidden in the foldings of textile. Therefore G_I would be an incomplete mesh by the conflicts on mesh index. Fig.2 shows a real photo image and its corresponding image graph. Fig.2(b) shows that there are some missing edges along wrinkles. In the next section, we will give a greedy graph algorithm to resolve these conflicts.

3 Determining Mesh Topology

In previous sections, we explained how to get G_I from a photo image. G_I consists of more than two disconnected components and has hidden points and edges. Since it is easy to warp a texture pattern into mesh structure using two topologically equivalent graph, we need to make a complete mesh topology from G_I.

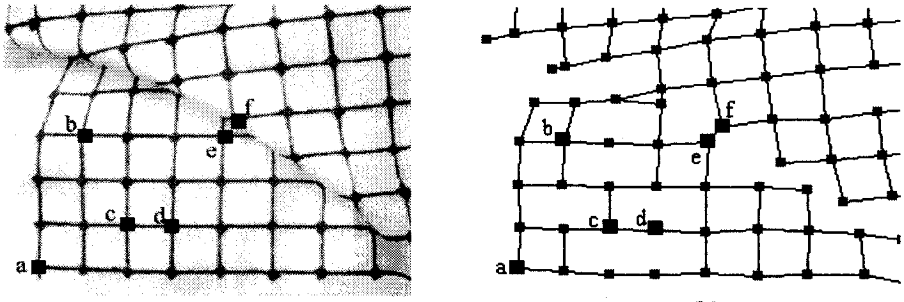


Figure 1: (a) corner point a; mesh point a, b, c, d, e, f; a mesh edge (c, d); a pseudo edge (e, f) , (b) the corresponding image graph of Fig.1(a)

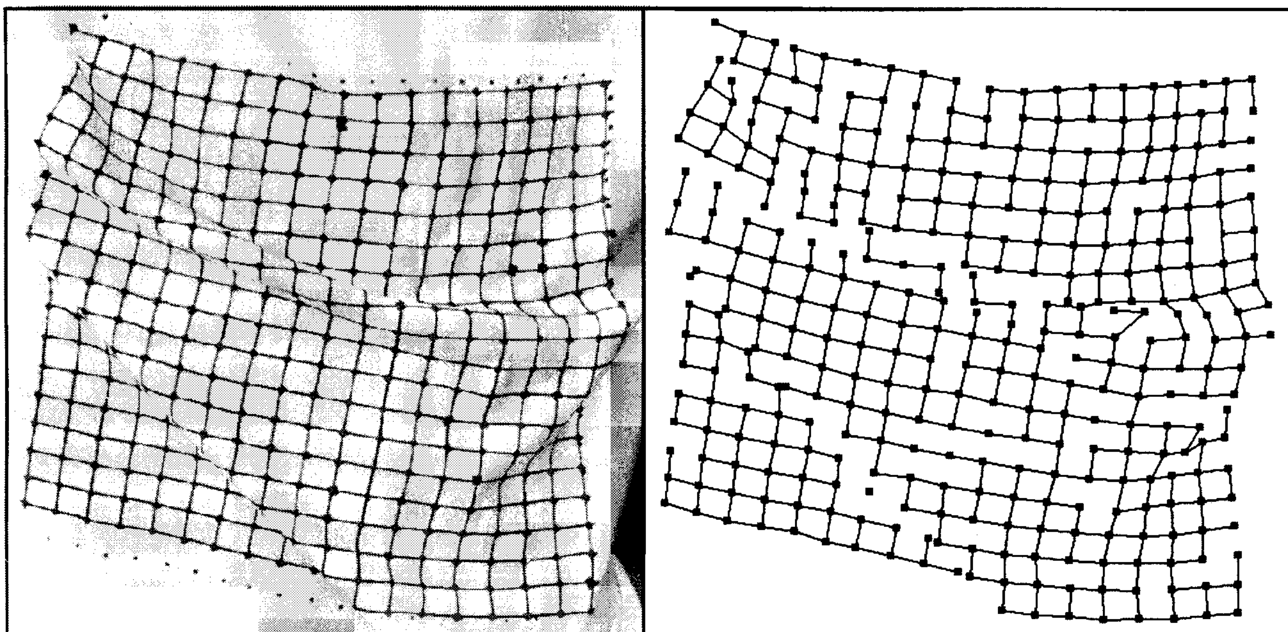


Figure 2: (a) original image, (b) the corresponding image graph of (a)

Three steps are needed to get G_F. The first step is to convert G_I into G_C. Converting G_I into G_C is not trivial problem, since G_C should reflect the original mesh pattern on clothing by examining index conflicts.

The second step is to construct a mesh topology G_P from G_C. But this mesh topology is incomplete because there are a few hidden points and edges. Constructing complete mesh topology from incomplete image graph needs a good heuristics to construct a complete mesh topology.

The final step is to construct a complete mesh topology graph G_F from an incomplete mesh topology G_P by adding some supplement points and edges to reveal hidden points and edges.

3.1 Phase 1 - Constructing a connected component

G_I obtained from an original photo image has many distortion due to noises of scanned photograph or wrinkles of cloth. So G_I may have more than one connected component.

If there are more than two disconnected components in G_I, we have to add extra edges in a greedy fashion. Our system prefers an edge whose addition connects two separate components of G_I and its length is the closest to that of a unit mesh.

3.2 Phase 2 - Building complete mesh topology

Our system assumes some constraints to build a mesh topology from G_I easily.

assumption 1 *At least the one corner point is visible in P.*

assumption 2 *Mesh pattern is uniform : all sub-meshes are of equal size.*

assumption 3 *Textile with a mesh pattern cannot be stretched or compressed.*

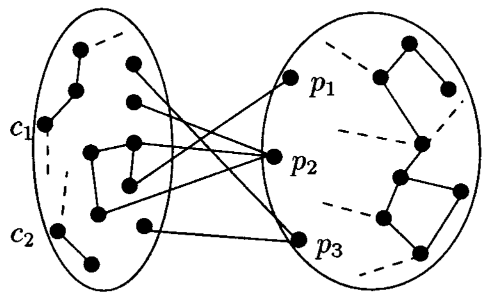


Figure 3: p_2 is moved to KNOWN, since it is highly connected to points with fixed index(KNOWN)

Initially, we put all corner points in buffer KNOWN. And other points except corners are placed in buffer UNKNOWN. We select the point in UNKNOWN that has the highest connection to KNOWN. If there are ties in connection degree to KNOWN, we choose a point with an edge whose length is the closest to the size of unit mesh. The index of a selected point is determined by ASSIGN_INDEX() function.

Let us describe a graph searching algorithm for constructing mesh G_P from G_C.

1. KNOWN := corner points of index $[i, j]$ assigned;
   UNKNOWN := all other points G_C without index;

2. /* Find the most closely related point in UNKNOWN to KNOWN. */
   Let $N(y) = \{x | (x, y), x \in \text{KNOWN}, y \in \text{UNKNOWN}\}$.
   We select the point in UNKNOWN with the highest value of $|N()|$.

3. Let $dist(x, y)$ be the eucledian distance between two points x, y.
   For all v that has the same value of $|N(v)|$, we will apply the tie-breaking rule.
   $S(v) := \text{Length(unit edge of } G_I) - (\sum_{x \in N(v)} dist(v, x)/|N(v)|)$;
   Select a not indexed point b satisfying $b = \min\{S(h) | h \in \text{UNKNOWN}\}$;
   Remove b from UNKNOWN, and move b to KNOWN;

4. We assign index $[i,j]$ of x considering its neighborhood points in KNOWN.
   $x[i,j] :=$ ASSIGN_INDEX($\{y|(x,y) \in E(G_C), x \in \text{KNOWN}\}$);

5. if (UNKNOWN is empty) then exit;
   else goto step 2;

Fig.3 is a snapshot in determining mesh topology $x[i,j]$ for all $x \in$ UNKNOWN. In the situation of Fig.3, p_2 should be selected since it has the highest $N(\ )$ value among points in UNKNOWN.

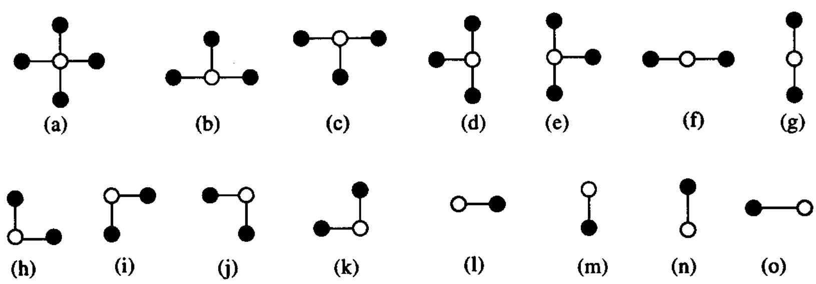


Figure 4: White points denote one supplement point to be inserted to G_C.

Now we explain the function ASSIGN_INDEX for G_I. Since we choose the highest-degree point to confirm $[i,j]$ index in greedy fashion, we should consider all possible cases. There are four types of points in step(2) according to $N(\ )$ value. Determining mesh topology of a selected point of degree 4 or degree 3 is very easy, since it has only two adjacent vertical or horizontal neighborhoods. For example, if a point p has two vertical neighbor points x and y, where $x[i,j+1]$ and $y[i,j-1]$, then p should be indexed as $p[i,j]$. Determining mesh topology of a point of degree 2 is ambiguous. In Fig.4, (h) and (j), (i) and (k) denote several ambiguous situations. For example, suppose that x is adjacent to $y[5,4]$ and $z[6,5]$, then $x[6,4]$ and $x[5,5]$ are possible. To remove this ambiguity we consider the physical location of x with respect to y and z in image domain. If a selected point x has $N(x) = 1$, then we cannot determine its mesh index. Then we do not estimate its index till one of its neighborhood points have degree more than 1.

3.3 Phase 3 - Constructing complete mesh topology

Though all visible points are indexed by the previous algorithm, hidden points and edges are not included in G_P yet. So we must add some points (=supplement points) into an appropriate location to make mesh topology complete.

We add supplement points one by one greedily to reveal more accurate clothing structure. If there are more than one supplement points to be added, determining the order of point insertion sequence is important. Now we give another algorithm to supply hidden points and edges in G_I with respect to the complete mesh.

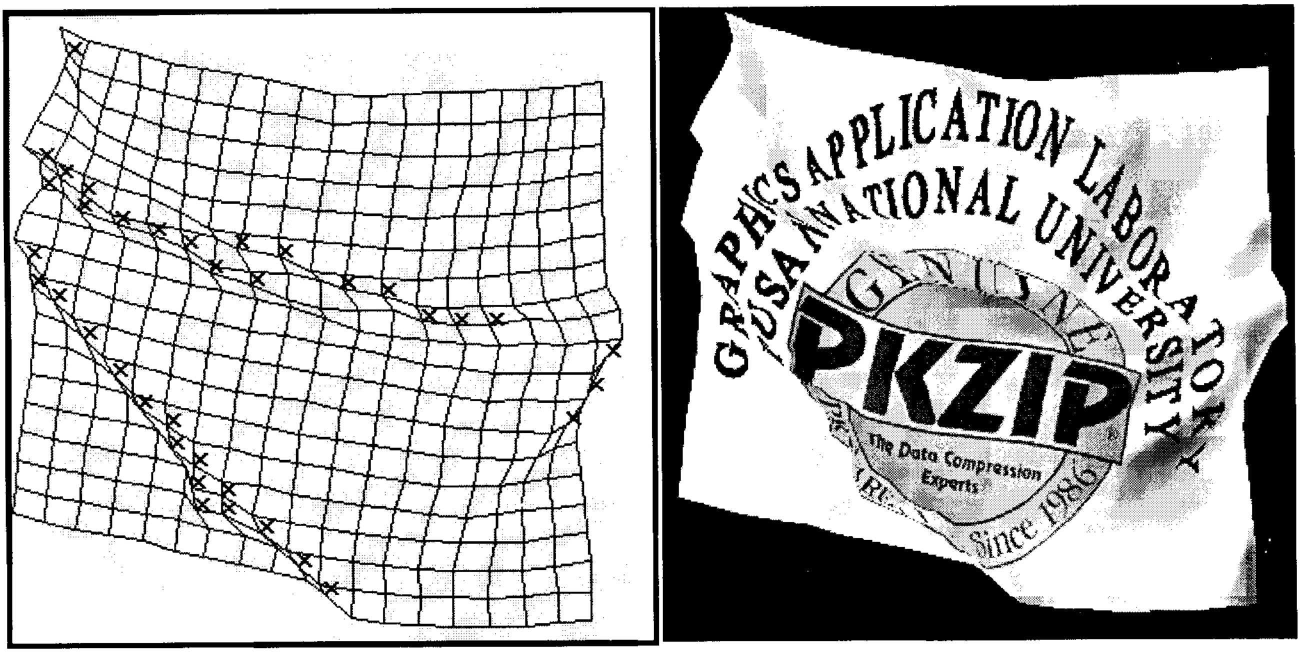
1. Let x be a supplement point to be added. And HIDDEN and MESH denote a set of hidden points and original mesh, respectively.
   Let $M(x) = \{v|(v,x) \in \text{E(MESH)}\}$.

2. For all points $h \in$ HIDDEN,
   $L(h) :=$ the center of $N(h)$, where $L(h)$ is the physical location of point h in image;
   $S(h) :=$ Length(one unit size of mesh edge in G_I) $- \left(\sum_{v \in M(h)} dist(h,v)/|M(h)| \right)$;

3. Find a hidden point b satisfying $b = \min\{S(h) | h \in \text{HIDDEN}\}$;
   Remove b from HIDDEN;

4. If we have a complete mesh, then exit;
   otherwise goto step(2);

Fig.5(a) shows the corresponding G_F of Fig.2(a). In Fig.5(a) the mark 'X' represents the supplement points to compensate each hidden points. Fig.5(b) shows the finally mapped image.



(a) (b)

Figure 5: (a) the corresponding mesh of Fig.2(a), (b) the finally mapped image of Fig.2(a)

4 Experiments

In the previous sections, We gave an algorithm to construct a mesh topology from an image graph. Up to this procedure we have constructed a complete mesh topology from G_I in the photo image. We are ready to simulate various textiles using a real cloth photo images. At first, we divide the textile according to the original mesh dimension in the clothing. And then we warp the textile into mesh topology sub-mesh by sub-mesh. In order to produce more realistic 3-D textile effects, we blend a warped image with the previously extracted shadow image in the original photo.

Our algorithm has been tested on various photo images and textile patterns. We implemented our system with C using the Motif/Mesa library, and it works on Sun SparcStation-10. 1200dpi scanner was used to obtain the photo image file from a photo.

For an experimental clothing, we prepared one white T-shirt and printed the mesh pattern with red(point) and blue(edge) color mark-up pens.

Fig.6 - Fig.7 show original photo images and their corresponding image graphs and results of mapped clothes. Each original image has different features. That is, Fig.6(a) has four corner points but it has many hidden points and edges. Fig.7(a) is a more complicated mesh structure than Fig.6(a), where one corner point is hidden and that textile is wrinkled deeply. Fig.6(b) represents the corresponding image graph of Fig.6(a). Fig.6(c) ,Fig.7(b) and Fig.7(c) show the results of finally mapped image with various textiles. And in order to simulate the real clothes effect, we synthesized a mapped image with original photo image using the different textures(See Fig.8). Our experimental results show that the proposed technique gives a realistic mapping effects for 3-D clothing model without any real 3-D information and human works.

5 Discussion

In this paper we presented a new textile warping technique for real clothing. To give conclusions, our system has the following features.

- Users can simulate their textile images on a real cloth in image domain without any manual interventions.
- We give real 3-D effects including shadow, wrinkles, discontinuity of textile image without any 3-D information of clothing.
- We do not need any polygonal patches(e.g., NURBS)to model clothing, which is a widely-used methodology and is believed to be a computational hard work.
- We newly introduced an interesting graph algorithm to determine mesh index only by considering a few neighborhood points, which starts with visible corner points.

Experimental results show that this greedy algorithm determines mesh topology effectively and could find hidden points and edges. Thus we can easily find wrinkled region by analyzing the location of hidden points, edges, conflict indexes in image graph. These are clearly illustrated in Fig.6 - Fig.8.

We give interesting research problems to improve our warping techniques.

- How to extract a 3-D(height) information from an image graph ? If we can estimate the height of each mesh points, then we can make a more realistic 3-D model.
- What kinds of patterns in a cloth will be helpful to decide the topology of that pattern ? We expect a triangular mesh will be more helpful to get information of folded cloth dynamic.
- What is the optimal mesh dimension (n, m) ? We believe that too dense mesh pattern increases the number of pseudo edges, which results lots of conflict in determining mesh index $[i, j]$, while too sparse mesh pattern do not reveals the structure of a real cloth in detail.

We hope that our mesh index searching algorithm will find another practical applications in CAD field.

References

[1] D.P. Mahoney, *The fabric of Animation*, Computer Graphics World, Aug. pp.29-34, 1996

[2] H.N.Ng and R.L. Grimsdale, *"Computer Graphics Techniques for Modeling Cloth"*, IEEE Computer Graphics And Application Sep. Vol.16, No. 5, pp. 28-41 , 1996.

[3] H. Kambe, N. Yamasaki and M.Y. Lee, *SUPER TEX-SIM* Interactive textile design and simulation, Proc. of Pacific Graphics, 1993.

[4] G. Wolberg, *Digital Image Warping*, IEEE Computer Society, Los Alamos, CA, 1990.

[5] P.S. Heckbert, *Survey of Texture Mapping*, IEEE CG&A, Vol. 6, No. 11, pp.56-67, Nov. 1986.

[6] H. Kambe, *A Simulation System for Pre-Dyeing textiles*, PIXEL, No. 62, pp.91-94, in Japanese.

[7] C. Bennis, J.M. Vezien, G. Iglesias, *Piecewise Surface Flattering for Non-Distorted Texture Mapping*, Computer Graphics, Vol. 25, No. 4, pp. 237-246, Jul. 1991.

[8] F. Ulupinar, R.Nevatia, *Perception of 3-D Surfaces from 2-D Contours*, IEEE Transaction On PAMI, Vol. 15, No. 1, Jan. 1993.

[9] D. Blostein, N.Ahuja, *Shape from Texture: Integrating Texture-Element Extraction and Surface Estimation*, IEEE Transaction On PAMI, Vol. 11, No. 12, Dec. 1989.

[10] F.S. Cohen, J.Y. Wang, *Part1:Modeling Image Curves Using Invariant 3-D Object Curve Models-A Path to 3-D Recognition and Shape Estimation from Image Contours*

[11] Arad, Nur, N. Dyn, D. Reisfeld, and Y. Yeshurun, *Image Warping by Radial Basis Functions: Applications to Facial Expressions*, Graphical Models and Image Processing, Vol.56, No. 2, pp. 161-172, Mar. 1994.

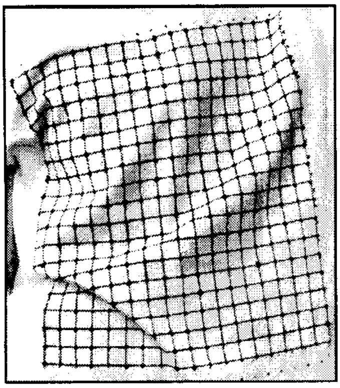


Figure 6(a)

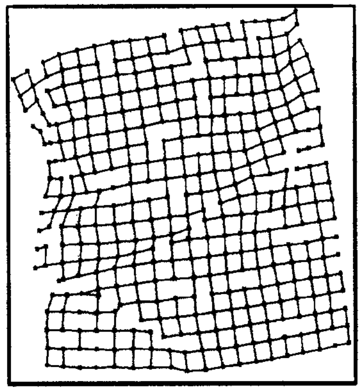


Figure 6(b)

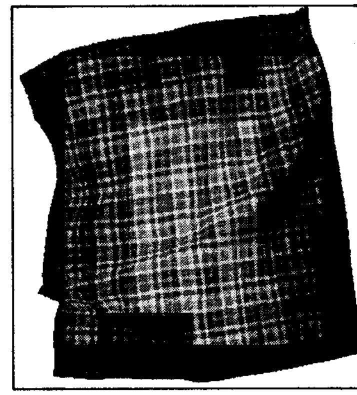


Figure 6(c)

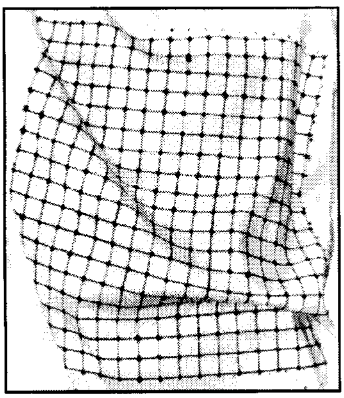


Figure 7(a)

Figure 7(b)

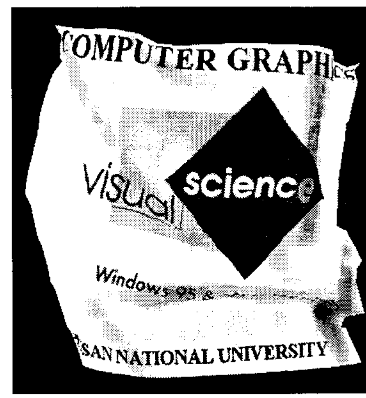


Figure 7(c)

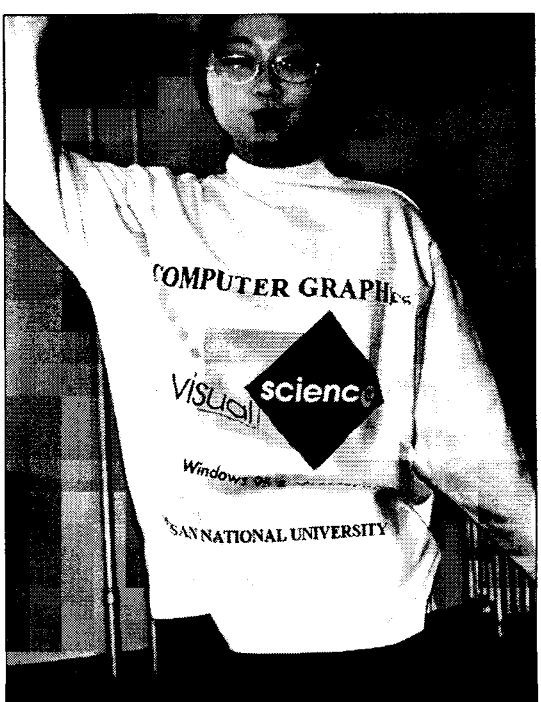


Figure 8(a)

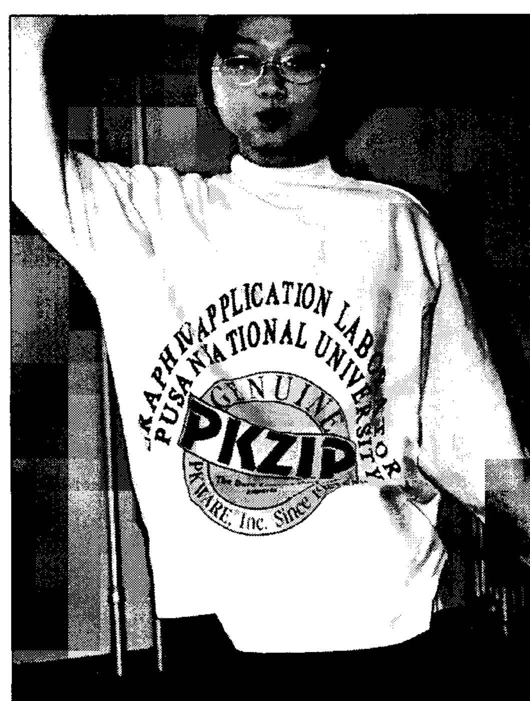


Figure 8(b)