Copyright © 17th Int'l. Conf. on Pattern Recognition, vol. 4, pp.155-158, Cambridge UK, August 2004

Detecting Deformable Objects with Flexible Shape Priors

Tien-Lung Chang Tyng-Luh Liu
Institute of Information Science, Academia Sinica
Nankang, Taipei 115, Taiwan

Abstract

We address the problem of detecting objects/shapes with large deformation and articulation in cluttered images. The approach requires a shape prior that describes the approximated outline and articulation property of a given model. While dynamic programming is often used in solving shape detection, our focus is on formulating a more effective energy function to evaluate the optimality of a matching between the shape prior and image features. For efficiency, the detection via optimization is carried out over a non-uniform elastic grid based on referencing the edge information. Experimental results are included to illustrate our method.

1. Introduction

Detecting objects (or shapes) with large deformation and articulation is an important and difficult task in vision research. While exploiting global non-rigid transformations alone for finding similar counterparts of a given model in image would be infeasible, the problem can often be alleviated with a proper object representation, and then reasonably solved via an effective solution grouping process.

The recent work of Felzenszwalb [5] provides a good example that representing and detecting deformable shapes are accomplished by coupling a triangulated polygon representation with an elegant dynamic programming argument. Motivated by his work [5], we propose a more general approach to represent and detect shapes with *large* articulation and deformation. Specifically, the detection is carried out by referencing a given shape prior, of which knowledge about the flexibility of its part structures is specified, and by examining image features over a non-uniform elastic grid, where the solution is derived by energy minimization.

1.1. Previous work

A number of techniques for detecting deformable shapes have been developed over the years, including the very primitive work by Thompson [13], where he finds that two similar but not identical shapes can often match each other by a simple coordinate transformation. In [6], Fischler and Elschlager use an energy function to describe the deformation and the quality of the mapping between two shapes. It is a *recognition-by-parts* approach that the spatial arrangement of all parts is constrained by *springs*. Grenander [9] proposes the well-known pattern theory to describe complex shapes. Jain et al. [10] integrate the ideas from several previous works with their proposed deformable model and probability function to establish a systematic approach for shape detection. In addition, they use a coarse-to-fine scheme to speed up the detection. The same acceleration method is also adopted by Gavrilla and Philomin [7] to detect objects in a real-time vision process. However, they use multiple models to represent one deformable shape (with possible articulation) such as a pedestrian. The *snake* (or *active contour model*) introduced by Kass et al. [11] has been very successful, especially for processing medical images. Cootes et al. [4] later extend it to an *active shape model*, where the shapes are deformed under constraints learned from a training set. Zhao [14] has an interesting result in human shape detection, but her method needs presegmentations of images.

2. Shape Representation and Image Features

The most straightforward way to represent 2-D shapes is to describe them by their bounding contours. However, though such a representation has the advantages of compactness and simplicity, it captures only the local characteristics. On the other hand, representations designed to encode global properties of shapes may not be appropriate for detecting shapes with large deformation and articulation, since they would change the global appearances of shapes considerably. In [1], Amit et al. propose to use *graphical templates* and *dynamic programming* for elastic matching. These graphical templates satisfy the definition of a *decomposable graph*, and thus have the *running intersection property* for a dynamic programming implementation. More recently, Felzenszwalb [5] describes an efficient method to represent and detect deformable shapes based on a triangu-

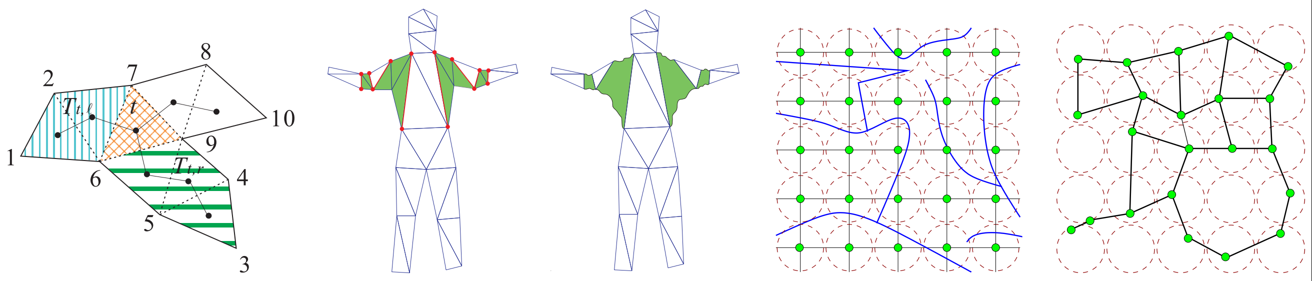

Figure 1. (a) The triangulation of a polygon and its dual graph. The numbers are an instance of a perfect elimination order. (b) A triangulated polygon human shape model. (c) A flexible human shape prior. (d) Uniform Cartesian grid (blue contours are edges). (e) Non-uniform elastic grid.

lated polygon model and a dynamic programming scheme. Since a triangulated polygon is also a decomposable graph–these two approaches are indeed closely related.

2.1. Shape priors and articulation

Similar to Felzenszwalb's work [5], we choose the triangulation of a polygon as our shape model because a representation of this kind combines the advantages of local description and region information in depicting a shape.

To construct a model, we first segment the shape from a silhouette. (Only 2-D shapes with no holes are considered.) Then, we compute a polygon and its triangulation to approximate the outline of the resulting shape. Typically, the number of vertices of a polygon needed for an adequate representation is much fewer than the number of points of the original shape [12]. In Figure 1b, it takes 36 and 8 vertices to represent the human shape and each arm, respectively.

While the triangulated polygon is useful in detecting deformable shapes, it still can not handle those with articulation. Nevertheless, the issue can be addressed properly by embedding prior knowledge into the model. Again, using Figure 1b and 1c to illustrate, if we aim to detect human shapes with possible arm articulation, the deformation penalty in the energy function (defined later) for those triangles in the areas that articulation occurs could be lessened to accommodate large changes. That is, these areas can be considered flexible regions of the resulting *shape prior*.

2.2. Image features and elastic grid

For the sake of computational efficiency, we carry out the detection task over a coarse grid of a testing image.

We apply the Canny edge detector [3] on an image I to derive a binary edge map, and then partition the edge map into a Cartesian grid of m' lattice points, say, spaced equally by a distance w. Then, at each lattice point, a circular region of radius $w/2$ is considered: If there exist edge points within the region, the lattice point will be replaced by the nearest edge point. Otherwise, the lattice point will be removed. We shall further assume that a final grid has m lattice points, and denote this set of grid points as Q. In Figure 1d, the original uniform Cartesian grid and the edges (those blue contours) are plotted together. The derived non-uniform elastic grid Q is shown in Figure 1e.

3. Detection via Energy Minimization

Each triangulated polygon implicitly defines a dual graph that has the *perfect vertex elimination scheme* [8]. This fact implies that whenever a vertex is deleted from a triangulation, exactly one triangle is eliminated and the remaining part is still a triangulation of some *simplified* polygon (see Figure 1a). This is an important property that guarantees the feasibility of dynamic programming in solving the problem.

Let P and T be a polygon and its (Delaunay) triangulation for representing a given shape prior, respectively. To account for articulation, we further divide the triangulation by $T = T_1 \cup T_2$, where T_2 contains those triangles located at the regions of possible articulation. An indicator function θ is defined to describe this classification, where $\theta(t) = 1$ if $t \in T_2$, and 0, otherwise. The problem to detect a similar shape in I can now be reduced to finding an *optimal* mapping $f : V \rightarrow Q \subseteq I$, where V denotes the set of all vertices of the polygon P. Then the mapping of the whole polygon can be estimated by interpolation. In the following, we formulate an energy function that would be used to quantify the optimality of a candidate mapping f.

3.1. Energy function

In shape matching, a typical energy function often has two terms: The first term is to measure the *deformation* between the correspondences induced by a mapping, and the second is to measure the *fitness* between the mapped features and salient features in the image.

Deformation cost: For every triangle $t \in T$, we want to estimate the cost induced by deformation between t and $f(t) \in I$. (We have somewhat abused the notation here, but it is clear that $f(t)$ denotes the mapped triangle obtained by interpolations between the mapped vertices.) Amit et al. [1] have constructed a deformation cost between two triangles by measuring sum of squared differences of the *log-ratios* for the corresponding lengths. In [5], the *log-anisotropy* cost is used for the same purpose. Since both criteria give comparable detection results in our experiments, we will simply denote them as C_ℓ. The C_ℓ is scale-invariant in evaluating the deformation cost between t and $f(t)$. However, it does not guarantee to yield a target shape satisfying $t_i : f(t_i) \approx t_j : f(t_j), \forall t_i, t_j \in T$. We define below a *uniformity factor* U to reinforce this constraint.

$$U(t, T_{t,\ell}, T_{t,r}, f) = \\ \prod_{s,s' \in \{t, T_{t,\ell}, T_{t,r}\}, s \neq s'} \frac{\max(A(f(s))/A(s), A(f(s'))/A(s'))}{\min(A(f(s))/A(s), A(f(s'))/A(s'))},$$

where $T_{t,\ell}$ and $T_{t,r}$ are the two subgraphs adjacent to t and all the triangles in them should be eliminated before t according to the perfect elimination scheme (see Figure 1a). $A(s)$ is the area of the region, excluding those triangles in T_2, enclosed by s. Finally, the deformation cost C_D between t and $f(t)$ is

$$C_D(t, f) = \theta(t) \times U(t, T_{t,\ell}, T_{t,r}, f) \times C_\ell. \tag{1}$$

Fitness cost: There are many different measures to evaluate the fitness of a mapped $f(t)$ and the image features. We consider a typical normalized snake model:

$$C_s(f(u), f(v)) = \frac{1}{|f(u)f(v)|} \int_{f(u)}^{f(v)} [\frac{1}{E(x)} + \beta \cdot k(x)]\, dx, \tag{2}$$

where $E(x) = 1$ if pixel x is an edge point, and 0.01 (a rather small number relative to 1), otherwise. Furthermore, $k(x)$ means the curvature at x, and β is a parameter. Let the three vertices of a triangle t be u, v, and w. Then the fitness cost can be defined as follows.

$$\begin{aligned} C_F(t, f) = C_s^{'}(f(u), f(v)) + C_s^{'}(f(v), f(w)) \\ + C_s^{'}(f(w), f(u)), \end{aligned} \tag{3}$$

where $C_s^{'}(f(u), f(v)) = C_s(f(u), f(v))$ if (u, v) is an edge of the polygon P, and 0, otherwise.

With (1) and (3), we are ready to define the energy function C and the optimality of f:

$$f^* = \underset{f}{argmin}\, C(f; T) = \sum_{t \in T} COST(f, t), \tag{4}$$

where $COST(f, t) = C_D(f, t) + \lambda C_F(f, t)$, and λ is a parameter to balance the importance between C_D and C_F. In passing, we note that other distance measures, such as the *Chamfer distance* [2], may also be used for defining a proper fitness function (see Figure 2m and 2n).

3.2. Detection through dynamic programming

The optimization problem (4) can be solved via dynamic programming [1], [5]. We will follow the formulation described in [5] to illustrate the solution grouping process. Let $\{v_1, v_2, ..., v_n\}$ be vertices of the triangulation T indexed by the perfect vertex elimination order. It follows that, at the ith eliminating stage, v_i is the vertex to be removed and belongs to exactly one triangle of the remaining triangulation. Let v_j and v_k be the other two vertices of the same triangle of v_i, of which the two indices can be referenced from two pre-defined functions g_j and g_k, i.e., $j = g_j(i)$ and $k = g_k(i)$. Then, v_i must appears before v_j and v_k in the elimination order. Also, v_i, v_j, and v_k are assumed to be positioned in a counter-clockwise manner. Form Algorithm 1, it is clear that $V[j,k](q,r)$ records the optimal cost for mapping the polygon(s) formed by the i deleted triangles to a subset of Q, subject to the constraint that $f : v_j, v_k \mapsto q, r$.

Algorithm 1: Dynamic Programming [5]

```
  for $q, r \in Q \wedge (v_i, v_j) \in edge(P)$ do
      $V[i,j](q,r) = 0$
  for $i \leftarrow 1$ to $n-2$ do
      $j \leftarrow g_j(i)$
      $k \leftarrow g_k(i)$
1     for $q, r \in Q$ do
2         $V[j,k](q,r) = \min_{p \in Q} COST(i,j,k,p,q,r)$
                  $+V[i,j](p,q) + V[k,i](r,p)$
  Find $q, r \in Q$ minimizing $V[n-1,n](q,r)$, and then
  backtrack to get the mappings for all other vertices.
```

To reduced the computational complexity, in flag 1 of Algorithm 1, we select only those position pairs (q, r) such that $0.8 \cdot |v_j v_k| \leq |qr| \leq 1.2 \cdot |v_j v_k|$ if $v_j v_k$ is an edge of some triangle from T_1. ($|v_j v_k|$ is the length of the edge.) Moreover, the best position of p stated in flag 2 is predicted by a similarity transformation, which maps v_j and v_k to q and r respectively, and the searching of p is taken only in the neighborhood of the estimated position. The two simplifications reduce the time complexity to $O(nm)$.

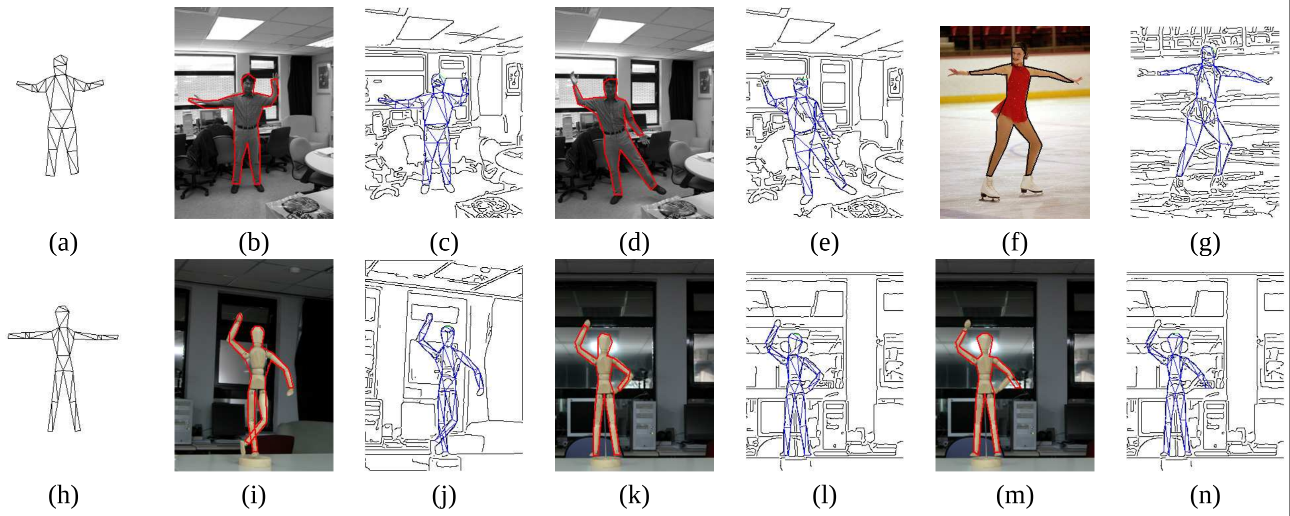

Figure 2. Two shape priors, (a) and (b), are used to detect shapes with articulation and deformation in images of the upper and the lower row, respectively. The detected shapes are plotted on top of their original images and also over the edge maps so that it is easier to evaluate the quality of solutions. Notice that the results in (m) and (n) are derived by replacing the *snake cost* **with the** *Chamfer distance* **in the energy function. In this case, some vertices of the left elbow and the connected forearm are mapped to unsatisfied positions.**

4. Experimental Results and Discussions

For each testing image, we create a grid with a spacing width $w = 6$ pixels. The edge map is made by a Canny edge detector using the default parameter settings in Matlab 6.5. Once the non-uniform elastic grid is available, we could initiate the dynamic programming process to find the optimal shape in the image. The values of the two parameters β and λ are set to $0.255 = 0.001 \times 255$ and $0.51 = 0.002 \times 255$, respectively. We report several results for detecting shapes with deformation and articulation in Figure 2. Notice that all the experiments are with complex backgrounds, and the detections are accomplished without any initialization. It takes about 4 to 6 minutes on a P-4 2.53 GHz PC for running each experiment with image size around 240×320 pixels. Overall, we have established an efficient method to detect shapes with deformation and articulation. Our experimental results are satisfactory but can still be improved. Encoding more global information into the representation and formulating more effective energy function are the two main directions for our future work.

References

[1] Y. Amit and A. Kong. Graphical templates for model registration. *PAMI*, 18(3):225–236, 1996.

[2] G. Borgefors. Distance transformations in digital images. *CVGIP*, 34(3):344–371, 1986.

[3] J. Canny. A computational approach to edge detection. *PAMI*, 8(6):679–698, 1986.

[4] T. F. Cootes, C. J. Taylor, D. H. Cooper, and J. Grahan. Active shape models: Their training and application. *CVIU*, 61(1):38–59, 1995.

[5] P. F. Felzenszwalb. Representation and detection of deformable shapes. *CVPR*, 2003.

[6] M. A. Fischler and R. A. Elschlager. The representation and matching of pictorial structures. *IEEE TC*, 22(1):67–92, 1973.

[7] D. M. Gavrila and V. Philomin. Real-time object detection for smart vehicles. In *7th ICCV*, pages 87–93, 1999.

[8] M. C. Golumbic. *Algorithmic Graph Theory and Perfect Graphs*. Academic Press, New York, 1980.

[9] U. Grenander. *Elements of Pattern Theory*. Johns Hopkins University Press, 1996.

[10] A. K. Jain, Y. Zhong, and S. Lakshmanan. Object matching using deformable templates. *PAMI*, 18(3):267–278, 1996.

[11] M. Kass, A. Witkin, and D. Terzopoulos. Snakes: Active contour models. In *1st ICCV*, pages 259–268, 1987.

[12] L. Prasad. Morphological analysis of shapes. *CNLS Newsletter 139*, July 1997.

[13] D. Thompson. *On Growth and Form*. Cambridge Univ. Press, 1917.

[14] L. Zhao. *Dressed Human Modeling, Detection, and Parts Localization*. PhD thesis, Robotic Institute, Carnegie Mellon University, Pittsburgh, PA, 2001.