

---

# Object Detection at the Optimal Scale with Hidden State Shape Models

---

**Jingbin Wang**   **Vassilis Athitsos**   **Stan Sclaroff**   **Margrit Betke**  
Computer Science Department, Boston University, Boston, MA 02215  
{jingbinw, athitsos, sclaroff, betke}@cs.bu.edu

## Abstract

Hidden State Shape Models (HSSMs) [2], a variant of Hidden Markov Models (HMMs) [9], were proposed to detect shape classes of variable structure in cluttered images. In this paper, we formulate a probabilistic framework for HSSMs which provides two major improvements in comparison to the previous method [2]. First, while the method in [2] required the scale of the object to be passed as an input, the method proposed here estimates the scale of the object automatically. This is achieved by introducing a new term for the observation probability that is based on a object-clutter feature model. Second, a segmental HMM [6, 8] is applied to model the “duration probability” of each HMM state, which is learned from the shape statistics in a training set and helps obtain meaningful registration results. Using a segmental HMM provides a principled way to model dependencies between the scales of different parts of the object. In object localization experiments on a dataset of real hand images, the proposed method significantly outperforms the method of [2], reducing the incorrect localization rate from 40% to 15%. The improvement in accuracy becomes more significant if we consider that the method proposed here is scale-independent, whereas the method of [2] takes as input the scale of the object we want to localize.

## 1 Introduction

One of the core problems of computer vision is localizing and recognizing objects or shapes in images with clutter. An important class of shapes for which the majority of existing localization/recognition methods cannot be applied are classes that exhibit variable structure. As defined in [2], shape classes of “variable structure” are classes in which some shape parts can be repeated an arbitrary number of times, some parts can be optional, and some parts can have several alternative appearances. As shown in Fig. 1, examples of shapes with variable structure are branches of leaves, where the number of leaves can vary, and images of hands, where each finger can be fully extended, partially extended, or hidden.

Hidden State Shape Models (HSSMs) are introduced in [2], as a generalization of Hidden Markov Models (HMMs) [9] that can be used for modeling shape classes of variable structure and for efficiently detecting such shapes in heavily cluttered images. Using HSSMs, object detection and recognition is achieved by finding a globally optimal registration between model states and image features. This globally optimal registration is found using dynamic programming (DP), and thus the complexity of the registration algorithm is polynomial to the total number of model states and the total number of image features.

Our goal in this paper is to address two important limitations of the original HSSM method presented in [2]. Those limitations are illustrated in Fig. 2. The first limitation of the original HSSM method was that the length of the registration (i.e., the number of image features to be matched to the model) was assumed to be known and had to be specified by the user. That limitation makes the method

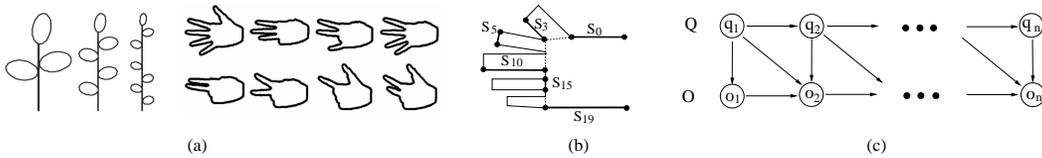


Figure 1: Introduction of Hidden State Shape Models (HSSM). (a): Two object classes that exhibit variable shape structure: branches with leaves and hand contours, by courtesy [2]. (b): Model states and shape components for hand examples in HSSMs. (c): Bayesian network for modeling the object detection problems in HSSMs, where layer  $Q = [q_1, \dots, q_n]$  refers to a sequence of HMM states, and layer  $O = [o_1, \dots, o_n]$  refers to an observation sequence.

of [2] relatively impractical for the task of detecting objects in unconstrained cluttered images. We refer to the problem of finding an optimal registration without knowing the registration length a priori as the *unknown-scale problem*.

The reason the method of [2] cannot address the unknown-scale problem is that that method, similar to HMMs, used a cost function inherently biased towards short registrations, as illustrated in Fig. 2. Under that cost function, adding an additional feature to a registration can never decrease the registration cost (or increase the registration likelihood). In this paper we address that limitation using a novel cost function, which is based on modeling probabilistically both the object appearance and the appearance of clutter. Incorrectly assigning either a “clutter” feature to the object, or an object feature to “clutter” increases registration cost. This way, our method is not biased towards registrations that are short, or registrations of any other particular length.

The second limitation of the original HSSM method that we address in this paper is illustrated in Fig. 2. In that figure, we show examples of incorrect registration results, where the combination of scales assigned to shape parts (like the finger lengths estimated in Fig. 2) are extremely unlikely. We use the term “scale-dependency problem” for the problem of identifying hypotheses where the combination of scales assigned to shape parts is implausible. In the method proposed in this paper, we capture the dependencies between scales of different object parts using the segmental HMM formulation by Gales and Young [6]. Segmental HMMs can model how long the registration process may remain in each HMM state (“duration probability”).

Sections 3, 4 and 5 formally define the proposed variable-structure shape models, and describe how such models can be learned and inferred, with focus on our example application, which is hand detection in cluttered images. Experiments in Section 6 demonstrate that, by making the proposed improvements over the original HSSM formulation of [2], the proposed method can handle more general real-world scenarios (i.e., not knowing the object’s size in the image) and achieves significantly more accurate object localization and recognition.

## 2 Related Work

A large amount of literature in computer vision addresses the issue of detecting deformable shapes in images. Examples include active contours [7], active shape models [3], graphical models [4, 10], and dynamic programming [1, 5]. The main difference between the method we introduce in this paper and all above-mentioned methods is that our method can be used for modeling and detection of shape classes that exhibit *variable* structure. We should stress that “structure variation” is not synonymous with “deformation.” Deformable model methods can model deformations of individual shape parts and deformations in the spatial arrangements between shape parts; unlike our method, they cannot capture structure variations, like the possibility that a shape part may be repeated an arbitrary number of times.

The only existing method for detecting shapes of variable structure is the HSSM method described in [2]. The method proposed in this paper builds on top of that method. As mentioned in the introduction, our method addresses two problems that the original HSSM method cannot address: the unknown-scale problem and the scale-dependency problem.

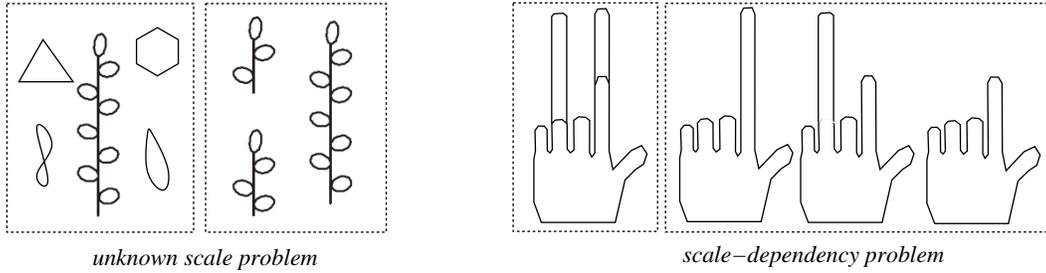


Figure 2: An illustration of the unknown-scale problem (left), and the scale-dependency problem (right), from which the original HSSM method of [2] suffers, and which we address in this paper. Left: an image including a branch of leaves, and some “clutter” objects, and an image showing different possible outputs of branch detection. The method of [2] cannot evaluate which of those outputs is better. Right: a hand image, including some “clutter” edges, and three possible registrations. The method of [2] cannot capture the fact that the combinations of finger lengths produced by the first two registrations are implausible.

### 3 Learning an HSSM: A Probabilistic Framework

An HSSM is specified by the following elements [2]:

- $\mathbb{S} = \{s_1, \dots, s_k\}$ : a set of state labels that include  $k$  foreground labels  $\{s_1, \dots, s_k\}$ . The foreground labels are associated with different object components during the traversal of the object contour.
- $\mathbb{E}$ : a subset of  $\mathbb{S}$  that defines legal end states of the HSSM. A registration process must terminate in a legal end state, in order to localize an object with possible shape components in an image.
- $\pi(s_i)$ : the initial probability that state  $s_i$  is the initial state.
- $A(s_i, s_j)$ : the state transition function that represents the transition probability from state  $s_i$  to state  $s_j$ . In particular, we define  $A(s_i, s_i) = 0$  to prohibit self-transitions.
- $B(f_p, s_i)$ : the state observation function that represents the probability of observing feature  $f_p$  in state  $s_i$ .
- $T(f_p, f_q, s_i, s_j)$ : the feature transition function that represents the probability of observing some feature  $f_q$  in state  $s_j$  and some other feature  $f_p$  in state  $s_i$ .
- $D(d_i, s_i, \omega)$ : the state duration function that represents the probability of continuously observing  $d_i$  features in state  $s_i$  under the given scale  $\omega$ . This function  $D$  comes from the segmental HMM framework [6, 8] and is a difference between the method proposed here and the original HSSM method of [2].

The key problem to be solved in this paper is find the most likely registration of an object model with features extracted from an image. Fig. 3 shows a Bayesian network that models the current problem. The first layer  $Q = [q_1, \dots, q_m]$  refers to a sequence of model states, where  $q_i \in \mathbb{S} \cup \{c\}$ , and  $c$  is a label associated with clutter. Note that observations due to clutter may appear in the background as well as in the foreground. The second layer  $O$  refers to a sequence of observed image features, described by observation sequence  $O = [o_1, \dots, o_m]$ , where  $o_i \in \mathbb{F}$ , and  $\mathbb{F} = \{f_1, \dots, f_m\}$  is a set of unordered image features that are extracted from an input image  $I$ .

The graphical model in Fig. 3 has two important differences from the original HSSM method of [2]. First, we explicitly model the observations in an image as being caused either by the object or by clutter. The original HSSM method does not include a clutter model. In our method, the detection process explicitly assigns each feature either to an object state or to clutter. Incorrectly assigning an object feature to clutter or a clutter feature to an object state would be penalized. Second, we adopt the segmental HMM formulation [6, 8] to model the “state duration behavior.” In a segmental HMM, a single shape state is allowed to represent a sequence of (similar) observations. By this

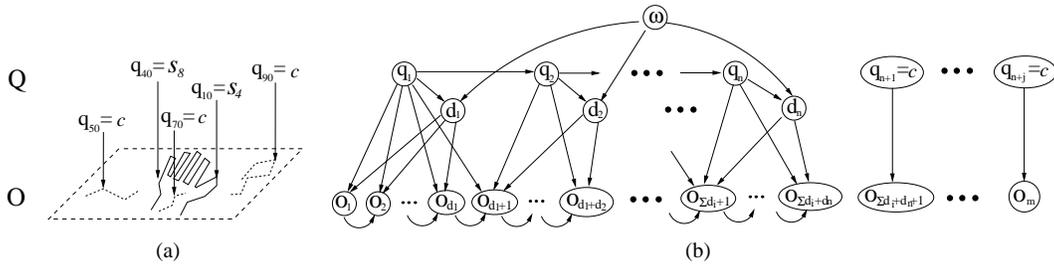


Figure 3: Bayesian network for the object localization and recognition by HSSMs. (a): an illustration of image features being assigned to object states or to the “clutter” state. (b): A graphical model of the detection problem we address in this paper.

way, the spatial constraints between the shape parts of the object to be located can be learned from a training set and applied for guiding the registration process.

Since each state variable is equal to either an object state  $s_i$  or to the clutter label  $c$ , we can partition the state sequence as subsequence  $Q_o = \{q_i | q_i \in \mathbb{S}\}$  that is matched with the object and subsequence  $Q_c = \{q_i | q_i = c\}$  that is matched with clutter, i.e.,  $Q = Q_o \cup Q_c$ . Accordingly, the observation sequence can be partitioned as  $O = O_o \cup O_c$  based on the associated state labels (note that we obtain the ordered sequence  $O$  from the unordered set  $\mathbb{F}$  of image features only after we find a registration between image features and model states). As a result of the inference algorithm, the image features in set  $\mathbb{F}$  belong to either the set of object observations  $\Phi_o$  or the set of observations that are due to clutter  $\Phi_c$ , and thus  $\mathbb{F} = \Phi_o \cup \Phi_c$ . The registration of the object provides an ordered sequence of object components. Each object component is composed of  $d_i$  object features that are modeled by a state  $q_i$ .

Given the graphical model defined in Fig. 3 and the model parameters summarized by  $\Omega = (\mathbb{S}, A, B, T, D, \pi)$ , the goal of the proposed method is to maximize the conditional joint probability:

$$\begin{aligned}
p(Q, O; \mathbb{F}, \Omega) &= p(Q_o, O_o; \Phi_o, \Omega) p(Q_c, O_c; \Phi_c, \Omega) \\
&= p(Q_o, O_o; \Phi_o, \Omega) \left[ \prod_{o_i \in \Phi_c} p(q_i = c, o_i; B) \right] \\
&= p(Q_o, O_o; \Phi_o, \Omega) \left[ \frac{\prod_{o_i \in \mathbb{F}} p(q_i = c, o_i; B)}{\prod_{o_i \in \Phi_o} p(q_i = c, o_i; B)} \right] \\
&\propto \frac{p(Q_o, O_o; \Phi_o, \Omega)}{\prod_{o_i \in \Phi_o} p(q_i = c, o_i; B)}
\end{aligned}$$

where  $\prod_{o_i \in \mathbb{F}} p(q_i = c, o_i; B)$  is a constant and thus can be omitted. In the above derivation, we considered each clutter feature to be conditionally independent. By expanding the foreground conditional probability  $p(Q_o, O_o; \Phi_o, \Omega)$  we obtain that  $p(Q, O; \mathbb{F}, \Omega)$  is proportional to the following quantity:

$$\begin{aligned}
& p(q_1; \pi) \prod_{i=2}^n p(q_i | q_{i-1}; A) p(o_{\ell(i)+1} | o_{\ell(i)}, q_i, q_{i-1}; T) p(d_i | q_i, \omega; D) \\
& \prod_{j=\ell(i)+1}^{\ell(i)+d_i} \frac{p(o_j | q_i; B)}{p(q = c, o_j; B)} p(o_{j+1} | o_j, q_i, d_i; T), \quad \text{for } q_i \in \mathbb{S}, o_j \in \Phi_o
\end{aligned} \tag{1}$$

where function  $\ell(i) = \sum_{k=1}^{i-1} d_k$  represents the total length of observation sequence before the  $i$ -th state.

The two important differences of this formulation compared to the formulation in [2] are:

- We maximize the likelihood ratio  $\frac{p(o_j | q_i; B)}{p(q = c, o_j; B)}$ , as opposed to maximizing  $p(o_j | q_i; B)$ . Maximizing the likelihood ratio does not suffer from bias towards short registrations, and as a by-product of the maximization process we obtain automatically the optimal registration length.

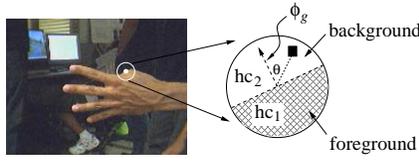


Figure 4: Image feature patch

- We introduce the “state duration probability”  $p(d_i|q_i, \omega; D)$  to capture the fact that the scales of individual object parts depend on each other.

The model inference procedure needs to assign each unknown  $q_i$ ,  $d_i$  and  $o_j$  in Equation 1 with an optimal value so that the joint probability of Equation 1 can be maximized. In a manner similar to [2], this maximization can be performed using dynamic programming.

The subsequent sections below will give the details about how we define and learn the probability ratio  $\frac{p(o_j|q_i; B)}{p(q=c, o_j; B)}$ , and other model parameters.

## 4 Model Learning

### 4.1 Object-Clutter Observation Model

For the current application, we define an image feature  $f \in \mathbb{F}$  to be a local image patch surrounding an edge pixel, as shown in Fig. 4. Each observation variable  $o_j$  is a random variable that can be assigned to some image feature  $f$ . Typically, an image feature is measured by its appearance  $\phi$  and location  $\ell$ , i.e.,  $f = (\phi, \ell)$ . Furthermore, the patch appearance  $\phi$  is summarized by its color  $\phi_\chi$  and its intensity gradient  $\phi_g$ . Accordingly, each shape state  $q = s \in \mathbb{S}$  models the color distribution of a foreground boundary patch and the image gradient  $s_g$  on the center of the patch, which is represented by the observation probability:

$$p(o = f|q = s; B) = p((\phi_\chi, \phi_g)|q = s; B) = p(\phi_\chi|q = s; B)p(\phi_g|s_g; B),$$

where  $p(\phi_g|s_g; B)$  is modeled by a Gaussian distribution with mean  $s_g$  and covariance  $\sigma_g$ . In practice, finding a good model of the likelihood function  $p(\phi_\chi|q = s; B)$  is difficult because the dimensionality of the feature patch representation is high and the training data limited. Instead, we rewrite the likelihood ratio in Eq. 1 as:

$$\frac{p(o = f|q = s; B)}{p(q = c, o = f; B)} = \frac{p(\phi_\chi|q = s; B)p(\phi_g|s_g; B)}{p(\phi_\chi|q = c; B)p(\phi_g|q = c; B)p(q = c)} \propto \frac{p(q = s|\phi_\chi; B)p(\phi_g|s_g; B)}{p(q = c|\phi_\chi; B)}, \quad (2)$$

where we consider  $p(\phi_g|q = c; B)p(q = c)$  and  $p(q = s)$  are constants, then we can approximate the posterior probability  $p(q = s|\phi_\chi; B)$  as

$$p(q = s|\phi_\chi; B) \approx \frac{1}{1 + \exp(-\gamma h(\phi_\chi))} \quad (3)$$

where function  $h$  is the decision value computed by a linear Support Vector Machine (SVM) classifier, and  $\gamma$  is a scalar factor. We learn a single two-class classifier for all  $s \in \mathbb{S}$  to approximate the likelihood ratio. Namely, given an input image patch, the classifier output indicates whether the input feature is a patch that belongs to the hand boundary or to clutter, i.e.,  $q = s$  or  $q = c$ . Naturally, an alternative is to learn a separate classifier for each object state, i.e., a classifier that discriminates between patches belonging to that state and patches belonging to clutter.

The SVM was learned from a training set of 40 hand images, where the correct hand boundary edges were localized by a HSSM or marked by a human. The resulting two sets of labelled edge pixels respectively included about 15,000 hand boundary edges and about 48,000 edges that were due to clutter. A local image patch was created for each edge pixel, as described above. To compute the training feature  $\phi_\chi$  of a patch, a weighted average of color values was determined by computing  $\cos \theta$ , where  $\theta$  is the angular difference with respect to the patch center and the local intensity gradient direction, as indicated in Fig. 4.

## 4.2 Feature Transition Model

We consider the feature locations to be dependent between continuous observations given the corresponding shape states, and we model feature transition by an exponential distribution:

$$p(l_{j+1}|l_j, q_i, d_i; T) \approx \lambda \exp(-\lambda(|l_{j+1} - l_j| - 1)), \quad (4)$$

where  $|l_{j+1} - l_j|$  represents the Euclidean distance between the centers of two patches  $o_{j+1}, o_j$ . Given the same image training set used in Section 4.1, we compute  $\lambda = \frac{1}{n} \sum_k (|l_{j+1} - l_j|_k - 1) \approx 0.4$ , where  $n$  is the total number of neighboring feature pairs in the training set.

## 4.3 State Duration Model

For our application, each shape state models the appearance of a certain part of the hand boundary (Fig. 1). In order to model the state duration in this HSSM, we need to learn the length distribution for each of shape states from a training set. The variable shape structure in the current application results from the action of bending or extending the finger. In the data collecting process, two human subjects were asked to bend or extent each of their fingers in a combinatorial way, which resulted in 16 different poses<sup>1</sup>. At each pose, the user was asked to vary the hand pose slightly to produce enough intra-shape variations. We took a total of 1,600 images of two left hands against a clean background. For each hand pose, 20 images were randomly chosen and added into a hand shape training set, which included 320 ( $16 \times 20$ ) images in total. Afterwards, we registered the HSSM with the hand image to produce the labelled boundary pixel sequences. Notice that the length of each finger part can be measured by the number of pixels that are associated with the same state. We normalized the state length with respect to the length of a fixed state, e.g., the length of the thumb in each of these examples. The resulting relative length statistics were used to approximate the duration distribution of each shape state, for which we applied a Gaussian mixture model:

$$p(d|q, \omega; D) = \sum_i^n p(d|\alpha_i(\omega), q; D)p(\alpha_i(\omega)|q, \omega; D), \quad (5)$$

where  $\omega$  is a scale parameter specified by the user,  $p(d|\alpha_i(\omega), q)$  is a normal distribution with mean  $\alpha_i$  and covariance  $\sigma_i$ , and  $p(\alpha_i(\omega)|q, \omega; D)$  is the conditional prior for Gaussian distribution. We chose  $n = 2$  to model the state duration that represents the finger length, and  $n = 1$  to model the state duration that is associated with the length of the finger tip. We applied an expectation-maximization procedure to find the optimal estimates for parameters  $\alpha_i$  and  $\sigma_i$  with respect to all states  $s \in \mathbb{S}$ .

## 5 Model Inference by Dynamic Programming

In our hand detection application, we assigned  $p(q_1; \pi)$  as a constant and defined the transition probability function  $p(q_i|q_{i-1}; A)$  as a uniform distribution with respect to all legal transitions given the current state. We also assigned the transition probabilities to be zero for illegal state transitions.

Given all probability terms defined above and their learned parameters, the inference problem described in Eq. 1 can be solved by a dynamic programming method similar to the Viterbi algorithm. In the general case, the complexity of the inference algorithm is  $O(m^2n^2k)$ , where  $m$  is the total number of model states,  $n$  is the total number of image edge features, and  $k$  is the maximum registration length we could have, e.g.,  $k = 600$  in the current implementation. In our implementation, the above complexity is reduced to  $O(mnk)$  by imposing the following restrictions: first, we allow no more than a certain number of legal state transitions for every state. Second, when  $T(f_p, f_q, s_i, s_j)$  is less than a threshold, we do not allow a simultaneous transition from  $s_i$  to  $s_j$  and from  $f_p$  to  $f_q$ .

## 6 Experiments

We implemented two versions of the proposed method and compared them with the method of [2] and also with the chamfer distance (taking edge orientations into account, as in [11]), which has

<sup>1</sup>In this experiment, we required the subjects to keep their thumb relatively still and only bend or extend the other four fingers

been used in the literature for a similar hand detection task [11]. One version of our algorithm only included the object-clutter modeling, while the other includes both the object-clutter modeling and the duration model as a segmental HMM. The results are reported in Table 1; some representative images are given in Fig. 5. In the current experiments, we specified a fixed minimum and maximum registration length  $(L_{min}, L_{max}) = (100, 600)$  for all images. The DP-based inference algorithm identified an optimal solution based on the minimum cost stored in the DP table within this length range.

The implementation of the segmental-based HSSM is slightly different. For each state duration model, we chose four different scales to establish the Gaussian mixture distribution. With respect to this scale factor, we performed the DP process four times for  $(L_{min}, L_{max}) \in \{(75, 150), (150, 300), (225, 450), (300, 600)\}$ , and the optimal solution was found as the minimum cost among all costs computed by four DP computational procedures.

method:	Chamfer distance	HSSM	HSSM + SVM-NC	HSSM +SVM-NC+SEG-HMM
Number of orientations:	72	8	8	8
Correct recognition	21.8%	33.7%	58.9%	60.9%
Correct localization	54.6%	59.5%	84.7%	83.3%
Incorrect localization	45.4%	40.5%	15.3%	16.7%
Computational time	15 s	5–6 min	2–3 min	5–8 min

Table 1: Comparison results on 353 hand images. “SVM-NC” stands for the “SVM-based negative cost.” “SEG-HMM” stands for “segmental HMM.”

In Table 1 “correct recognition” refers to the case where the system has found the shape at the correct location and orientation, and has correctly registered each shape part. “Correct localization” refers to the case where the system has identified the correct object location and orientation. In particular, we require that 75% of the palm edges be registered correctly; and “incorrect localization” refers to the case where the method failed to find the correct object location and orientation. Also note that “correct recognition” is a subcase of “correct localization.”

Fig. 5 shows the comparison results between the proposed methods and the original HSSM method of [2] without knowing the registration length as a priori. We note that the object-clutter modeling leads to a significant improvement in both the correct recognition rate and the correct localization rate, compared to the original HSSM method [2]. We should emphasize that results reported for the algorithm of [2] were obtained by passing as input to that algorithm, for every image, the desired registration length for that image. Therefore, compared to the algorithm of [2], our algorithm had to perform a harder task, since our algorithm also had to estimate the optimal registration length. Therefore, compared to the method in [2], the proposed method is both more general, since it addresses the unknown-scale problem, and significantly more accurate.

## 7 Conclusion

We have presented a method for modeling shape classes of variable structure and detecting instances of such classes in heavily cluttered images. Compared to the only other existing method for this problem [2], the proposed method is both more general and more accurate. It is more general because it can detect objects whose scale is not known a priori. It is more accurate, because it includes a model of clutter in addition to modeling object appearance, and because it captures dependencies between the scales of different object parts. In experiments on hand detection in cluttered images, the proposed method is significantly more accurate than other methods previously applied.

## References

- [1] A. A. Amini, T. E. Weymouth, and R. C. Jain. Using dynamic programming for solving variational problems in vision. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12(9):855–867, 1990.
- [2] V. Athitsos, J. Wang, S. Sclaroff, and M. Betke. Detecting instances of shape classes that exhibit variable structure. In *9th European Conference on Computer Vision*, Graz, Austria, 2006.

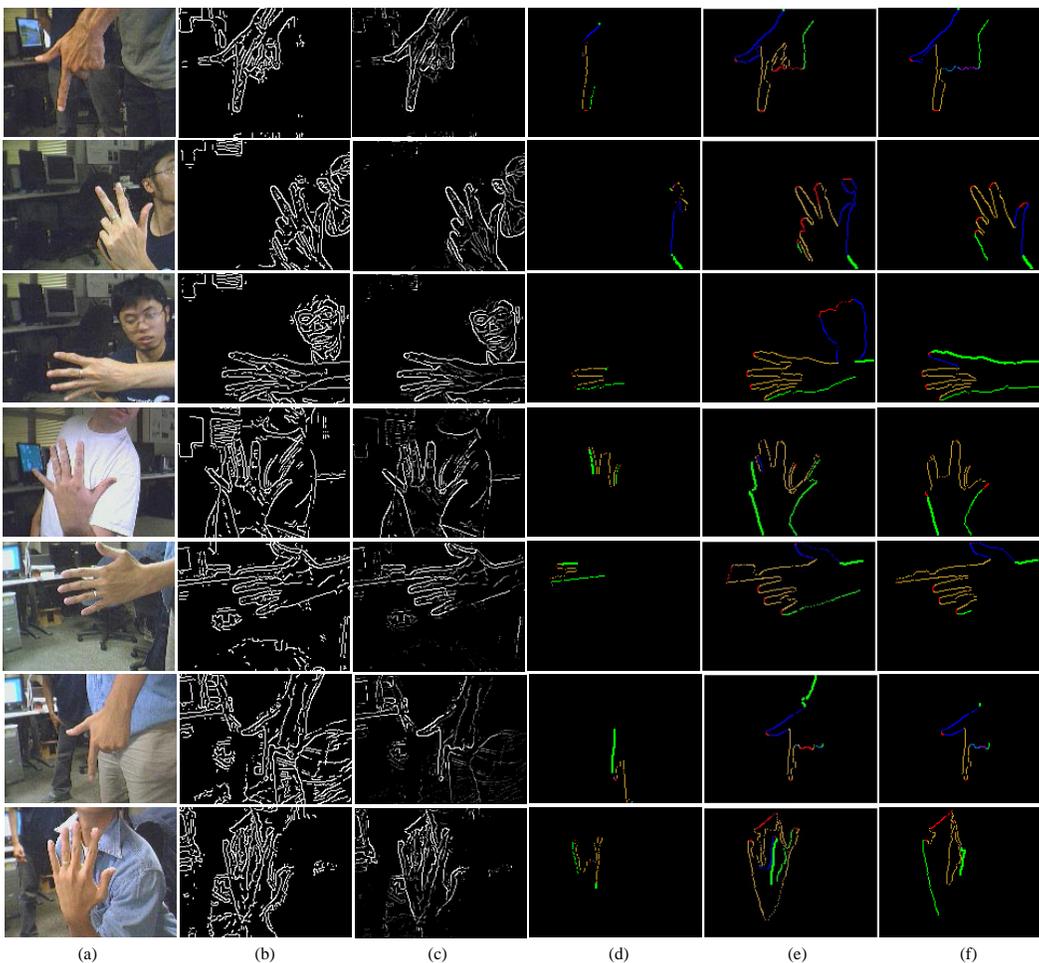


Figure 5: Example images of our results. (a): input images; (b): edge image after pruning; (c): SVM ratio posterior map; (d): labelling results computed by normalizing the registration costs with respect to the length in HSSM [2]; (e):labelled results computed by HSSM+SVM-NC; (f): labelling result computed by HSSM+SVM-NC+SEG-HMM. For column (f), rows 1-3 represent correct recognition, row 4 illustrates correct detection, and rows 5-7 represent incorrect detection.

- [3] T. F. Cootes, C. J. Taylor, D. H. Cooper, and J. Graham. Active shape models - their training and application. *Computer Vision and Image Understanding*, 61(1):38–59, 1995.
- [4] J. M. Coughlan and S. J. Ferreira. Finding deformable shapes using loopy belief propagation. In *Proceedings of the 7th European Conference on Computer Vision*, volume 3, pages 453–468, 2002.
- [5] P. F. Felzenszwalb and D. P. Huttenlocher. Pictorial structures for object recognition. *International Journal on Computer Vision*, 61(1):55–79, 2005.
- [6] M. J. F. Gales and S. J. Young. The theory of segmental Hidden Markov Models. Technical Report CUED/F-INFENG/TR.133, Cambridge Univ. Eng. Dept., 1993.
- [7] M. Kass, A. Witkin, and D. Terzopoulos. Snakes: Active contour models. *IJCV*, 1(4):321–331, 1988.
- [8] M. Ostendorf, V. Digalakis, and O. Kimball. From HMMs to segment models: a unified view of stochastic modeling for speech recognition. *IEEE Transactions on Speech and Audio Processing*, 4(5):360–378, 1996.
- [9] L. R. Rabiner. A tutorial on Hidden Markov Models and selected applications in speech recognition. In *Proceedings of the IEEE*, volume 77:2, 1989.
- [10] L. Sigal, M. Isard, B. H. Sigelman, and M. J. Black. Attractive people: Assembling loose-limbed models using non-parametric belief propagation. In *NIPS*, 2003.
- [11] A. Thayananthan, B. Stenger, P. H. S. Torr, and R. Cipolla. Shape context and chamfer matching in cluttered scenes. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pages 127–133, 2003.