

# Parametrized Shape Models for Clothing

Stephen Miller, Mario Fritz, Trevor Darrell, Pieter Abbeel

**Abstract**—We consider the problem of recognizing the configuration of clothing articles when crudely spread out on a flat surface, prior to and during folding. At the core of our approach are parametrized shape models for clothing articles. Each clothing category has its own shape model, and the variety in shapes for a given category is achieved through variation of the parameters. We present an efficient algorithm to find the parameters that provide the best fit when given an image of a clothing article. The models are such that, once the parameters have been fit, they provide a basic parse of the clothing article, allowing it to be followed by autonomous folding from category level specifications of fold sequences. Our approach is also able to recover the configuration of a clothing article when folds are being introduced—an important feature towards closing the perception-action loop. Additionally, our approach provides a reliable method of shape-based classification, simply by examining which model yields the best fit. Our experiments illustrate the effectiveness of our approach on a large set of clothing articles. Furthermore, we present an end-to-end system, which starts from an unknown spread-out clothing article, performs a parametrized model fit, then follows a category-level (rather than article specific) set of folding instructions, closing the loop with perceptual feedback by re-fitting between folds.

## I. INTRODUCTION

Robotic manipulation abilities are still rather limited in comparison to human manipulation abilities. This is largely due to the limited perceptual understanding current algorithms are able to provide. The problem of perception is particularly challenging when considering deformable objects, as they have very high dimensional configuration spaces.

The perceptual challenges involved in the manipulation of deformable objects are greatly reflected in the state of the art in robotic automation of laundry folding. Indeed, the current state of the art is far from enabling general purpose manipulators to fully automate such a task.

One of the critical challenges in automating laundry folding is the ability to recognize the configuration of a crudely spread out, previously unseen clothing article. For example, when faced with a new t-shirt, one would like for a robot to be able to “understand” the t-shirt’s configuration to the extent of recognizing where the sleeves are, where the main body is, how long the sleeves are, where the collar is, and so forth. If the robot is able to acquire such a parametrized shape representation, then it can proceed to, for example, fold in the sleeves, or fold along “vertical” lines running next to the collar. Moreover one would like the robot to be able to continue to track the configuration after folds have been executed, thus enabling it to close the perception-action loop.

Stephen Miller, Mario Fritz, Trevor Darrell, and Pieter Abbeel are with the University of California at Berkeley. E-mail: {sdmiller,mfritz,trevor,pabbeel}@eecs.berkeley.edu.

In this paper, we propose parametrized shape models for clothing categories and algorithms that enable automatic fitting of those parameters to new instances. Concretely, we consider clothing articles that either (i) are crudely spread out on a flat surface, as might result from an initial handling procedure which tries to maximally spread a clothing article; or (ii) were crudely spread out initially, and have since undergone folding manipulations.

We describe an efficient algorithm to fit the parameters of our models to a new clothing instance. Our algorithm starts by optimizing over coarse parameters, such as the position, orientation, and scale of the article, and then gradually introduces more detailed parameters into the model, such as sleeve length, width, and angle. Our experiments show our approach to be highly reliable in both determining the category of an instance, and fitting the proper model to that instance.

We further illustrate the effectiveness of our algorithm through implementation of an end-to-end folding system on the Willow Garage PR2 robot. For each clothing category we specify a parametrized folding sequence—which is defined relative to the parametrized shape model. We present the robot with fairly spread out clothing articles. The robot fits the shape model parameters, and then executes the folds. In between folds, the robot re-fits the shape model, which then includes additional parameters to capture the fold lines—providing the robot feedback about the accuracy of the folds and where to grasp next.

## II. RELATED WORK

The estimation of clothing configuration is a relatively new problem, and has received scant attention in the literature. Clothing forms follow the structure of the human body, and thus are an instance of an articulated pose estimation task. Classic articulated pose estimation methods iteratively fit or track an articulated model, updating the pose of individual part segments subject to the overall body constraints. Early methods were based on optic flow and linearized exponential map-based constraints [4]; subsequent approaches developed efficient sampling methods [24], exemplar methods [6], regression strategies [25], and subspace optimization methods [23]. An early approach used an energy-optimization strategy to match edge points between two images [2]. Related models for fully non-rigid shape modeling and estimation are typically based on a learned manifold, e.g., active appearance models [5]. Few methods investigate clothing explicitly. Notable exceptions to this is the recent work of [8], which expands the SCAPE manifold-based model [1] to include a model of 3-D clothing forms, person tracking systems

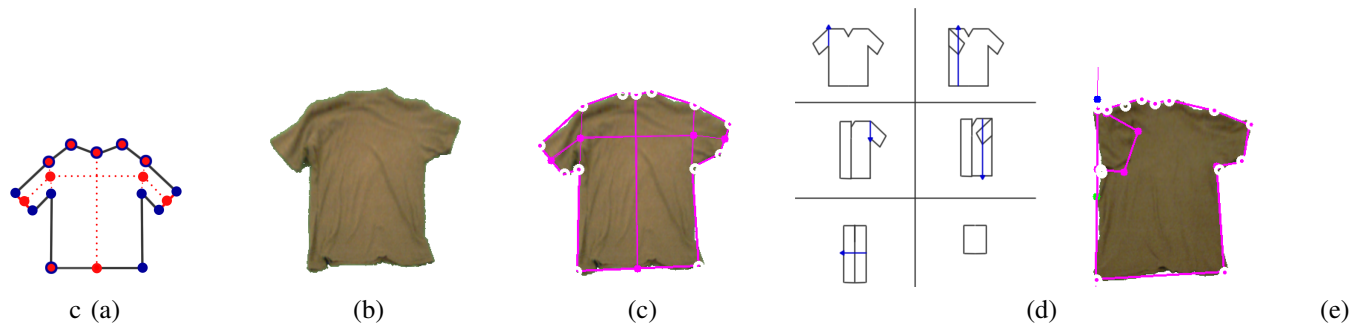


Fig. 1. (a) A parametrized shape model for a t-shirt. Red indicates a skeletal parameter, blue indicates a landmark point. (b) An example instance. (c) The result from running our approach: our model (pink) is overlayed onto the image of the clothing it is trying to fit. The determined landmark points are shown as white dots. (d) An example of a parametrized specification of a folding sequence. All fold-lines shown are defined relative to the parametrized shape model. (e) Result from running our approach when a fold line is present.

which attempt to account for clothing variation [18], and methods for estimating folds in deformable surfaces [22]. In this work we adopt a much simpler model, and propose schemes for direct optimization of specific shape forms that can be directly related to ensuing manipulation; we expect our method to become even more robust when the above statistical shape and observation model methods are applied to our approach, but the results with classic methods prove to be significant as shown below.

From the application angle, the prior work on robotic laundry manipulation and folding is most closely related to our work. The current state of the art is still far removed from having a generic robot perform end-to-end laundry, but progress has been made on various aspects.

Paraschidis and collaborators [7] describe the isolated executions of grasping a layed-out material, folding a layed-out material, laying out a piece of material that was already being held, and flattening wrinkles. Their perception system relies on a library of exact, polygonal shape models of the instances considered and then matches the sequence of extracted edges.

There is a body of work on recognizing categories of clothing. For example, Osawa and collaborators [17] and Hamajima and Kakikura [9] present approaches to spread out a piece of clothing using two robot arms and then classify its category.

Yamakazi and Inaba [28] present an algorithm that recognizes wrinkles in images, which in turn enables them to detect clothes laying around. Kobori and collaborators [11] have extended this work towards flattening and spreading clothing. Kita and collaborators [10] fit the geometry of the silhouette of a hanging piece of clothing to the geometry of a mass spring model of the same piece of clothing and are able to infer some 3-D information about the piece of clothing merely from the silhouette.

Our earlier work [12] focused on perception algorithms for detection of the vertices of towels. It enabled folding starting from a crumpled towel.

Some of the prior work describes robots using tools and the design of special purpose end-effectors as a step towards laundry folding. For example, Osawa and collaborators [16] developed a robot capable of using a “flip-fold” for folding

and a plate for straightening out wrinkles. Salleh and collaborators [21] present an inchworm gripper for tracing the edge of a piece of clothing. A range of gripper designs is presented by Monkman. [13].

In the more general context of folding, the work of Balkcom and Mason [27] stands out, having developed a robot that can perform origami.

### III. OVERVIEW

Our aim is to describe an arbitrarily complex article of clothing by a simpler, parametrized model. We consider parametrized models with which we can naturally associate a polygon. This polygon can then be used to perform intelligent manipulation of the clothing article.

The main components of our approach are:

- Parametrized shape models for clothing categories. For each category of clothing, we attempt to find the minimal set of parameters that can describe its range of shapes. Every (legal) setting of the parameters defines a polygonal description of the shape. We call the vertices of this polygonal description the *landmark points* of the shape. To further guide the optimization and preclude nonsensical results from consideration, we impose legality constraints on the parameters. Fig. 1 (a) shows a parametrized shape model for t-shirts as well as the polygon associated with this particular model instantiation.
- When presented with a previously unseen clothing article, we optimize over the parameters in the relevant category model to find the best fit according to an energy function which evaluates how well the contour  $C$  of the perceived clothing article matches the contour of polygons that can be generated by the parametrized shape model. Fig. 1 (b) shows an example image, with background and perspective effects automatically removed. Fig. 1 (c) shows the best fit of the t-shirt model to the image as obtained by our algorithm.
- We use the resulting fit polygon as input to a category-level parametrized folding sequence, by which the robot is able to systematically fold the article. Fig. 1 (d) shows an example of a parametrized folding specification for t-shirts. See previous work for details. [26]

- After completion of each fold, our approach augments the parametrized shape model to include the location of a fold line. It then fits this augmented set of parameters to provide feedback during the folding process. Fig. 1 (e) shows an example result.

#### IV. PARAMETRIZED SHAPE MODELS

We define a *model*  $M$  by the following components:

- A *landmark generator*

$$M_{LG} : \{P \in \mathbb{R}^p\} \rightarrow \{L \in \mathbb{R}^{2 \times \ell}\}$$

which takes a parameter vector  $P$  as input, and returns the associated collection of landmark points,  $L$ .

- A *contour generator*

$$M_{CG} : \{P \in \mathbb{R}^p\} \rightarrow \{C \in \mathbb{R}^{2 \times c}\}$$

which takes a set of scalar parameters  $P$  as input, and returns the contour of the polygon which would arise from the given parameters, with a fixed number of samples per side. The number of contour points  $c$  is chosen to be much larger than the number of landmark points  $\ell$ .

- A *legal input set*

$$M_{\mathcal{L}} \subseteq \mathbb{R}^p$$

which define set of parameters over which  $M$  is said to be in a legal configuration

- A *transformation operator*

$$M_T : \{P \in \mathbb{R}^p, T \in \mathbb{R}^2, \theta \in \mathbb{R}, s \in \mathbb{R}\} \rightarrow \{P' \in \mathbb{R}^p\}$$

which transforms a set of parameters in such a way that the resultant contour  $M_{CG}$  will be translated, rotated, and scaled by the given values of  $T$ ,  $\theta$ , and  $s$ .

##### A. Skeletal models

To capture the structure of the clothing we parametrize a model about a set of interior (or *skeletal*) points, as well as features which detail the distance from the interior points to the contour. These may include landmark vertices, displacements between a skeletal vertex and its nearest edge, or scalars such as height and width.

Fig. 1 (a) shows an example skeletal model for a t-shirt; a more detailed list of the parameters is shown in Fig. 4. The parameters are highlighted in red, and the landmark points are highlighted in blue. A red point with a blue outline indicates a landmark point which is itself a parameter. The generated contour is outlined in black. The legal input set is detailed in Section VIII-A.3.

##### B. Folded Models

Once the pose of the spread-out cloth has been determined, we wish to visually track the progress and accuracy of our folding procedure. With an initial model  $M_0$  and associated parameters  $P_0$ , we associate a folded model such that:

$$P_{folded} = [\Theta \mid P_0]$$

$$\mathcal{L}_{folded} \equiv \mathbb{R}^4 \times \mathcal{L}_0$$

where all parameters of the original model are allowed to vary, and in addition, the parameters  $\Theta$  specify a line about which the model is to be folded. The resulting landmark points are computed by folding the polygon specified by  $M_{0LG}(P_0)$  about this line.<sup>1</sup>

If we are certain the clothing article did not move during a folding operation, we may choose to optimize over the fold-line, leaving the unfolded portion of the model untouched. We therefore define a *Static Folded Model*, such that

$$P_{folded} = [\Theta]$$

$$\mathcal{L}_{folded} \equiv \mathbb{R}^4$$

#### V. ENERGY FUNCTION

We now aim to find the parameters which optimally fit a given image. Our approach extracts the contour of the clothing article in the image and uses an energy function which favors contour fit. We define the energy  $E$  as follows:

$$E(P) = (\alpha) \times \bar{d}(M_{CG}(P) \rightarrow C) + (1 - \alpha) \times \bar{d}(C \rightarrow M_{CG}(P))$$

where  $\bar{d}(A \rightarrow B)$  is the average nearest-neighbor distance<sup>2</sup> from  $A$  to  $B$ :

$$\bar{d}(A \rightarrow B) \equiv \frac{1}{|A|} \sum_{a \in A} \sum_{b \in B} \text{argmin} ||b - a||$$

The parameter  $\alpha$  is used to adjust the way in which the model fits to the contour. If  $\alpha$  is too low, the model will attempt to fit every point of the contour, often overfitting to deviations such as wrinkles. If  $\alpha$  is too high, the model may cease to cover the contour at all, fixating instead on a single portion. We've found that setting  $\alpha = 0.5$  is sufficient to counter both negative tendencies.

#### VI. ENERGY OPTIMIZATION

Our energy optimization follows a coarse-to-fine strategy, in which the parameter space begins small and increases as the procedure continues. It first only considers translation, rotation and scale, then considers all parameters but enforces certain symmetry constraints amongst them, and finally optimizes over all parameters without the symmetry constraints.

##### A. Initialization

1) *PCA Approach*: To infer the necessary translation, rotation, and scale, we rely on a Principal Component Analysis of the observed contour, and contour defined by the model.

We first compute the initial model contour as

$$M_c = M_{CG}(P_0).$$

<sup>1</sup>In our implementation, the folded model is always initialized with the proper fold direction – as the direction of a fold will never be accidentally flipped in any robotic task. While the direction of the fold is specified by  $\Theta$ , it is doubtful that a continuous optimization approach would compensate for such an error. For those interested in a general approach, we suggest multiple optimizations: one for each fold direction.

<sup>2</sup>We additionally considered the use of Dynamic Time Warping [15], [20] in our distance metric. The results, however, showed little improvement, so for the sake of simplicity and computational efficiency, we restrict our approach to nearest-neighbor.

We then calculate the centers of mass of the observed contour and the model contour;  $c_o$  and  $c_m$  respectively. We then compute the relative translation between the two contours,

$$T = c_o - c_m.$$

We then perform PCA to estimate the principal axes of each contour, denoted  $a_o$  and  $a_m$ . We compute the relative angle between the two axes

$$\theta = \arccos(a_o \cdot a_m).$$

Finally, for each contour we find the point of intersection between the top of the contour and its principal axis, denoted  $t_o$  and  $t_m$ . We compute the relative scale between the two contours as

$$s = \frac{\|t_o - c_o\|}{\|t_m - c_m\|},$$

which is approximately the ratio of the heights of the two contours. The resultant contour  $M_c(P)$  will be centered about  $c_o$ , and scaled and rotated such that  $t_o = t_m$ .<sup>3</sup>

Having computed these three values, we then update our model estimate such that

$$P' \leftarrow M_T(P, T, \theta, s).$$

2) *Multi-Angle Approach*: We additionally consider a second approach, in which the optimization is run with multiple initializations, attempting all possible rotations within a granularity of  $\delta\theta$ . Upon completion, the fitted model which yields the lowest energy function is chosen, and all others are discarded. The method for choosing translation and scale is the same as in the PCA approach.

## B. Optimization

To ensure the best possible fit, our standard approach performs the optimization in three phases: Orientation, Symmetric, and Asymmetric.

In the *Orientation Phase*, all parameters are held relatively fixed, with only one external degree of freedom:  $\theta$ , which defines the net rotation of the contour points about the center of gravity of the model. This phase is only run when using the PCA-based initialization, and it tends to improve the orientation estimate as it considers the entire contour, rather than just its principal component. When using the multi-angle initialization we found it better to skip the orientation phase as it reduced the variety of orientations explored.

In the *Symmetric Phase*, the model is free to translate, rotate, scale or deform within the limits determined by its legal input set—as long as left-right symmetry is maintained. In terms of implementation, this is done by optimizing over a subset of the model parameters – those which describe the left and center portions of the model – and computing the implied values for the remaining right parameters such that symmetry is enforced.

<sup>3</sup>Thus described, the PCA approach leaves an ambiguity in terms of which direction is assumed to be “up” on the principal axis. To resolve this, we attempt both upright and upside-down initializations, and choose the minimum-cost result after the optimization is complete.

In the *Asymmetric Phase*, all parameters are optimized over, and the model is free to translate, rotate, scale, or deform within the limits determined by its legal input set.

For the numerical optimization, we use coordinate-wise descent over the parameters—evaluating the gradients numerically (rather than analytically) and maintaining an adaptive step-size for each parameter.

To enforce legality constraints on the parameters, we augment the energy function with a penalty for constraint violation. We first normalize the fit such that

$$\forall P : 0 \leq E_{\text{norm}}(P) < 1$$

To do so, we set

$$E_{\text{norm}} = \frac{E}{E_{\text{max}}}$$

As a simple upper bound,  $E_{\text{max}}$  is set to  $\sqrt{h^2 + w^2}$ , where  $h$  and  $w$  denote the height and width of the image, respectively. This corresponds to the case in which the two contours are maximally distant given the size of the image.

We then define the structural penalty  $S$  as

$$S(P) = \begin{cases} 0 & \text{if } P \in M_{\mathcal{L}} \\ 1 & \text{else} \end{cases}$$

The resulting energy function is then given by:

$$\mathcal{E}(P) = E_{\text{norm}}(P) + S(P)$$

As the normalized energy  $E_{\text{norm}}$  lies between zero and one, the optimum of the cost function will never violate a constraint if a legal alternative exists.

## VII. CLASSIFICATION

For any image and specified model, the above procedure is able to return a set of fit parameters and an associated energy. By considering the value of the energy function as a measure of overall model fit, this provides a convenient means of category classification. When presented with an image and a set of possible categories, we run the above procedure multiple times, with one model associated with each category. The fitted model which results in the lowest final energy is selected, and the image is classified accordingly.

## VIII. EXPERIMENTAL RESULTS

### A. Parametrized Models

1) *Towels*: As there is little inherent structure to a towel, its Skeletal Model is simply parametrized about the location of its four vertices. Only one constraint was imposed, which is common to all of our models:

- The model contour cannot have any self-intersections.

See Fig. 2 for details.

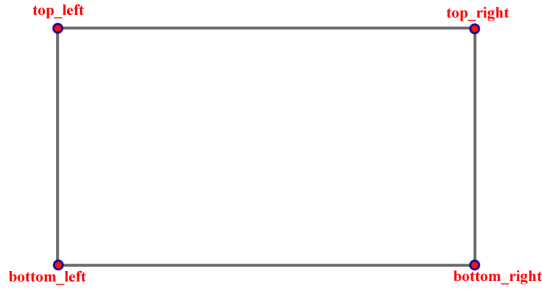


Fig. 2. A towel model has 8 total parameters, corresponding to 4 skeletal points. These are simply the four corners of the towel.

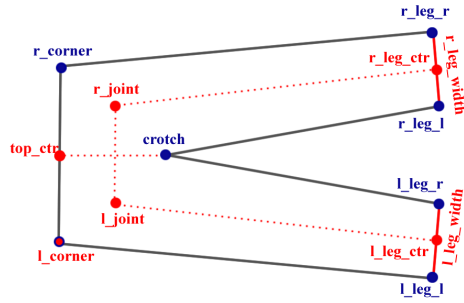


Fig. 3. The pants skeleton is defined by 14 scalar parameters, corresponding to 6 skeletal points and two scalar values, denoting the width of each pant leg. The remaining landmark points are generated as follows: the right corner is an extrapolation of the distance from the left corner to the top center; the croch is the top center mirrored about the axis spanning the left and right joints; the leg corners are determined by the line perpendicular to the leg axis, at a distance specified by the leg width.

2) *Pants*: A Skeletal Model of pants was devised, whose parameters are shown in Fig. 3.<sup>4</sup>

We found it was best to give the Pants Model as much freedom as possible. Therefore, only a small number of constraints were imposed, penalizing extreme deviations from the norm of:<sup>5</sup>

- The length of the legs relative to the height of the pants
- The width of the legs relative to the width of the pants
- The width of the pants relative to the height

For the fitting of pants two different initializations were attempted: the first with the legs virtually straight, and the second with the legs widely spaced. Both models were fit, and the one with the lowest final cost function was chosen.

3) *Short-sleeved shirts*: A Skeletal Model of short-sleeved shirts was also used, detailed in Fig. 4

In order to guide the optimization, a number of constraints were imposed, restricting:

<sup>4</sup>In all of these models, the preferred representation of parameters was in Cartesian coordinates. We additionally explored optimizing directly over angles and lengths. In practice, however, the optimization worked best when all parameters were aperiodic and similarly scaled. Hence, whenever possible, a length/angle combination was represented by a 2D point.

<sup>5</sup>For the precise numerical constraints of all of our models, see the attached code at [http://rll.berkeley.edu/ICRA\\_2011](http://rll.berkeley.edu/ICRA_2011)

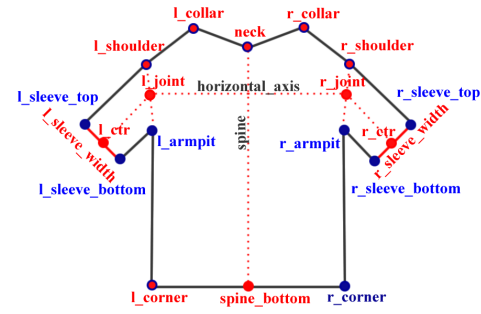


Fig. 4. A short-sleeved shirt skeleton is defined by 24 parameters, corresponding to 11 skeletal points and 2 scalar parameters for sleeve width. The remaining landmark points are generated as follows: the right corner is found by extrapolating the line from the left corner to the spine bottom; the armpit is determined by extrapolating the line from the shoulder to the shoulder joint; the sleeve corners are determined by the line perpendicular to the sleeve axis, at a distance specified by the sleeve width.

- The location of the collar points with respect to the neck and shoulders
- The location of the shoulders with respect to the armpits
- The angle between the spine and horizontal axis
- The relative size and angle of the sleeves
- The width-height ratios of the sleeves and torso

Two different initializations were attempted: the first with medium-length sleeves, and the second with extremely short sleeves. Both models were run, and the one with the lowest final cost function was chosen.

In addition to the Orientation, Symmetric, and Asymmetric phases of optimization, a fourth Fine Tuning phase was run. In this phase, the location of all sleeve vertices were free to move, while the rest remained fixed. This was meant to account for the irregular shape of many t-shirt sleeves.

4) *Long-sleeved shirts and sweaters*: The Skeletal Model for long-sleeved shirts is detailed in Fig. 5

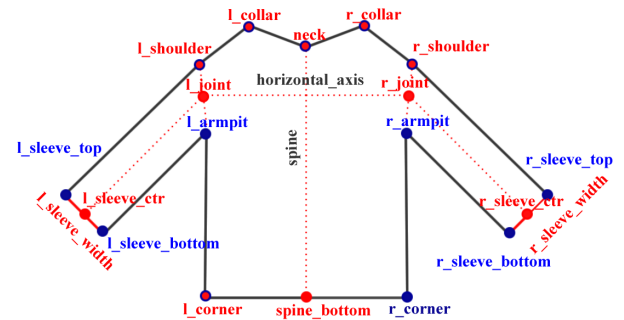


Fig. 5. A long-sleeved shirt skeleton is defined by same parameters as the short sleeved skeleton

This model is virtually identical to the Short-Sleeved model, with a single constraint added:

- Each sleeve must be at least twice as long as it is wide
- Only one initialization was used, with the arms at a downward angle.

As long-sleeved shirts have the potential for drastic asymmetry, both the Orientation and Symmetric phases of optimization proved to be unuseful, and occasionally damaging – the former settling on erroneous angles, and the latter on vastly incorrect poses. In some cases, the error was so great that the Asymmetric phase could not correct for it. For this reason, only the Asymmetric phase of optimization was used on this model.

5) *The Polygon Model*: To gauge the value of our skeletal approach, we introduce the *Polygon Model*, which is parametrized about

$$\begin{aligned} P_{poly} &= [l_1(x) \ l_1(y) \ \dots \ l_\ell(x) \ l_\ell(y)] \\ \mathcal{L}_{poly} &\equiv \mathbb{R}^{2\ell} \end{aligned}$$

This model has no interior structure, and no legality constraints beyond self-intersection. For every clothing category listed above, we construct a Polygonal Model whose initial landmark points are identical to those of the Skeletal Model for that category. This model provides a useful baseline for the performance of pure contour fitting, beginning with the same initialization and optimization techniques, but without taking any prior knowledge about clothing into consideration.

### B. Data Collection

To quantitatively gauge the accuracy of our approach, our shape-fitting code was run on a dataset of roughly 400 images, divided into four categories: towels, pants, short-sleeved shirts, and long-sleeved shirts. For each category, ten representative articles of clothing were considered. These varied greatly in size, proportion, and style. Each article was then further placed in ten or more poses, encompassing a variety of common spread-out configurations. (See Fig. 6.)

Each object was initially photographed on a green table. To ensure rotational invariance, each image was transformed to a birdseye perspective, using OpenCV’s checkerboard detector to locate the top-down frame. The background was then subtracted from each image. For most of these images, hue thresholding against the green background was sufficient: however, in cases where the complex texture of the clothing precluded hue thresholding, the Grabcut algorithm [19] was used to perform the subtraction, with foreground and background pixels manually selected by a user. Finally, the location of each landmark point was hand-annotated, to provide ground truth data for the model fitting task. The pipeline is illustrated in Fig. 7.

### C. Implementation Details

We ran our experiments on a Lenovo Thinkpad, running an Intel Core 2 Extreme Processor. A typical model fit took roughly 30 seconds; for more complex procedures such as the four-phase multi-model approach for T-shirts, convergence would occasionally take up to 2.5 minutes. To rapidly compute nearest neighbor distances for the cost function, the Flann library [14] was used. The bulk of the image processing, including transformations, thresholding, and contour detection, was done with OpenCV [3].



Fig. 6. The article of clothing is put in various poses.

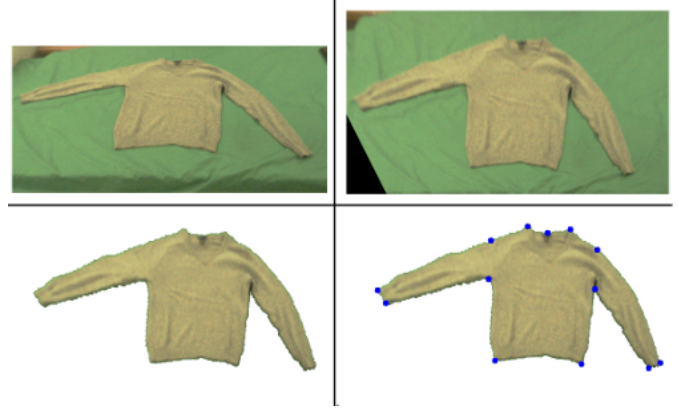


Fig. 7. The dataset pipeline. **Top Left**: Initially, the clothing is spread out on a green table. **Top Right**: A birdseye transformation is then performed. **Bottom Left**: The image is cropped, and the background is segmented out. **Bottom Right**: To provide ground truth for the fitting procedure, the resulting image is hand-annotated.

### D. Perception Results

Each image was first fit to the proper model according to its known category. The Table in Fig. 8 shows the accuracy of our approach on the 400 image dataset using both the PCA and Multi-Angle initializations, and the performance of the associated Polygon Model on the same set. These results are represented pictorially in Fig. 9.

Our approach performs very well, obtaining typical accuracies of within 8 pixels per landmark point and significantly outperforming the Polygonal approach, the shortcomings of which are detailed in Fig. 11.

Moreover, the relative gain of the Skeletal approach on each category is quite telling. As the Towel Model is effectively structureless, there is no distinction between the two models, and hence no improvement. In the case of pants, the proximity between the two legs frequently caused the Polygonal approach to attract to poor local minima; whereas the Skeletal approach, with its implicit knowledge of structure, performed quite well. Short-sleeved-shirts, being fairly homogeneous in shape, proved extremely difficult for the Polygonal approach to fit, as can be readily seen in Fig. 9. Despite the subtlety of shoulder and collar point locations, the longer sleeves of sweaters tend to sketch out a very clear polygonal shape; thus the Polygon Model performed somewhat reasonably, with most errors centered



Category	Polygon Model		Skeletal Model (PCA)		Skeletal Model (Multi-Angle)	
	pixels	cm	pixels	cm	pixels	cm
Towels	$2.89 \pm 1.78$	$0.75 \pm 0.46$	$2.89 \pm 1.78$	$0.75 \pm 0.46$	<b><math>2.86 \pm 1.75</math></b>	<b><math>0.74 \pm 0.45</math></b>
Pants	$14.91 \pm 35.97$	$3.88 \pm 9.35$	$4.23 \pm 1.64$	$1.10 \pm 0.43$	<b><math>4.13 \pm 1.54</math></b>	<b><math>1.07 \pm 0.40</math></b>
Short-Sleeved	$89.63 \pm 44.88$	$23.30 \pm 11.67$	$6.58 \pm 3.14$	$1.71 \pm 0.82$	<b><math>6.41 \pm 3.05</math></b>	<b><math>1.67 \pm 0.79</math></b>
Long-Sleeved	$14.77 \pm 8.27$	$3.84 \pm 2.15$	<b><math>7.09 \pm 3.68</math></b>	<b><math>1.84 \pm 0.96</math></b>	$8.06 \pm 4.52$	$2.09 \pm 1.17$

Fig. 8. Results of fitting our Skeletal Models to the dataset. Model Accuracy is measured as the average pixel distance from the predicted landmark point to the annotated landmark point.

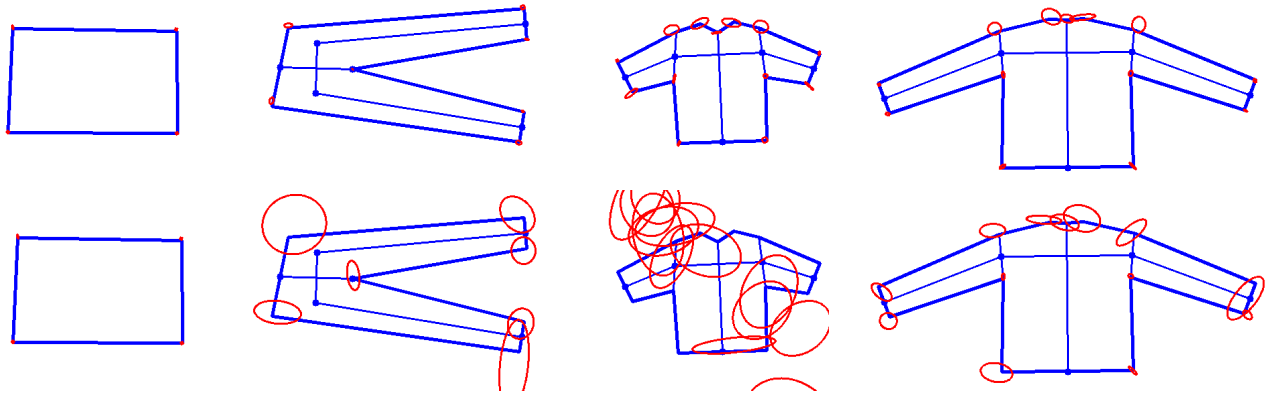


Fig. 9. Comparison of individual landmark point errors. The center of the ellipses denotes mean error, and the size and skew their covariance, projected onto a canonical version of the article. **Top:** Pointwise error for Skeletal Models using the PCA approach. **Bottom:** Pointwise error for Polygon Models.

about shoulders, collars, and sleeve edges.

The results of the Multi-Angle approach were extremely consistent with that of PCA Initialization, suggesting that the latter approach is sufficient for most purposes. Indeed, given the inherent ambiguity in landmark location and small number of examples on which the two differed, any perceived performance advantage would best be attributed to noise.

We then examined the case of unknown clothing category. On 100% of test images, our method was able to accurately classify the clothing category. The classification scheme in Section VII was used to distinguish shirts, pants, and towels. Thresholding the sleeve length at 35% the shirt width further distinguished all long-sleeved shirts from short-sleeved shirts. Therefore, the correct model is always chosen, and the performance is identical to the known, tabulated case.

Our approach, however, was not perfect. The location of collar points proved to be quite ambiguous, and were often incorrectly identified. Shoulders, while significantly localized by structural constraints, still proved a source of difficulty. Finally, the initialization was poor on a small number of instances, and in very rare cases could not be recovered from.

#### E. Robotic Integration

We implemented our approach on the Willow Garage PR2.

In prior work [26], we demonstrated an open-loop method for executing parametrized folds on a clothing article, given a user-specified initial polygon. Using the methods detailed above, we were able to improve this task in two ways:



Fig. 10. Example results of our approach on the four categories of clothing.

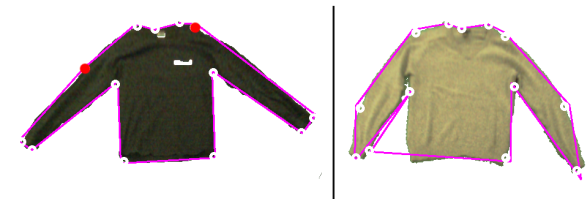


Fig. 11. Failures of the Polygon Model. **Left:** Without more detailed structural information, the model is unable to detect more subtly defined points, such as the shoulder (detected shoulder points in red). **Right:** The unconstrained Polygon approach will generally be attracted to the nearest edge: a poor initialization can easily ruin it.

- The initial polygon is detected automatically. This is done by fitting a Skeletal Model to the observed contour on the table. To avoid grasping virtual points, the generated landmark points are then relocated to their nearest neighbor on the observed contour.
- After each fold, the robot re-examines the cloth. It then fits a Static Folded Model to the new contour, with parameters  $\Theta$  seeded by the robot's intended fold.

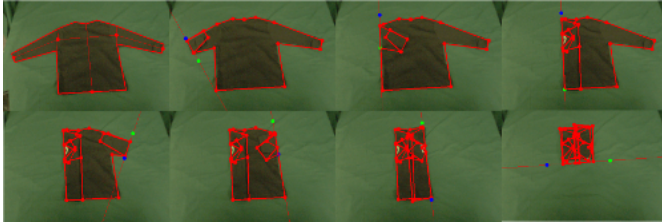


Fig. 12. The robot fits a model to the initial configuration, then tracks each fold in the procedure

We found that our perception approach consistently worked well and had a number successful end-to-end completions on categories of clothing. Videos of representative successful runs as well as software implementations of our algorithm are available at:

[http://rll.berkeley.edu/ICRA\\_2011](http://rll.berkeley.edu/ICRA_2011)

Currently the main limiting factors for end-to-end accuracy are the quality of calibration available for the PR2, and the size of its grippers.

## IX. CONCLUSIONS

We examined the problem of recognizing the pose of an article of clothing in a crudely spread out configuration, and finding the best polygonal representation of it. We proposed a general method for performing such recognition with the use of parametrized shape models, and presented a robust optimization strategy for converging upon the proper parameters. We then detailed particular models for detecting long-sleeved shirts, short-sleeved shirts, pants, and towels, and illustrated the success of such models on a large dataset of test images. Our experiments showed that the resulting energy can also be used to recognize the clothing category. We used this approach to close the loop on a previously-established method for the motion planning aspects of laundry folding, implemented on the Willow Garage PR2.

## ACKNOWLEDGMENTS

This work was supported in part by NSF under award IIS-0904672, Willow Garage under the PR2 Beta Program, and a Feodor Lynen Fellowship granted by the Alexander von Humboldt Foundation.

## REFERENCES

- [1] Dragomir Anguelov, Praveen Srinivasan, Daphne Koller, Sebastian Thrun, Jim Rodgers, and James Davis. Scape: shape completion and animation of people. *ACM Trans. Graph.*, pages 408–416, 2005.
- [2] G. Borgefors. Hierarchical chamfer matching: A parametric edge matching algorithm. *IEEE Transactions on pattern analysis and machine intelligence*, 10(6):849–865, 1988.
- [3] G. Bradski. The OpenCV Library. *Dr. Dobbs's Journal of Software Tools*, 2000.
- [4] C. Bregler and J. Malik. Tracking people with twists and exponential maps. In *Computer Vision and Pattern Recognition, 1998. Proceedings. 1998 IEEE Computer Society Conference on*, pages 8–15, 1998.
- [5] T. Cootes and C. Taylor. Statistical models of appearance for medical image analysis and computer vision, 2001.
- [6] D. Demirdjian, T. Ko, and T. Darrell. Constraining human body tracking. *Computer Vision, IEEE International Conference on*, 2003.
- [7] N. Fahantidis, K. Paraschidis, V. Petridis, Z. Doulgeri, L. Petrou, and G. Hasapis. Robot handling of flat textile materials. *Robotics & Automation Magazine, IEEE*, 4(1):34–41, Mar 1997.
- [8] P. Guan, O. Freifeld, and M. J. Black. A 2d human body model dressed in eigen clothing. *European Conf. on Computer Vision, ECCV, Part I, LNCS 6311*, page 285298, 2010.
- [9] K. Hamajima and M. Kakikura. Planning strategy for task of unfolding clothes. In *Proc. ICRA*, volume 32, pages 145–152, 2000.
- [10] Y. Kita, F. Saito, and N. Kita. A deformable model driven visual method for handling clothes. In *Proc. ICRA*, 2004.
- [11] Hiroko Kobori, Youhei Kakiuchi, Kei Okada, and Masayuki Inaba. Recognition and motion primitives for autonomous clothes unfolding of humanoid robot. In *Proc. IAS*, 2010.
- [12] J. Maitin-Shepard, M. Cusumano-Towner, J. Lei, and P. Abbeel. Cloth grasp point detection based on multiple-view geometric cues with application to robotic towel folding. In *Proc. IEEE Int. Conf. on Robotics and Automation*, 2010.
- [13] G.J. Monkman. Robot grippers for use with fibrous materials. *Int. J. Rob. Res.*, 14(2):144–151, 1995.
- [14] Marius Muja and David G. Lowe. Fast approximate nearest neighbors with automatic algorithm configuration. In *Int. Conf. on Computer Vision Theory and Application (VISSAPP)*, 2009.
- [15] Saul B. Needleman and Christian D. Wunsch. A general method applicable to the search for similarities in the amino acid sequence of two proteins. *Journal of Molecular Biology*, 48(3):443 – 453, 1970.
- [16] F. Osawa, H. Seki, and Y. Kamiya. Clothes folding task by tool-using robot. *Journal of Robotics and Mechatronics*, 2006.
- [17] F. Osawa, H. Seki, and Y. Kamiya. Unfolding of massive laundry and classification types by dual manipulator. *JACIII*, 11(5):457–463, 2007.
- [18] B. Rosenhahn, U. Kersting, K. Powell, R. Klette, G. Klette, and H. Seidel. A system for articulated tracking incorporating a clothing model. *Mach. Vis. App.*, 18, 2007.
- [19] Carsten Rother, Vladimir Kolmogorov, and Andrew Blake. "grabcut": interactive foreground extraction using iterated graph cuts. *ACM Trans. Graph.*, 23(3):309–314, 2004.
- [20] H. Sakoe and S. Chiba. Dynamic programming algorithm optimization for spoken word recognition. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 26(1):43–49, 1978.
- [21] K. Salleh, H. Seki, Y. Kamiya, and M. Hikizu. Inchworm robot grippers in clothes manipulation optimizing the tracing algorithm. In *Intelligent and Advanced Systems, 2007. ICIAS 2007. International Conference on*, pages 1051–1055, Nov. 2007.
- [22] M. Salzmann and P. Fua. Reconstructing sharply folding surfaces: A convex formulation. *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2009.
- [23] M. Salzmann and R. Urtasun. Combining discriminative and generative methods for 3d deformable surface and articulated pose reconstruction. *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2010.
- [24] Hedvig Sidenbladh, Michael Black, and Leonid Sigal. Implicit probabilistic models of human motion for synthesis and tracking. *Computer Vision ECCV 2002*, pages 784–800, 2002.
- [25] R. Urtasun and T. Darrell. Sparse probabilistic regression for activity-independent human pose inference. In *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pages 1–8, 2008.
- [26] J. van den Berg, S. Miller, K. Goldberg, and P. Abbeel. Gravity-based robotic cloth folding. In *Proc. 9th Int. Workshop on Algorithmic Foundations of Robotics (WAFR)*, 2010.
- [27] D.J. Balkcom and M.T. Mason. Introducing robotic origami folding. In *Proc. International Conference on Robotics and Automation*, 2004.
- [28] K. Yamakazi and M. Inaba. A cloth detection method based on image wrinkle feature for daily assistive robots. In *IAPR Conf. on Machine Vision Applications*, pages 366–369, 2009.