3D Non-Rigid Deformable Surface Estimation Without Feature Correspondence

Bryan Willimon, Ian Walker, and Stan Birchfield
Department of Electrical and Computer Engineering
Clemson University, Clemson, SC 29634
{rwillim,iwalker, stb}@clemson.edu

Abstract—We propose an algorithm, that extends our previous work, to estimate the current configuration of a non-rigid object using energy minimization and graph cuts. Our approach removes the need for feature correspondence or texture information and extends the boundary energy term. The object segmentation process is improved by using graph cuts along with a skin detector. We introduce an automatic mesh generator that provides a triangular mesh encapsulating the entire non-rigid object without predefined values. Our approach also handles in-plane rotation by reinitializing the mesh after data has been lost in the image sequence. Results display the proposed algorithm over a dataset consisting of seven shirts, two pairs of shorts, two posters, and a pair of pants.

I. INTRODUCTION

Handling non-rigid objects is a growing area for domestic robotics. Most recently, the laundry process of washing, drying, folding, and storing clothing has been the core study for a lot of work. Such works as folding a T-shirt [1], [2], matching socks [3], and classifying articles of clothing [4], [5], [6], [7], [8], [9], [10], [11], among other tasks, have gained much attention in the past few years. Most recently, we proposed a system that estimated the 3D configuration of a non-rigid object [12] for the purpose of helping the grasping and path planning problem.

In this paper, we continue our work in [12] by using an energy minimization approach to estimate the configuration of a non-rigid object. We have improved our approach by removing the need for feature correspondences and extended the boundary energy term to handle areas of an object with a loss of information. We have increased our dataset to various articles of clothing in order to illustrate the performance of our approach. The results of our approach is tested on changes in translation, scale, in-plane and out-of-plane rotation.

II. PREVIOUS WORK

Other related works that estimate the pose of a non-rigid object [13], [14], people [15], [16], or both [17] use a triangular mesh to determine the non-rigid structure from motion (NRSfM). These works utilize the movement of features within a video sequence to provide the data used to create a triangular mesh that is overlaid on top of the object or person. Not all of the authors use a connected mesh to estimate the configuration of the object, instead Taylor et al. [17] provide a triangle mesh soup that is overlaid on the non-rigid object (e.g. a newspaper). Fayad et al. [13] capture locally rigid patches of an object and combine them together to provide a global pose estimation.

Our work differs from NRSfM in that we calculate the depth of the scene using a single frame while other authors calculate the depth of the scene using a video sequence. Our approach uses mesh configurations from previous frames to estimate the current pose of the object. Unlike [18] [19], our approach does not use fiducial markers to track the current pose of the object. Our approach is capable of handling non-rigid objects without any form of feature correspondence. Our work demonstrates the ability to estimate the configuration of a non-rigid object after it has rotated in-plane to the camera, unlike [20], [21] [22] [23].

This paper reports on new developments designed to extend the work in [12]. In [12], we manually calculated and overlayed the initial mesh onto each object. This paper introduces an automatic mesh generator to initialize the pose of the object without human input. In the previous work, our energy equation consisted of a correspondence term, a depth term, a smoothness term, and a boundary term. The newly proposed energy equation consists of a depth term, a smoothness term, and a redesigned boundary term. We have removed the correspondence term from this equation in order to demonstrate how well our algorithm calculates a triangular mesh with using any sort of texture or feature information. With a new boundary term introduced into this paper, we designed a graph-cuts approach to segment the object from the background using depth and color information. This allows us to accurately locate the contour of each object. We also handle the problem of in-plane rotation by reinitializing our triangular mesh when a large loss of data has occurred. All of the extensions from our previous work are described in later sections.

III. METHOD

A. Mesh Model Generator

The initial mesh is created from the first image of the video sequence using a mesh model generator (MMG). The MMG assumes that the initial configuration of the object is approximately fronto-parallel. The MMG automatically creates a triangular mesh by overlaying a grid of triangles on top of the initial configuration. All triangles that do not contain more than 10% of the object are omitted from the mesh model. The triangles that are not omitted make up the initial mesh, see Figure 1.
B. Extracting contour using graph cuts

The contour extracting approach in this paper uses the min-cut / max-flow algorithm (graph cuts) [24]. The depth and color information is used to provide the foreground / background segmentation. The graph cuts framework comprises of a set of nodes combined by weighted links (n-links). Each node is also connected to a source and a sink with weighted links (t-links). The graph cuts algorithm calculates the binary segmentation within the framework so that a subset of the nodes are connected to the source and the remaining nodes are connected to the sink. The binary segmentation requires that a subset of the n-links and t-links will need to be removed / cut. Since the input images use humans to interact with the objects, a skin detector [25] is needed to segment out the human from the scene. The skin detector used in this paper is trained on 1000 images with hand-labeled skin and non-skin pixels.

The following equations describe the weights placed on the t-links to the source, (1), and the sink, (2).

\[
w_{\text{source}} = \begin{cases} 
\exp\left(\frac{(d_i - \mu_d)^2}{\sigma_d^2}\right), & \text{if not skin} \\
\exp\left(\frac{(d_i - \mu_d)^2}{\sigma_c^2}\right), & \text{if skin}
\end{cases}
\]

(1)

\[
w_{\text{sink}} = 1 - w_{\text{source}}
\]

(2)

where \(d_i\) is the current depth pixel, \(\mu_d\) is the average depth value in the image, \(\sigma_d\) is the deviation at which the object moves within the depth image, and \(d_\infty\) is the highest depth value captured by the sensor.

The next equation describes the weight placed on the \(k\) neighboring n-links.

\[
w_k = \exp\left(\frac{-(d_i - d_{i+k})^2}{\sigma_d^2} + \frac{-(I_i - I_{i+k})^2}{\sigma_c^2}\right)
\]

(3)

where \(I_i\) is the current color pixel, \(d_{i+k}\) and \(I_{i+k}\) are the \(k\)-th neighboring pixels, \(\sigma_d\) and \(\sigma_c\) are the deviations of neighboring pixels in the depth image and color image, respectively. Figure 2 illustrates an example of a shirt segmentation using the graph cuts algorithm and the skin detector.

C. Extended boundary term

For every boundary vertex, we expect it to remain near the boundary of the object even as it undergoes non-rigid deformations. To deal with this situation, the boundary term minimizes the distances between boundary vertices and the 3D boundary points within the image, as determined by the foreground / background segmentation procedure (described in section III-B):

\[
\Psi_B(V) = \frac{1}{2} \sum_{v_i \in B} (g_d(v_i) - v_i)^2 + \frac{1}{2} \sum_{g_i \in \mathcal{G}} (g_i - v_d(g_i))^2
\]

(4)

where \(v_i\) is the boundary vertex, \(g_d(v_i)\) is the nearest boundary point to the vertex \(v_i\) in the current image \(d\), \(g_i\) is the boundary point, \(v_d(g_i)\) is the nearest vertex point to the boundary \(g_i\) in the current image \(d\), \(B\) is the set of boundary vertices, and \(\mathcal{G}\) is the set of boundary points. This term, extended from [12], introduces two lateral forces on the vertices to ensure the mesh fits the data when the object’s motion has components parallel to the image plane, as illustrated in Figure 3. Figure 4 displays an example of a situation when the extended boundary term provides a better mesh representation of the non-rigid object.

D. Reinitialization

When a large amount of input data is lost, the algorithm halts momentarily and begins the reinitialization process. This process is designed to recover the configuration of the object after the transition of being grasped by (two hands)
⇒ (one hand) ⇒ (two hands). The reinitialization process attempts to calculate the current orientation of the object by matching a 3D model to the current input data. The 3D model is found by calculating the average canonical configuration of each object similar to the current object. A 3D plane is then calculated, using a least squares approach, to fit the 3D input data. The equation used to describe the plane is then used to estimate the configuration of the object. The partial derivatives of the boundary term require rewriting using the vector $F_i$:

$$v_i = V^T F_i, \quad (15)$$

The following sections describe the energy terms in detail using the notation and terminology used in [12].

**A. Smoothness term**

The smoothness term is constructed using the same properties described in [12]. The smoothness term is as follows:

$$\Psi_S(V) = \frac{1}{2} (X^T K X + Y^T K Y + Z^T K Z), \quad (7)$$

where $K$ is an $n \times n$ constant matrix capturing the collinear and adjacent vertices in the initial mesh, and let

$$X = [x_1 \ x_2 \ \cdots \ x_n]^T, \quad (8)$$

$$Y = [y_1 \ y_2 \ \cdots \ y_n]^T, \quad (9)$$

$$Z = [z_1 \ z_2 \ \cdots \ z_n]^T, \quad (10)$$

**B. Depth term**

The depth term measures the difference between the $z$ coordinate of each 3D vertex in the current mesh and the measured depth value. That is, it measures the distance along the ray passing through the 3D vertex to the current depth image obtained by the RGBD sensor:

$$\Psi_D(V) = \frac{1}{2} \sum_{i=1}^{n} |d(x_i, y_i) - z_i|^2, \quad (11)$$

where $d(x_i, y_i)$ is the value of the depth image evaluated at the position $(x_i, y_i)$, while $z_i$ is the depth of the vertex at that same position.

**C. Energy minimization process**

Our goal is to locate the mesh that best explains the data while adhering to the smoothness prior, as explained in [12]. To find the configuration of minimum energy, we compute the partial derivative of the energy with respect to the vectors $X$, $Y$, and $Z$, and set the result to zero:

$$\frac{\partial \Psi(V)}{\partial X} = \frac{\partial \Psi_S(V)}{\partial X} + \lambda_D \frac{\partial \Psi_D(V)}{\partial X} + \lambda_B \frac{\partial \Psi_B(V)}{\partial X} \quad (12)$$

$$\frac{\partial \Psi(V)}{\partial Y} = \frac{\partial \Psi_S(V)}{\partial Y} + \lambda_D \frac{\partial \Psi_D(V)}{\partial Y} + \lambda_B \frac{\partial \Psi_B(V)}{\partial Y} \quad (13)$$

$$\frac{\partial \Psi(V)}{\partial Z} = \frac{\partial \Psi_S(V)}{\partial Z} + \lambda_D \frac{\partial \Psi_D(V)}{\partial Z} + \lambda_B \frac{\partial \Psi_B(V)}{\partial Z} \quad (14)$$

The partial derivatives for the smoothness term and the depth term are calculated in [12].

The partial derivatives of the boundary term require rewriting $v_i$ using the vector $F_i$:

$$v_i = V^T F_i, \quad (15)$$
leading to

$$\frac{\partial \Psi_B(V)}{\partial X} = - \sum_{v_i \in B} (g_d(v_i) - X^T F_i) F_i$$
$$- \sum_{g_i \in G} (g_i - X^T F_i) F_i$$
(16)

$$\frac{\partial \Psi_B(V)}{\partial Y} = - \sum_{v_i \in B} (g_d(v_i) - Y^T F_i) F_i$$
$$- \sum_{g_i \in G} (g_i - Y^T F_i) F_i$$
(17)

$$\frac{\partial \Psi_B(V)}{\partial Z} = - \sum_{v_i \in B} (g_d(v_i) - Z^T F_i) F_i$$
$$- \sum_{g_i \in G} (g_i - Z^T F_i) F_i$$
(18)

Rearranging terms leads to

$$(K + \alpha I) X_t = \alpha X_{t-1}$$
$$+ \lambda_B \sum_{v_i \in B} (g_d(v_i) - X_{t-1}^T F_i) F_i$$
$$+ \lambda_B \sum_{g_i \in G} (g_i - X_{t-1}^T F_i) F_i$$
(19)

$$(K + \alpha I) Y_t = \alpha Y_{t-1}$$
$$+ \lambda_B \sum_{v_i \in B} (g_d(v_i) - Y_{t-1}^T F_i) F_i$$
$$+ \lambda_B \sum_{g_i \in G} (g_i - Y_{t-1}^T F_i) F_i$$
(20)

$$(K + \alpha I) Z_t = \alpha Z_{t-1}$$
$$+ \lambda_D \sum_{i=1}^n (d(x_i, y_i) - Z_{t-1}^T F_i) F_i$$
$$+ \lambda_B \sum_{v_i \in B} (g_d(v_i) - Z_{t-1}^T F_i) F_i$$
$$+ \lambda_B \sum_{g_i \in G} (g_i - Z_{t-1}^T F_i) F_i$$
(21)

Solving this equation for the unknowns $X_t$, $Y_t$, and $Z_t$ yields the desired result as an iterative sparse linear system, similar to the energy minimization equations in [12].

V. EXPERIMENTAL RESULTS

We captured RGBD video sequences of seven shirts, one pair of pants, two shorts and two posters to test our proposed method’s ability to handle different non-rigid objects in a variety of scenarios (e.g. change in translation / scale; in-plane and out-of-plane rotation). We also verified the contributions made by the novel depth and boundary energy terms to the accuracy of the estimated object configuration. For our experiments, the weights were set according to $\lambda_B = 0.8$, and $\lambda_D = 0.6$. Further results of the system can be seen in the online video.\(^1\)

\(^1\)http://www.ces.clemson.edu/~stb/research/laundry_pose_estimation

Fig. 8. From top to bottom: Image before in-plane rotation, image after in-plane rotation without reinitializing, binary mask rotated to reinitialize (green outline is mask contour, red outline is actual contour), and image after in-plane rotation with reinitializing. From left to right: Orange shirt, green shirt, and gray shorts.

A. Estimating the pose of different articles of clothing

In this experiment, we test our algorithm on various image sequences involving shirts, pants, and shorts. We create various changes in translation, scale, and rotation on each article of clothing to show how well our algorithm performs. Figure 6 displays each image sequence with the resulting mesh output.

B. Estimating the pose of posters

In this experiment, we test our algorithm on another group of non-rigid objects, e.g. posters. This experiment tests our approach on semi-rigid poster board with out-of-plane oscillations. Figure 7 displays the results on two poster sequences.

C. Reinitializing a mesh after in-plane rotation

Our approach does not perform as well when most of the information is lost within the image sequence. This occurs when the objects are rotated in-plane and are momentarily held by one arm. This loss of information requires a reinitialization phase whenever the object is regrasped by a second arm. Figure 8 illustrates three examples of in-plane rotation and how they result with and without the reinitialization process.

D. Comparison of previous work

This experiment is designed to provide a comparison from our previous algorithm to our current algorithm. Figure 9 illustrates the results of running two sequences through the current and previous algorithms.

649
Fig. 6. From left to right: Six images selected from image sequences involving clothing. From top to bottom: Orange shirt moving out-of-plane, pair of pants moving out-of-plane, white shirt being partially self-occluded, blue shirt moving in-plane and out-of-plane, black shirt changing scale, blue shirt translating from side to side, and checkered shorts moving out-of-plane.

Fig. 7. From left to right: Six images selected from image sequences involving posters. From top to bottom: White poster used in [12]. Poster with various colored regions. Each poster moves out-of-plane of the camera.
VI. CONCLUSION

We have presented an algorithm to estimate the 3D configuration of a non-rigid object through a video sequence without using feature correspondences or texture information. We have improved our previous algorithm by introducing new techniques (i.e., graph-cuts) for improving segmentation and improving existing energy terms (i.e., boundary) to provide an accurate configuration. Our results show how well our new approach can estimate the pose of multiple non-rigid objects in various scenarios.

In the future we plan to extend this research to handle a two-sided 3D triangular mesh that covers both the front and back of the object. Another step is to integrate this algorithm into a robotic system that can grasp and handle non-rigid objects in an unstructured environment.

VII. ACKNOWLEDGEMENTS

This research was supported by the U.S. National Science Foundation under grants IIS-1017007, and IIS-0904116.

REFERENCES