Image and Vision Computing 40 (2015) 16–27

Contents lists available at ScienceDirect

Image and Vision Computing

journal homepage: www.elsevier.com/locate/imavis

Approaches for automatic low-dimensional human shape refinement with priors or generic cues using RGB-D data☆

Mehmet Kemal Kocamaz [a], Christopher Rasmussen [b]

[a] Robotics Institute, Carnegie Mellon University, United States
[b] Department of Computer and Information Sciences, University of Delaware, United States

ARTICLE INFO

Article history:
Received 14 July 2014
Received in revised form 10 May 2015
Accepted 11 May 2015
Available online 19 June 2015

Keywords:
Human shape refinement using RGB-D data
Multi-layer graph cut
Human body shape descriptor
Random decision forests
Refinement of low-dimensional representations
RGB-D

ABSTRACT

Some human detection or tracking algorithms output a low-dimensional representation of the human body, such as a bounding box. Even though this representation is enough for some tasks, a more accurate and detailed point-wise representation of the human body is more useful for pose estimation and action recognition. The *refinement* process can produce a point-wise mask of the human body from its low-dimensional representation. In this paper, we tackle the problem of refining low-dimensional human shapes using RGB-D data with a novel and accurate method for this purpose. This algorithm combines low-level cues such as shape and color, and high level observations such as the estimated ground plane, in a multi-layer graph cut framework. In our algorithm, shape prior information is learned by training a classifier. Unlike some existing work, our method does not utilize or carry features from the internal steps of the methods which provide the bounding box, so our method can work on the outputs of any similar shape providers. Extensive experiments demonstrate that the proposed technique significantly outperforms other suitable methods. Moreover, a previously published refinement method is extended by incorporating more generic cues to serve this purpose.

1. Introduction

Detecting and tracking humans are important tasks for a wide range of computer vision applications, such as human behavior understanding, surveillance systems, autonomous driving, interactive games, and gesture recognition. A rough low-dimensional representation of the human, such as a bounding box, is commonly output by human detection and tracking algorithms [1–14].

Even though this representation is enough for some tasks, a more accurate and detailed point-wise representation is useful to obtain better object descriptors which could be more beneficial for action recognition [15,16] and pose estimation tasks [17,18]. It is possible to obtain a point-wise representation of a human from a low-dimensional representation, a process which can be called shape *refinement*. An illustration of this process using color and depth images can be seen in Fig. 1.

The refinement of a low-dimensional human shape representation is a challenging problem. Several reasons make this process difficult. Representation of the human, most commonly a bounding box in an image, $B(x,y,w,h)$, where x and y is the top left point, w is the width, and h the height of the box, not only contains the human points, but it also includes some background points. The background in the bounding box might have colors, texture, or 3-D geometric features which are similar to those of the human. Additionally, the human might be standing in any position, which causes pose variance and possible self-occlusion of body parts. All of these factors make the refinement process complicated. On the other hand, the bounding box representation provides some hints about the appearance and color of the object. Also, a smaller search space is given to label the points as foreground/background, and the existence of a human in the box is guaranteed.

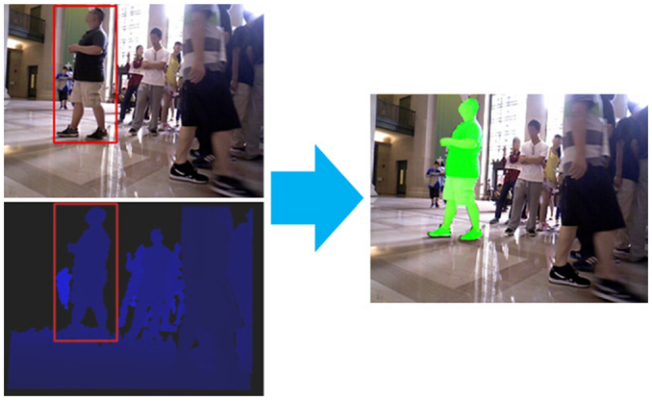

Fig. 1. Our proposed refinement method takes the low-dimensional representation of the human to produce more detailed point-wise representation. It uses both color and depth data for this process.

☆ This paper has been recommended for acceptance by Stan Sclaroff.

E-mail addresses: kocamaz@cmu.edu (M.K. Kocamaz), ras@udel.edu (C. Rasmussen).

http://dx.doi.org/10.1016/j.imavis.2015.05.001
0262-8856/

These hints and advantages do not exist if the human body points need to be found and segmented in the entire image space.

In this paper, a novel algorithm which combines low- and high-level observations obtained from RGB-D data in a multi-layer graph cut framework is proposed for refining a low-dimensional representation of the human shape—a bounding box $B(x,y,w,h)$. We assume that this representation is provided by some other human detection or tracking method. The proposed algorithm does not leverage the internal computations of the methods which provide the bounding box. Hence, it is generic and applicable on the output of any kind of method which produces a similar low-dimensional human shape. A point-wise descriptor is built to employ the shape information of the point neighborhood. This powerful descriptor utilizes the depth data of the scene by generating three cues that are (1) relative geodesic and (2) vectorial spatial distances of a point to the middle point of $B(x,y,w,h)$, and (3) the local structure information encoded as the normal. The descriptors are used to train a Random Forest machine learning algorithm [19] which favors the most important cues in the descriptor. Multiple low- and high-level observations–e.g. the point-wise confidence scores output by the classifier, the color, and estimated ground plane–are combined jointly in a multi-layer graph. The proposed method outperforms existing comparable algorithms in our experiments.

Moreover, a previously published graph cut-based refinement algorithm [20] is extended to serve the same purpose. This method does not incorporate any prior information which must be learned by a classifier as in our first proposed technique. It fuses three generic cues: the color, the depth, and the normal of the points obtained from a single color and depth (RGB-D) image.

A review of related work is summarized in the next section. The human shape descriptor, the proposed classifier, and integration of low- and high-level observations in the multi-layer graph are described in Section 3. The generic cues, and the details of the extended graph cut-based method are explained in Section 4. Experimental results of the proposed and extended methods are analyzed in Section 5. Finally, the proposed work is summarized and possible extensions of this work are drawn in the last section.

2. Related work

GrabCut [21] can be considered as one of the most suitable methods which obtains features from a color image for the refinement of a low-dimensional human shape estimate. *GrabCut* is designed as a semi-automatic segmentation algorithm which takes a box surrounding the object as the input. Gaussian Mixture Models of the object and background are formed by using the regions inside and outside of the input box. Graph cut [22,23] is applied iteratively by feeding the models. At the end of each iteration, border matting is performed to achieve smooth and more accurate results.

The algorithm explained in [24] does not particularly aim to refine the low-dimensional human shape, but it simultaneously detects and obtains person silhouette by integrating top-down and bottom-up approaches in a balanced way. In this method, the pedestrians are simultaneously detected and segmented by integrating appearance and motion cues. It learns the silhouette information from the training data.

Some refinement methods leverage features or confidence scores taken from the internal steps of the methods which provide the low-dimensional human shape [25–29]. These methods do not follow a generic way to refine any given low-dimensional human shape. Hence, they depend on their rough shape estimators. The human silhouette cue computed by the HOG classifier [1] builds the essential parts of the human model used in [27–29]. The faces of the humans are detected by Haar-like features [34] and help to initialize the seed points of *GrabCut* in [28,29]. The method explained in [25] uses some features obtained by applying a human body part detector. A pre-processing step which utilizes Edgelet features defines the region of the interest in [26]. Then, the points of the human body are segmented in this region [26]. *Humanising GrabCut* [30] is a specialized version of *GrabCut* method to refine the low-dimensional human shape. The predictions of the HOG detector are used to build the appearance models to initialize *GrabCut*.

Applying background subtraction techniques is another common approach to segment human body points. The main disadvantage of these methods is that they cannot readily be deployed on moving platforms such as autonomous vehicles. [32,31] introduce multimodal background models for labeling the human points in the scene. They combine the features obtained from thermal and color cameras, where a Gaussian distribution forms a temperature model for the human body and the background models of each pixel in the color image are described by a list of codewords. In [33], the texture and color of each pixel in the image are modeled to segment the humans in indoor scenes. In addition to the image-based features, [35] integrates the shape and height of the human, and the camera model.

The evaluation of methods which are specifically designed for low-dimensional human shape refinement or suitable for this purpose according to different criteria is shown in Table 1. The second column in this table indicates the methods which take a bounding box to refine.

3. Refinement with low- and high-level observations

The method proposed in this section combines in one joint graph cut framework the low-level observations that are the point neighborhood shape and the color information of the point, and a high-level observation that is the estimated ground plane. The point neighborhood shape information of the human body is learned by training a classifier.

3.1. Point-wise descriptor

A point-wise descriptor is formed for each point in the image. The point-wise descriptor, f_s, utilizes the 3-D point cloud of the scene, so the depth image of the scene is converted to a 3-D point cloud. f_s includes the following shape-related cues:

1) Normals: A cue about the local shape information surrounding the point, p_i, can be encoded in the descriptor, f_s, by calculating the normal, η_i, of the point, p_i. It is computed for all three dimensions of the point cloud space, $\eta_i = (\eta_x, \eta_y, \eta_z)$. The neighborhood search of the points is performed by building a FLANN-based Kd-tree [36] to reduce the computation time.
2) Vectorial Spatial Distance: First, the middle point, $mid_B = (mid_x, mid_y, mid_z)$, of the bounding box, $B(x,y,w,h)$, is calculated as formulated in the following equations:

$$mid_B^{2D} = (x + w/2, y + h/2) \tag{1}$$

$$mid_B \leftarrow T\left(mid_B^{2D}\right) \tag{2}$$

where T is the function which gives the corresponding 3-D location of a pixel in the image. The vectorial distance relative to the middle point of $B(x,y,w,h)$, $\Delta_v = (\Delta_x, \Delta_y, \Delta_z)$, is computed for the point $p_i = (p_x, p_y, p_z)$. This computation can be formulated as:

$$\Delta_v = \left(p_x - mid_x, p_y - mid_y, p_z - mid_z\right). \tag{3}$$

3) Geodesic Distance: As mentioned in [37], the geodesic distance between two points on the human body is constant in different poses. This cue is incorporated into our descriptor, f_s. The relative geodesic distance, GD_i, to the middle point, mid_B, of the point, p_i, is computed by Dijkstra's Shortest Path Algorithm. The image is converted to a graph, $G(V, E)$, where V is the graph nodes, and E is the edges between the nodes. Each point, p_i, in the image is represented as a node in the graph, $G(V, E)$. The neighbors of each node in the graph are restricted to 4 pixels. The edge weight, w_{ij}, between two

Table 1
Evaluation of the methods specifically designed for refinement of low-dimensional human shape or can serve for this purpose. Each method is evaluated according to four criteria. These criteria are: 1) Is it designed specifically to refine low-dimensional human shape? 2) Can it work for non-stationary platforms? 3) Is it independent of the low-dimensional shape provider method in terms of carrying some features from the provider? 4) Which sensor data sources are used for the refinement?

Method name	Specifically designed for refinement	Works for non-stationary cameras	Independent of shape provider	Sensors
GrabCut [21]	X	X	X	Color
Sharma and Davis [24]		X	X	Color
Vineet et al. [25]	X	X		Color
Wu and Nevetia [26]	X	X		Color
Migniot et al. [27]	X	X		Color
Vela et al. [28,29]	X	X		Color
Gulshan et al. [30]	X	X		Color & Depth[a]
Zhao and Cheung [31,32]			X	Color & Thermal
Luke et al. [33]			X	Color
Proposed methods	X	X	X	Color & Depth

[a] [30] uses the depth camera data only for labeling some ground truths, not for the refinement.

points is set to the Euclidean distance between p_i and p_j in the corresponding point cloud of the scene as in Eq. (4).

$$w_{ij} = \sqrt{|p_{i_x} - p_{j_x}|^2 + |p_{i_y} - p_{j_y}|^2 + |p_{i_z} - p_{j_z}|^2} \tag{4}$$

If there is no depth data is available for the neighbor, the edge weight, w_{ij} is assigned a large distance.
A sample geodesic distance map for the given image can be seen in Fig. 2.
The proposed point-wise human shape descriptor, f_s, is the combination of the normal, η_i, vectorial distance, Δ_v, and geodesic distance, GD_i of a point p_i. Then, fs becomes:

$$f_s = \left[\eta_x \ \eta_y \ \eta_z \ \Delta_x \ \Delta_y \ \Delta_z \ GD_i\right]^T. \tag{5}$$

3.2. Training the classifier

The human refinement task can be considered as a 2-label classification problem. In short, the label of the first class is "human", while the other label is for the non-human points and is called "background". In order to train a point-wise human classifier, *H-Classifier*, positive human descriptors, f_s^+, and negative human descriptors, f_s^-, are necessary. The samples of f_s^- are chosen from the non-human body points of the scene.

Randomized Decision Forests are a state of the art, fast, and effective machine learning technique [38,19,39,40] which are suitable and applicable for wide range of different tasks and problems [41–43]. Therefore, it is used to train *H-Classifier*. A Decision Forest consists of some number, T, of decision trees. A tree includes split and leaf nodes. Each split node consists of an axis, $f_s(x)$, of f_s, and a threshold τ. To classify the given descriptor of a point, f_s, the split nodes of the decision tree are evaluated by starting from the root of the tree. Whenever a leaf node is hit in a tree, t, a decision distribution, $P_t(d|f_s)$, is obtained.

In the case of low-dimensional human shape refinement problem, $P_t(d|f_s)$ can be considered as a 2-bin histogram. The labels of this histogram are the human and background. The result label of the randomized decision forest classifier can be the average of all distributions given by the trees in the forest:

$$P(d|f_s) = \frac{1}{T}\sum_{t=1}^{T} P_t(d|f_s). \tag{6}$$

Or the result can be the label with the maximum number of votes by each decision tree, t, in the forest as formulated in the following equations:

$$L_t(P_t(d|f_s)) = \begin{cases} 1 & \text{if } \ P_t(d_H|f_s) \geq P_t(d_B|f_s) \\ -1 & \text{otherwise} \end{cases} \tag{7}$$

where $L_t(x)$ is the decision label function of a given decision distribution of a tree, and t. d_H and d_B are the bin values of the human and background labels in the distribution. The normalized confidence score of a point which belongs to the human region becomes:

$$C_i = 0.5 + \frac{1}{T}\sum_{t=1}^{T} L(P_t(d|f_s)). \tag{8}$$

Then, the final decision label of the forest, $L(P(d|f_s))$ becomes:

$$L(P(d|f_s)) = \begin{cases} 1 & \text{if } \ C_i \geq 0.5 \\ -1 & \text{otherwise} \end{cases}. \tag{9}$$

(a) **(b)** **(c)**

Fig. 2. Geodesic distance calculation. (a) Color image with overlaid bounding box of the object, (b) depth image, and (c) colored geodesic distance map of the given image. The red point corresponds to the middle point of the bounding box. The points which have darker intensity in the map are farther from the middle point and lighter points are closer to it.

Each tree is trained on a different set of randomly selected positive and negative samples using the following algorithm [42]:

1) Randomly obtain a set of splitting candidates for a tree node, $\Phi = (f_s(x), \tau)$. $f_s(x)$ is an axis of a point-wise descriptor, and τ is the split threshold.
2) The set of training points, $S = \{p_i\}$, are divided into two sets, S_l and S_r, for left and right leaves of the node by each Φ:

$$S_l(\Phi) = \{p_i \mid f_s(x) \le \tau\} \tag{10}$$

$$S_r(\Phi) = S - S_l(\Phi). \tag{11}$$

3) Find the best splitting candidate, Φ^*, which produces the largest information gain:

$$\Phi^* = \underset{\Phi}{\operatorname{argmax}}\, G(\Phi) \tag{12}$$

$$G(\Phi) = H(S) - \sum_{\psi \in (l,\ r)} \frac{|S_w(\Phi)|}{|S|} H(S_w(\Phi)) \tag{13}$$

where $H(S)$ is the Shannon Entropy. It is computed on the normalized distribution of the labels of the points in the set of S as in the following equation:

$$H(S) = -\sum_{i=1}^{n} Pr(l_i|P_L)\ log_2 Pr(l_i|P_L) \tag{14}$$

where P_L is the label distribution in the set S, and l_i is the label name.
4) If the current depth of the tree is under a maximum threshold, create left and right children of the current node by using left and right subsets, $S_l(\Phi^*)$ and $S_r(\Phi^*)$.

3.3. High-level observation and color discontinuity

Graph cut [44,22,23,45] provides a powerful framework to produce globally optimal segmentation results. Its graph structure enables the combination of multiple different kinds of features in one joint framework. In our approach, graph cut is chosen as the infrastructure to incorporate the cues for a joint final solution.

It is difficult to generalize the color models of the human and background for all possible scenes. The point-wise descriptor, f_s, described in the previous section does not include the color cue of the human body. However, the refinement procedure can utilize the discontinuity of the color between points in the scene. This can be achieved by employing the color discontinuity in the graph cut framework.

The idea of putting high-level observations into graph cut was first introduced in [46]. In order to incorporate high-level observations, a second layer of nodes are added to the standard first layer of the nodes. Each node in the second layer represents a high-level observation defined by a group of the points in the first layer. In our case, one node is added to the second layer to represent the estimated ground plane points. The interactions between the first and second layers are established in a way that a second-layer node is connected to some of the nodes in the first level. These connected nodes in the first level define the high-level observation.

All points of a high-level observation could be treated as the background. However, there is an important drawback of this assumption. If the high-level observation is obtained by some estimation process, they might include some foreground points. For example, some points of the foot are estimated as the ground plane points as can be seen in Fig. 3. In the graph cut framework, it is desirable to state two attributes of these points. First, they all together define an observation. Second, some of the points within this observation might be misclassified by the estimator, and these are subject to modification of their labels.

3.3.1. Multi-layer graph

A multi-layer undirected graph, $G_{Multi} = (V, \mathcal{E})$, is defined by a set of nodes, V, and a set of edges, $\mathcal{E}$. The set of nodes consists of two subsets. The first subset of the nodes, V_L, are the first-layer nodes which represent the low-level observations. Each point in the scene is defined by a node, n_i, where $n_i \in V_L$. The second subset of the nodes, V_H, represents the high-level observations employed in the second layer of the graph G. In our case, V_H consists of a single node, n_H, which is for the estimated ground plane. Thus, the set of the nodes, V, becomes $V = \{n_1, \ldots, n_k\} \cup \{n_H\}$, where k is the number of the points in the image.

The set of edges, $\mathcal{E}$, consists of two types. The low-level interactions between the points in the first layer are formed by a subset of the edges, denoted by $\mathcal{E}_L$. $\mathcal{E}_L$ consists of the edges, $e_{i,j}$, between two neighbor nodes, n_i and n_j, in V_L. The connections between the low- and high-level observations are established by the edges, $\mathcal{E}_H$. Each node, $n_i \in V_L$, in the low-level, is connected to the node, $n_H \in V_H$, in the second level by the edges, $e_{i,H}$. Thus, $\mathcal{E}$ becomes $\mathcal{E} = \{e_{1,2}, \ldots, e_{i,j}, \ldots, e_{k-1,k}\} \cup \{e_{1,H}, \ldots, e_{k,H}\}$. The structure of the multi-layer graph is illustrated in Fig. 4.

Segmentation of the multi-layer graph, $G_{Multi} = (V, \mathcal{E})$, is equivalent to assigning a label l_i, from a set of labels, $\{l_H, l_B\}$, to each node, n_i, in V. In this case, l_B refers to the background, and l_H refers to human points. The set of all labels assigned to the nodes in V is denoted by $\tilde{L}$. The final mask of the human is formed by the points whose labels are assigned to l_H by the graph cut algorithm.

3.3.2. Graph cut energy functions

The energy function of the multi-layer graph cut consists of two terms, namely the regional term, R, and the boundary term, B, as in the standard graph cut energy equation:

$$E\left(\tilde{L}\right) = \sum_{i \in V} R(l_i) + \sum_{\{i,j\} \in V} B_{i,j}(l_i, l_j) \tag{15}$$

where i and j are the nodes of any edge, $e_{i,j}$, in $\mathcal{E}$.

(a) **(b)** **(c)**

Fig. 3. Ground plane estimation. (a) shows a given image, (b) is its depth image, and (c) displays its corresponding 3-D point cloud with registered color of the points and the estimated ground plane. The points of the estimated ground plane are colored as red.

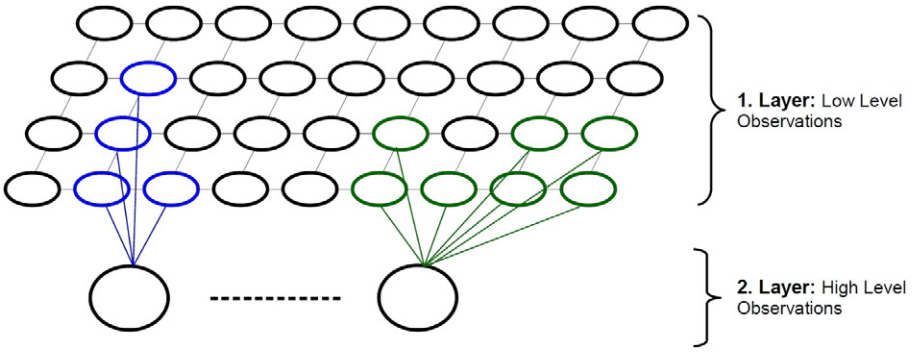

Fig. 4. Illustration of multi-layer graph structure. Each node that represents a high-level observation in the second layer is connected to some nodes in the first layer.

The regional term of the multi-layer graph cut employs the confidence score of the $H-Classifier$, $\mathcal{C}_i$, for each point as defined in Eq. (8). Also, the high-level observation which is the ground plane estimation is incorporated into the regional term. More precisely, the regional term of the multi-layer graph cut energy function becomes:

$$\sum_{i\in V} R(l_i) = \sum_{i\in V_L} -ln(p_c(l_i)) + \alpha(\mathcal{H}_L(l_H)). \tag{16}$$

The first term in the above equation describes the confidence score of $H-Classifier$. The second term defines the high-level observation. α sets the relative influence between the two terms.

The likelihood of being in the object region of a point, p_c, is formulated using the confidence score, $\mathcal{C}_i$, produced by $H-Classifier$ as following:

$$p_c(l_i) = \begin{cases} \mathcal{C}_i & \text{if} \quad l = \text{"human"} \\ 1-\mathcal{C}_i & \text{if} \quad l = \text{"background"} \end{cases}. \tag{17}$$

$\mathcal{H}_L$ defines the likelihood function of the high-level observation node for a given label, l_H. $\mathcal{H}_L(l_H)$ is set to 1 if l_H is *background* and 0 if l_H is *foreground*. In our case, the estimated ground plane is considered simply as the background.

The color discontinuity and the interactions between low- and high-level observations are defined in the boundary term of the energy function, $E(\tilde{L})$. As in [47], the color discontinuity between neighbor points in the first layer of the graph is formed by the following equation:

$$B{-}Color_{i,j\in V_L} = \lambda_1 \frac{1}{dist(i,j)} e^{-\left(\|c_i-c_j\|^2\right)/2\sigma^2} \tag{18}$$

where c_i and c_j are the colors of the points i and j, $dist(i,j)$ is the standard L_2 Euclidean norm yielding point distance, and σ^2 is the average squared norm in the image.

The graph edge between one node of the first layer and the high-level observation node depends on the distance between the normal of the point, $\eta_i = (\eta_x, \eta_y, \eta_z)$, in the estimated ground plane and the estimated normal of the ground plane, $\overline{\eta_H}$. It is defined by:

$$B{-}High_{i\in V_L} = \lambda_2 e^{-\left(\|\eta_i-\overline{\eta_H}\|^2\right)/2\sigma_H{}^2} \tag{19}$$

where σ_H is the averaged squared distance between the normal of the points, η_i, and estimated normal of the ground plane, $\overline{\eta_H}$. λ_1 and λ_2 are to weight these two boundary terms.

The final labeling, $\widetilde{L_F}$, can be achieved by minimizing the energy function in Eq. (15):

$$\widetilde{L_F} = \underset{\tilde{L}}{\text{argmin}} \; E\left(\tilde{L}\right). \tag{20}$$

The graph cut algorithm in [45] is used to minimize this equation. The proposed refinement process which utilizes the multi-layer graph cuts and $H-Classifier$ is called $H-Classifier_{Multi-GC}$ after this point.

4. Refinement using only generic cues

It is possible to develop some point-wise human refinement algorithms without incorporating any learned prior information. In these approaches, the models of the human and background are obtained from a single input image. The models can include some generic cues, such as the color, depth, normals, and edges. All points inside of the bounding box can be used to form the foreground model. Or some pre-processing steps can be applied to remove some of the background points from the inside of the given bounding box. Applying pre-processing steps can produce more reliable foreground model.

The algorithm explained in [20] is a previously published graph cut-based refinement method. It takes the low-dimensional shape of any object and outputs its point-wise representation. This method uses only a monocular color camera data source. The foreground/background models are constructed by using the regions which are obtained by scaling the given initial low-dimensional shape. These models incorporate color and shape distance terms. We extended this method to serve as a low-dimensional human shape refinement algorithm. In addition to the color, we incorporated raw depth and normal information to build the models of this algorithm. Algorithm 1 outlines the steps of the extended method.

Algorithm 1. *GC-Refine.*

Inputs: Bounding box $B(x,y,w,h)$, color image, I_c, depth image, I_d
Output: Human body points, P_H
* Obtain foreground, R_F, and background masks, R_B, by scaling $B(x,y,w,h)$
* Compute normal of the points, η_p
* For each point in the image, form a feature vector, F_p, including raw depth, color and normal
* Scale each axis of F_p between 0 and 1
* Cluster features, F_p, in the image by *k-means*
* Obtain foreground, M_F, and background, M_B, models which are the histograms formed by *k-means* label frequencies inside the masks, R_F and R_B
* Compute distance penalty map, D_M, relative to R_F
* Set the edge weights of the graph using M_F, M_B, and D_M
* Apply graph cut to obtain human body points, P_H

Finding a good scale factor plays an important role in constructing discriminative foreground and background models. Using a small scale

factor causes the inclusion of some background points in the human model. On the other hand, choosing a large scale factor can erroneously eliminate some of the human body points. In order to reduce the number of the background pixels in the human model, some pre-processing steps can be applied. Therefore, the ground plane is estimated and excluded from the foreground model.

Removing the ground plane points from the foreground model is helpful only at the foot level of the human. However, there might remain some other background points in the foreground model, e.g. the points of a wall, an object, or another human. In order to remove these points, it is assumed that a human (modeled roughly as an upright cylinder) has a maximum radius of d_H. A slice of region which is perpendicular to the ground and whose radius is d_H is searched to extract the region of interest within $B(x,y,w,h)$. The slice which holds the maximum number of points is selected to form the foreground model. This slice can be estimated by a Random Sample Consensus (RANSAC) procedure as outlined in Algorithm 2. The method which includes these pre-processing steps is called *GC-Refine-Pre*.

Algorithm 2. Extracting ROI.

Inputs: $Pts = (x_1, y_1, z_1), ..., (x_n, y_n, z_n)$ points in $B(x, y, w, h)$, d_H
Output: $RegionPts = (x_1, y_1, z_1), ..., (x_m, y_m, z_m)$, the inlier points

```
* Compute L, the iteration number of RANSAC
for k=1, ... , L
* Choose a random point, p_k = (x_k, y_k, z_k), in Pts
* Set, c_in ← 0, the number of inliers
* Set, pts_in ← empty, the inlier points
  for each p_i = (x_i, y_i, z_i), in Pts
  * Calculate the distance, d_z = |z_k − z_i|, between p_k and p_i along z axis
    if d_z < d_H
    * Add p_i to pts_in
    * c_in ← c_in + 1, increment the number of inliers
  if c_in > c_max
  * RegionPts ← pts_in, copy inliers to the return list
  * c_max ← c_in, set max number of inliers
```

5. Experiments

Several experiments were conducted to quantify and analyze the performance of the proposed methods *H — Classifier*, *H — Classifier*$_{Multi-GC}$, *GC-Refine*, and *GC-Refine-Pre*. For these experiments, a subset dataset of *DontHitMe*, called as *DontHitMe-Refine*, was collected. The details of *DontHitMe* are explained in [10]. Briefly, this dataset includes low-dimensional (a bounding box) ground truths of 3600 humans both in color and registered depth images. In addition to these low-dimensional representations, *DontHitMe-Refine* contains point-wise ground truths of 1016 human images which were manually annotated.

The positive samples are obtained from the ground truth masks of *DontHitMe-Refine* dataset to train the classifier. All points inside a ground truth mask are used to generate the positive samples. However, not all points outside of the ground truth region are selected as negative samples. Only one of every two pixels in a row of the image is added to the set of the negative samples. In this way, the training time of the classifiers is aimed to be reduced. Also, the points which do not have valid depth data are not included in the training set. 5 decision trees were trained.

5.1. Analyzing the performance of shape cues and multi-layer graph cut

We performed a set of tests to analyze the performance of *H — Classifier* when it is trained with the point-wise descriptors, f_s, which includes different combination of the cues. The normal, η, vectorial spatial distance, Δ_v, and the geodesic distance, *GD* cues were combined in f_s in 7 different possible ways. A different *H — Classifier* was trained for each of these combinations using the same training set. In order to reduce the variability in the testing scores, we performed multiple rounds of 5-fold cross-validation. The following polygon area overlap formula is used to measure the overlap between the ground-truth and the result of the classifier suggested by [48]:

$$O(\mathcal{R}_1, \mathcal{R}_2) = A(\mathcal{R}_1 \cap \mathcal{R}_2)^2 / (A(\mathcal{R}_1)A(\mathcal{R}_2)) \quad (21)$$

where $\mathcal{R}_1$ and $\mathcal{R}_2$ are the two regions to calculate the overlap between.

The results of this experiment can be seen in Table 2. The best performance, for which the median overlap score is 0.89, was achieved when all of three cues were included in f_s. In the case of removing one of the cues from f_s, the scores dropped down. Also, the classifier which uses only the normals, η, was unable to distinguish between background and human points. Thus, the normal, η, alone is not capable of representing the human body points. However, when it is associated with the vectorial spatial distance, Δ_v, they both performed well by achieving the overlap score of 0.81.

In addition to three cues related to the shape, two more tests were conducted to see the effect of incorporating the color of the points into the point-wise descriptor, f_s. Simply, the three channels of RGB color of the points are included in f_s as additional dimensions. The same test was also performed over converting to LAB color space. The median overlap scores of these two classifiers which include the color information are shown in the last two rows of Table 2. Adding the color information reduced the performance of the classifier from 0.89 to 0.83. This is mainly because of the color variation of the human skin, clothes, background and different illumination conditions. No performance difference between different color spaces was observed. They both dropped the overlap scores by the same amount.

Furthermore, the images of *DontHitMe-Refine* were refined by *H — Classifier*$_{Multi-GC}$. The confidence scores of *H — Classifier* which produced the best median overlap score in Table 2 by combining three shape-related cues, Δ_v, η, and *GD*, in f_s were used in the multi-layer graph cut framework. This process can be considered as a second stage in the refinement process. The median overlap scores of *H — Classifier*$_{Multi-GC}$ is listed in Table 3. A remarkable improvement was achieved by this step. Incorporating the estimated ground plane as a high-level observation, utilizing the shape confidence score of *H — Classifier*, and employing the color information jointly in a multi-layer graph took the median overlap score from 0.89 to 0.95. Fig. 5 displays some examples refined by *H — Classifier* and *H — Classifier*$_{Multi-GC}$. (a) and (c) of this figure show the results of *H — Classifier*. (b) and (d) display the refinements results of *H — Classifier*$_{Multi-GC}$ for the same input images. In both of these results, the proposed multi-layer graph cut approach helped to remove some ground points and also background points classified as the human by *H — Classifier*. The color discontinuity term in the graph cut enabled the segmentation of points where there is no depth data is available. For example, some of the hair points on the left side of the neck of the woman in Fig. 5(a) do not have valid depth information. Hence, *H — Classifier* was unable to classify those points. Yet, *H — Classifier*$_{Multi-GC}$ included them into the final refined region as shown in Fig. 5(b), because of the color similarity to other hair points which have valid depth.

Table 2
Median overlap scores of $H-Classifier$ for different cue combinations in its descriptor, f_s. For example, $\Delta_v+\eta$ means the vectorial spatial distance and the normal of the points are used in the descriptor, f_s.

Method	Median overlap score
Only Δ_v	0.71
Only η	0.58
Only GD	0.66
$\Delta_v+\eta$	0.81
Δ_v+GD	0.78
$\eta+GD$	0.73
$\Delta_v+\eta+GD$	**0.89**
$\Delta_v+\eta+GD+$ RGB	0.83
$\Delta_v+\eta+GD+$ LAB	0.83

Table 3
The overlap scores of the methods for *DontHitMe-Refine* data set.

Method Name	Score
$H-Classifier_{Multi-GC}$	**0.95**
H-Classifier	0.89
GC-Refine-Pre	0.70
H-Classifier-SVMs (trained by SVMs)	0.67
GC-Refine	0.64
Baseline method	0.64
Depth difference features from [49,50]	0.62
GrabCut [21]	0.58

5.2. Other methods for comparison

The proposed $H-Classifier$ and $H-Classifier_{Multi-GC}$ are compared to the following methods:

1) *Baseline Method*: We implemented a simple baseline approach. It centers a cylinder of a fixed radius 0.5 m and height 2 m at the 3-D center of the human determined via the bounding box. The ground plane points are removed from inside the cylinder and all other points are taken as the object.
2) *GrabCut* [21] which is naturally suitable for the refinement of low-dimensional human shape representation. It uses only the color image.
3) The method explained in [49,50] was developed originally for labeling different body parts. We used the depth difference features of this method to classify points as human or background. The same set of images from *DontHitMe-Refine-Train* and samples are used to train this classifier. In this case, the human body part labeling problem is reduced to a simple human/background labeling problem.

The proposed $H-Classifier$ and $H-Classifier_{Multi-GC}$ methods performed better than all compared methods. The overlap scores of the methods can be seen in Table 3. Fig. 6 displays a gallery of the results of the methods. In our tests, the poor performance of *GrabCut* can be explained by the color similarities between the fore/background models, the lack of structural information, and possible background points inside the human model. Also, our proposed descriptor, f_s, is more successful than the depth similarity features of [49,50] at distinguishing the human points from the background points for the same training and test sets.

We observed that the clustering method of *GC-Refine* is not able to weight the cues according to their importance in different parts of the scene. Since it does not employ a learning technique which favors the important cue in the places where it is more discriminative than other cues, $H-Classifier$ outperformed its classification results. Fig. 7 shows some results of *GrabCut*, *Baseline method* and the proposed $H-Classifier_{Multi-GC}$ for direct comparison.

5.3. Random decision forests vs support vector machines (SVMs)

In order to analyze the performance difference between Random Decision Forests and SVMs, another point-wise human classifier, $H-Classifier-SVMs$, was trained using SVMs [51]. A Radial Basis Function

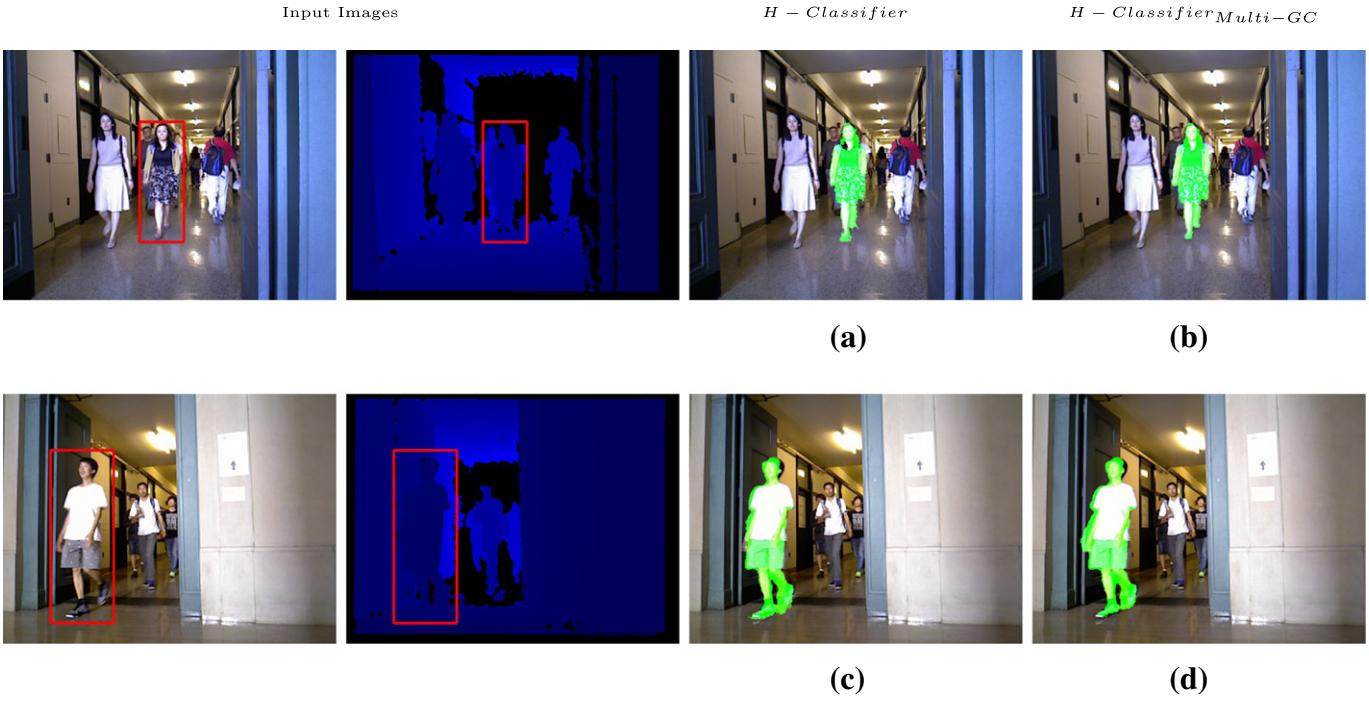

Fig. 5. Comparison between the results of $H-Classifier$ and $H-Classifier_{Multi-GC}$.(a) and (c) display classification results of $H-Classifier$. (a) and (c) show the refinement outputs of $H-Classifier_{Multi-GC}$ for the same image set. $H-Classifier_{Multi-GC}$ improves the results by eliminating more background points at the foot level of the human. Also, its color discontinuity term helps to classify some points correctly on the human body which have no valid depth data, such as in the hair of the woman in (a) and (b).

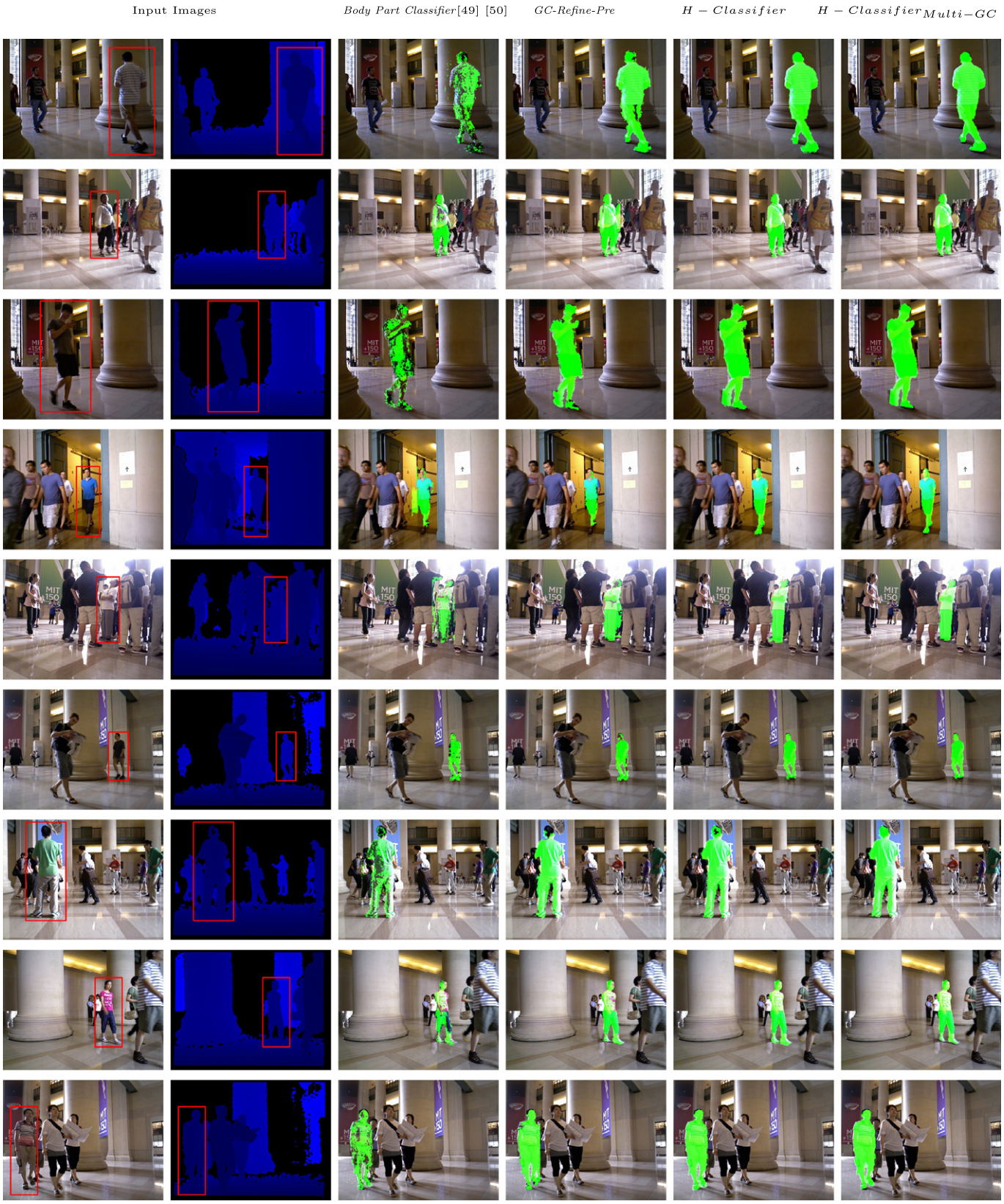

Fig. 6. Sample results of different refinement methods. Column headings show the names of the methods. The second column shows the depth images of scenes. Only the results of the cue combinations which produced the best overlap scores are illustrated for *GC-Refine-Pre* and *H − Classifier*. Please note that *Body Part Classifier* [49,50] does not use color image data.

(RBF) was used as the kernel function of the SVM. The same set of point-wise descriptors, f_s, to build *H − Classifier*, were used to train *H − Classifier − SVMs*. Also, all dimensions of f_s are scaled between 0 and 1 to assign the same weight to different cues in f_s. In this way, the same conditions were established to analyze and compare their performances. SVMs performed worse than Random Forests in the experiments.

Fig. 7. Sample results of *Baseline method*, *GrabCut* and $H - Classifier_{Multi-GC}$. Column headings show the names of the methods. Please note that *GrabCut* does not use the depth data.

The poorer performance of SVMs is related to its decision function. SVMs measure the distance to the hyperplane which is the decision margin of the space to classify a point. In this decision function, even though some of the dimensions of f_s are less discriminative than others, all cues in f_s take the same amount of importance, so they still affect the distance to the hyperplane. However, our descriptor includes different types of cues which may require different weights or pruning in the training and testing phases of the classifiers. SVMs does not contain a mechanism to balance the weights between different cues or to prune some of them in a decision tree way. Fortunately, Random Decision Forests supply these features. Some sample results of *H — Classifier — SVMs* can be seen in Fig. 8.

5.4. Performance of GC-Refine

Several tests which incorporate different combinations of the generic cues were performed to see the results of *GC-Refine*. Seven different combinations of three generic cues that are the color, depth and normal were used in the experiments. *GC-Refine* was separately tested with each possible combination of the cues. All images in *DontHitMe-Refine* dataset were refined in these tests. The median overlap scores of these experiments can be seen in Table 5. The highest performance was achieved when the feature vector of *GC-Refine* included all cues. Its median overlap score is 0.64. The combination of the depth and normal cues produced the second-best results.

Fig. 8. Sample results of *H — Classifier — SVMs* and *H — Classifier*. Column headings show the names of the methods.

In order to analyze the performance of extracting the non-region of interest from the human model, we experimented with *GC-Refine-Pre* in the same way. As in the previous tests of *GC-Refine*, all different combinations of the cues were analyzed. As it can be foreseen, providing cleaner models yielded better refinement results. The median overlap scores of this experiment are summarized in Table 6. The best score was achieved if all cues are used. The score went up to 0.70 from 0.64 which was achieved by *GC-Refine*.

Table 4
Average running time of the methods to classify the human points in one bounding box. They include the required time for pre-processing steps and computing the features.

Method name	Time (in sec)
$H - Classifier_{Multi - GC}$	0.7
H-Classifier	0.6
GC-Refine-Pre	0.47
H-Classifier-SVMs	1.9
GC-Refine	0.4
Baseline method	0.05
Depth diff features from [49,50]	0.02
GrabCut [21]	0.2

Table 5
Median overlap scores of *GC-Refine* for *DontHitMe-Refine* data set. Each row specifies an experiment in which different combinations of the cues used in the refinement process. For example, RGB + Depth means that the combination of RGB color and depth cues were incorporated into *GC-Refine* for that test.

Cues	Median overlap score
Only RGB	0.43
Only Depth	0.57
Only Normal	0.47
RGB + Depth	0.59
RGB + Normal	0.49
Depth + Normal	0.61
RGB + Depth + Normal	**0.64**

Table 6
Median overlap scores of *GC-Refine-Pre* for *DontHitMe-Refine* data set. As it is defined in Table 5, each row specifies different combinations of the cues used in the process.

Cues	Median overlap score
Only RGB	0.47
Only Depth	0.61
Only Normal	0.52
RGB + Depth	0.64
RGB + Normal	0.53
Depth + Normal	0.65
RGB + Depth + Normal	**0.70**

5.5. Computational load

A machine which has 32 GB RAM and Intel i7-2760QM quad processor was used in the experiments. The methods were implemented in C++. The implementations do not contain thread-level parallel processing. 5 decision trees were trained for $H-Classifier$. Since we performed 5-fold cross-validation, about 800 ground truths were trained for each test. It took about 7.5 h to train $H-Classifier$ including the time to compute the features. The training time of $H-Classifier-SVMs$ using SVMs took longer, or about 12 h. Training the classifier which uses depth difference features from [49,50] took about 8.5 h.

The running times of the methods to classify the human points in one bounding box can be seen in Table 4. The size of the images was not scaled to reduce the computational time. The image size is 640 × 480 for all experiments. The classifier which uses depth difference features from [49,50] was the fastest. Since $H-Classifier$, $H-Classifier-SVMs$ and $H-Classifier_{Multi-GC}$ include the steps to compute the normal of the points and their relative geodesic distances, they are slower than other methods. We believe that a parallel implementation of the proposed method can decrease the required computational time.

6. Conclusion

In this paper, we tackled the problem of low-dimensional human shape refinement in two different ways: by combining shape prior information learned by training a classifier, or by using only some generic cues obtained from given single image. We presented a novel and accurate method to refine the low-dimensional shape representation of a human. This method, $H-Classifier_{Multi-GC}$, combines low- and high-level observations obtained from the image and depth images of the scene jointly in a multi-layer graph framework. Unlike some existing work, our approach does not use or carry any features from the internal steps of the low-dimensional shape provider, so it is applicable to the output of any methods which provides such a shape. Also, it works on moving platforms and integrates multiple modalities by obtaining cues from the color and depth images. On the other hand, we extended a previously-published and graph cut-based refinement technique for this purpose. In addition to the color, we incorporated more generic cues that are the depth and normal of the points into this method.

Our extensive experiments showed that the proposed $H-Classifier_{Multi-GC}$ outperforms other suitable refinement algorithms. It achieves a 0.92 overlap score while *GrabCut* stays at 0.57. As future work, other high-level observations, such as estimated walls, or detected objects can be incorporated into the multi-layer graph framework in addition to the estimated ground plane.

References

[1] N. Dalal, B. Triggs, Histograms of oriented gradients for human detection, Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR) 2005, pp. 886–893.
[2] M.D. Breitenstein, F. Reichlin, B. Leibe, E. Koller-Meier, L. Van Gool, Online multiperson tracking-by-detection from a single, uncalibrated camera, IEEE Trans. Pattern Anal. Mach. Intell. 33 (9) (2011) 1820–1833.
[3] M. Luber, L. Spinello, K.O. Arras, People tracking in RGB-D data with on-line boosted target models, Proceedings of the International Conference on Intelligent Robots and Systems (IROS), 2011.
[4] L. Spinello, K.O. Arras, People detection in RGB-D data, Proceedings of the International Conference on Intelligent Robots and Systems (IROS), 2011.
[5] A. Ess, B. Leibe, K. Schindler, L. van Gool, Robust multiperson tracking from a mobile platform, IEEE Trans. Pattern Anal. Mach. Intell. 31 (10) (2009) 1831–1846.
[6] O. Tuzel, F. Porikli, P. Meer, Human detection via classification on Riemannian manifolds, Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR) 2007, pp. 1–8.
[7] L. Spinello, R. Triebel, R. Siegwart, Multimodal detection and tracking of pedestrians in urban environments with explicit ground plane extraction, Proceedings of the International Conference on Intelligent Robots and Systems (IROS), 2008.
[8] T. Kim, S. Cho, J. Yoon, D. Kim, Pose robust human detection in depth image using four directional 2-D elliptical filters, Proceedings of IEEE International Symposium on Multimedia (ISM) 2009, pp. 130–135.
[9] S. Ikemura, H. Fujiyoshi, Real-time human detection using relational depth similarity features, Proceedings of the Asian Conference on Computer Vision (ACCV) 2011, pp. 25–38.
[10] M.K. Kocamaz, F. Porikli, Unconstrained 1-D range and 2-D image based human detection, Proceedings of the International Conference on Intelligent Robots and Systems (IROS), 2013.
[11] C. Premebida, O. Ludwig, U. Nunes, Lidar and vision-based pedestrian detection system, J. Field Rob. 26 (9) (2009) 696–711.
[12] L. Spinello, R. Siegwart, Human detection using multimodal and multidimensional features, Proceedings of the International Conference on Robotics and Automation (ICRA), 2008.
[13] Z. Zivkovic, B. Krose, Part based people detection using 2-D range data and images, Proceedings of the International Conference on Intelligent Robots and Systems (IROS), 2007.
[14] R. Benenson, M. Mathias, R. Timofte, L.J.V. Gool, Pedestrian detection at 100 frames per second, Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR) 2012, pp. 2903–2910.
[15] L. Chen, H. Wei, J. Ferryman, A survey of human motion analysis using depth imagery, Pattern Recogn. Lett. 34 (15) (2013) 1995–2006.
[16] R. Vemulapalli, F. Arrate, R. Chellappa, Human action recognition by representing 3-D skeletons as points in a lie group, Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2014.
[17] M. Wai, R. Nevetia, Body part detection for human pose estimation and tracking, Proceedings of IEEE Workshop on Motion and Video Computing, 2007.
[18] T.-H. Yu, T.-K. Kim, R. Cipolla, Unconstrained monocular 3-D human pose estimation by action detection and cross-modality regression forest, Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2013.
[19] L. Breiman, Random forests, Mach. Learn. 45 (1) (2001) 5–32.
[20] M.K. Kocamaz, Y. Lu, C. Rasmussen, Deformable object shape refinement and tracking using graph cuts and support vector machines, International Symposium on Visual Computing 2011, pp. 506–515.
[21] C. Rother, V. Kolmogorov, A. Blake, Grabcut: interactive foreground extraction using iterated graph cuts, ACM Trans. Graph. 23 (2004) 309–314.
[22] Y. Boykov, O. Veksler, R. Zabih, Fast approximate energy minimization via graph cuts, Proceedings of the International Conference on Computer Vision (ICCV), 1999.
[23] Y. Boykov, O. Veksler, R. Zabih, Fast approximate energy minimization via graph cuts, IEEE Trans. Pattern Anal. Mach. Intell. 23 (11) (2001) 1222–1239.
[24] V. Sharma, J.W. Davis, Integrating appearance and motion cues for simultaneous detection and segmentation of pedestrians, Proceedings of the International Conference on Computer Vision (ICCV) 2007, pp. 1–8.
[25] V. Vineet, J. Warrell, L. Ladicky, P. Torr, Human instance segmentation from video using detector-based conditional random fields, Proceedings of British Machine Vision Conference (BMVC) 2011, pp. 80.1–80.11.
[26] B. Wu, R. Nevatia, Simultaneous object detection and segmentation by boosting local shape feature based classifier, Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2007.
[27] C. Migniot, P. Bertolino, J.-M. Chassery, Iterative human segmentation from detection windows using contour segment analysis, International Conference on Computer Vision Theory and Applications (VISAPP) 2013, pp. 405–412.
[28] A. Hernandez-Vela, M. Reyes, V. Ponce, S. Escalera, Grabcut-based human segmentation in video sequences, Sensors 12 (11) (2012) 15376–15393.
[29] A. Hernández-Vela, M. Reyes, S. Escalera, R. Petia, Spatio-temporal grabcut human segmentation for face and pose recovery, IEEE Conference on Computer Vision and Pattern Recognition (CVPR) — Workshops 2010, pp. 33–40.
[30] V. Gulshan, V. Lempitsky, A. Zisserman, Humanising grabcut: learning to segment humans using the kinect, Proceedings of the International Conference on Computer Vision (ICCV) — Workshops, 2011.
[31] J. Zhao, S.-C. Cheung, Human segmentation by geometrically fusing visible-light and thermal imageries, Multimedia Tools Appl. 1 (1) (2012) 1–29.
[32] J. Zhao, S.C. Sen-ching, Human segmentation by geometrically fusing visible-light and thermal imageries, International Conference on Computer Vision (ICCV) — Workshops 2009, pp. 1185–1192.
[33] R.H. Luke, D. Anderson, J.M. Keller, S. M., Human segmentation from video in indoor environments using fused color and texture features, Tech. rep, Electrical and Computer Engineering Department, University of Missouri, 2008.
[34] P. Viola, M. Jones, Robust real-time face detection, Int. J. Comput. Vis. 57 (2004) 137–154.
[35] T. Zhao, R. Nevatia, Bayesian human segmentation in crowded situations, Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR), vol. 2 2003, pp. 406–413.

[36] M. Muja, D.G. Lowe, Fast approximate nearest neighbors with automatic algorithm configuration, International Conference on Computer Vision Theory and Application 2009, pp. 331–340.
[37] C. Plagemann, V. Ganapathi, D. Koller, S. Thrun, Real-time identification and localization of body parts from depth images, Proceedings of the International Conference on Robotics and Automation (ICRA) 2010, pp. 3108–3113.
[38] J.R. Quinlan, Induction of decision trees, Mach. Learn. 1 (1) (1986) 81–106.
[39] A. Yali, G. Donald, Shape quantization and recognition with randomized trees, Neural Comput. 9 (1997) 1545–1588.
[40] B.A. Shepherd, An appraisal of a decision tree approach to image classification, Proceedings of International Joint Conference on Artificial Intelligence 1983, pp. 473–475.
[41] F. Moosmann, B. Triggs, F. Jurie, Fast discriminative visual codebooks using randomized clustering forests, Proceeding of Advances in Neural Information Processing Systems, 2007.
[42] V. Lepetit, P. Lagger, P. Fua, Randomized trees for real-time keypoint recognition, Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR) 2005, pp. 775–781.
[43] J. Shotton, M. Johnson, R. Cipolla, Semantic texton forests for image categorization and segmentation, Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR) 2008, pp. 1–8.
[44] Y. Boykov, O. Veksler, R. Zabih, A new algorithm for energy minimization with discontinuities, Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR) — Workshops 1999, pp. 26–29.
[45] Y. Boykov, V. Kolmogorov, An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision, IEEE Trans. Pattern Anal. Mach. Intell. 26 (9) (2004) 1124–1137.
[46] N. Papadakis, A. Bugeau, Tracking with occlusions via graph cuts, IEEE Trans. Pattern Anal. Mach. Intell. 33 (1) (2011) 144–157.
[47] Y. Boykov, M. Jolly, Interactive graph cuts for optimal boundary and region segmentation of objects in N-D images, Proceedings of the International Conference on Computer Vision (ICCV), 2001.
[48] S. Sclaroff, L. Liu, Deformable shape detection and description via model-based region grouping, IEEE Trans. Pattern Anal. Mach. Intell. 23 (5) (2001) 475–489.
[49] J. Shotton, A. Fitzgibbon, M. Cook, T. Sharp, M. Finocchio, R. Moore, A. Kipman, A. Blake, Real-time human pose recognition in parts from a single depth image, Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2011.
[50] J. Shotton, R. Girshick, A. Fitzgibbon, T. Sharp, M. Cook, M. Finocchio, R. Moore, P. Kohli, A. Criminisi, A. Kipman, A. Blake, Efficient human pose estimation from single depth images, IEEE Trans. Pattern Anal. Mach. Intell. 35 (12) (2013) 2821–2840.
[51] C.-C. Chang, C.-J. Lin, LIBSVM: a library for support vector machines, ACM Trans. Intell. Syst. Technol. 27 (2011) 1–27 (27, software available at http://www.csie.ntu.edu.tw/cjlin/libsvm).