Monocular 3D human pose estimation with a semi-supervised graph-based method

Mahdieh Abbasi
Computer Vision and Systems Laboratory
Department of Electrical Engineering and Computer Engineering
Université Laval, Québec, QC, Canada
mahdieh.abbasi.1@ulaval.ca

Hamid R. Rabiee
Digital Media Lab, AICTC Research Center
Department of Computer Engineering
Sharif University of Technology, Tehran, Iran
rabiee@sharif.edu

Christian Gagné
Computer Vision and Systems Laboratory
Department of Electrical Engineering and Computer Engineering
Université Laval, Québec, QC, Canada
christian.gagne@gel.ulaval.ca

Abstract

In this paper, a semi-supervised graph-based method for estimating 3D body pose from a sequence of silhouettes, is presented. The performance of graph-based methods is highly dependent on the quality of the constructed graph. In the case of the human pose estimation problem, the missing depth information from silhouettes intensifies the occurrence of shortcut edges within the graph. To identify and remove these shortcut edges, we measure the similarity of each pair of connected vertices through the use of sliding temporal windows. Furthermore, by exploiting the relationships between labeled and unlabeled data, the proposed method can estimate the 3D body poses, with a small set of labeled data. We evaluated the proposed method on several activities and compared the results with other recent methods. Our method significantly reduced the mean squared error, showing the positive effect of removing shortcut edges.

We consider the problem of 3D human body pose estimation from a sequence of monocular silhouettes. 3D human pose estimation is one of the imperative components in many intelligent systems such as visual surveillance. The human body is a 3D articulated object with a changing pose depending on the current activity. The silhouette refers to a solid shape and single color (usually black) image of a human body, with its edges matching the outline of the body.

In general, two main approaches exist to estimate the human pose: generative methods and discriminative methods. On one hand, generative approaches require a 3D human body model with parameters such as 3D joint angles, 3D joint positions, and 3D shapes, which constitutes the parameter space. With respect to the synthetic body model, a likelihood function, which indicates the probability of an observed image conditioned on a given 3D pose, is constructed. By determining the extreme points of the likelihood function, several probable poses (hypotheses) are given for an observed image [16]. The best hypothesis is selected by rendering the 2D images of the hypotheses, selecting the hypothesis with the image, which is most similar to the observed image [12]. Although generative approaches are able to explicitly express constraints of the human body and to estimate 3D poses of complex activities, modeling an accurate likelihood function is hard, and searching the high dimensional parameter space has a significant computational cost. Moreover, generative approaches require both a good initialization point to start the search in this space and an appropriate 3D human body model.

On the other hand, discriminative approaches directly learn a mapping from the feature space, X, into the pose space, Y, [3, 11]. Discriminative approaches do not have the issues of generative methods; however, they face a one-to-many mapping problem due to the loss of depth information from images. In other words, although two silhouettes are similar, their corresponding poses can be quite different. Hence, these two similar silhouettes can be mapped into distinct poses. In addition, since internal body edges are removed from silhouettes, this issue (called depth ambiguity challenge) is also more intense. Lastly, a considerable amount of labeled data from the entire feature space is required in order to learn an accurate mapping function [3, 17].

Our proposal consists in a discriminative approach designed to reduce the acuteness of the one-to-many mapping problem typical of these approaches. This is achieved by building a graph modeling the relations between silhouettes, using a temporal sliding windows to remove any shortcut edges of the graph that are likely to correspond to misleading associations between the silhouettes. The procedure is semi-supervised as it relies on the use of both labeled (*i.e.*, for which the real pose is known) and unlabeled data.

The remaining of the paper is organized as follows. In Sec. 1, an overview of related discriminative approaches for 3D pose estimation is presented. Follows in Sec. 2 some explanations on manifold learning and the Laplacian regularization framework. Details of the proposed method are presented in Sec. 3, and the experimental results are presented in Sec. 4. Concluding remarks are provided in Sec. 5.

1. Discriminative Approaches

A mixture of experts is often used in discriminative approaches for 3D pose estimation [3, 9, 17] by learning different regression functions for different regions (clusters) of the input space. The main aim of clustering the input space is to separate ambiguous silhouettes into different clusters and to learn a local regression function for each cluster. These clusters are obtained from labeled data, where each cluster contains similar silhouettes whose corresponding poses are similar too. The key issue of [3, 9, 17] is the requirement of a large amount of labeled data (several thousands) to correctly learn the local mappings.

Although the input and the output (pose) spaces are high dimensional, it has been shown that the high dimensional data points of a human activity lie on a low dimensional manifold [8]. In [8, 11], dimensionality of data points is explicitly reduced by using LLE [18] as a manifold learning method. Using the data manifold, the depth ambiguity challenge is resolved since two near ambiguous silhouettes, with respect to Euclidean distance, should be far away on the manifold. Therefore, instead of learning a mapping from the input space into the pose space, they learned a mapping from the manifold into the pose space. In another supervised method [7], a latent space is learned from the joint feature-pose space by GPLVM [10], a probabilistic dimension reduction method. The authors disambiguated the similar silhouettes that have dissimilar poses by considering temporal consistency among the sequential data. All of these supervised methods [7, 8, 11] explicitly reduce the dimension of data, however it is possible that the geometric relationships among data become distorted during this process.

Semi-supervised approaches, which are considering both labeled and unlabeled data, are able to estimate 3D poses even with the existence of a small set of labeled data [6, 9, 13, 14]. Kanaujia *et al.* [9] presented a semi-supervised method that extends the mixture of experts idea [3]. To accurately learn the weights of each local mapping, the authors approximated the data manifold by the k-NN (k Nearest Neighbors) method. By exploiting the data manifold as a prior knowledge, it can be determined whether two silhouettes are truly near each other. However, if the constructed graph contains shortcut edges, then the weights of each expert (*i.e.*, local mapping) are not accurately estimated. Similar to Agarwal and Triggs [3], this work still demands a large amount of labeled data.

Pourdamghani *et al.* [14] presented a semi-supervised method that neither considers a manifold as prior knowledge nor reduces explicitly the dimension of feature and pose space, rather estimating poses directly from both the input data manifold and the pose data manifold, which are approximated by k-NN. This is achieved first by estimating the poses based on the input data manifold, then constructing another graph (called pose data manifold) using the estimated poses. This graph shows a pose data manifold which is relatively close to the true manifold. Additionally, by exploiting the pose data manifold, the shortcut edges within the input data manifold are identified, and their weight is reduced. The main drawback of this work is that the shortcut edges within the input data manifold cause the poses to be inaccurately estimated, and then based on these inaccurate estimated poses, another graph is constructed with possibly some dissimilar poses still connected.

In another semi-supervised method [15], a mapping from the input space into a new space is learned from labeled data, so that the ambiguous silhouettes are moved far away. Afterwards, a graph is constructed by k-NN from data points in this new space where it is assumed that this constructed graph has no shortcut edges. However, the quality of this mapping function is highly dependent on the labeled data.

Besides the aforementioned semi-supervised graph-based methods for 3D human pose estimation, other graph-based methods, in other computer vision applications, have been proposed for learning new affinities without the need for any labeled data [20, 21]. In particular, Yang *et al.* [20]

have obtained higher order information about the relation between data points by applying the concept of a tensor product graph (TPG). However, the construction of a TPG has high time and memory complexity, such that the authors proposed an iterative approach for propagation on the original graph, and proved that this is equivalent to diffusion on TPG. So, this iterative algorithm (equivalently the diffusion of similarities on the TPG) creates new learned affinities (*i.e.*, new edge weights). The affinities are able to approximate geodesic distances between data points.

We present a semi-supervised graph-based method that estimates 3D poses of a sequence of silhouettes. As mentioned above, the depth ambiguity challenge causes the creation of shortcut edges within the constructed graph. We are proposing to use two sliding temporal windows for detection and elimination of the shortcut edges. Moreover, in order to avoid losing the geometric relationship among data, we have not explicitly reduced the dimensionality of data. Instead, we have employed the geodesic distance to approximate the real manifold. In contrast to some of the previous methods, the proposed method neither requires a large amount of labeled data since it can use both labeled and unlabeled data, nor does it require the learning of many parameters. Therefore, as we will show later, time complexity for learning and prediction phases of the proposed method are $O(n^2)$ and $O(n^3)$, respectively.

2. Learning on Manifold

It is known that many natural high dimensional data, $x_i \in \mathbb{R}^D$, such as human faces and human activities, typically lie on a low dimensional manifold $\mathcal{M} \in \mathbb{R}^d$ with $d \ll D$, which corresponds to the intrinsic structure of data x_i [5, 8]. Moreover, one could discover the relationships among data points by using the manifold structure rather than the Euclidean distance in high dimensional spaces.

In general, two data points may be considered as nearby in a high dimension space based on the Euclidean distance, but the same two data points may be faraway on the manifold, based on the geodesic distance. Geodesic distance could measure the distance of two data points as the length of the shortest path between them on the manifold $\mathcal{M}$ [5]. Learning approaches based on manifold consider two assumptions for predicting the data labels of data: 1) nearby data have the same label, and 2) labels of data points that have the same structure are similar (manifold or cluster assumption). Therefore, the labels of data that lie on a manifold should be predicted by the geodesic distance rather than the Euclidean distance.

Consider the labeling function f, where f indicates the labels ($y \in \mathbb{R}$) of data points on $\mathcal{M}$ ($f : \mathcal{M} \to \mathbb{R}$). We are proposing to use the Laplacian regularization framework [4] for learning f. Unlike [8], which explicitly reduces the dimension of input data to construct data manifold, this framework discretely models it using a graph construction algorithm [4]. The most common graph construction method is k-NN, in which each data point simply becomes connected to its k nearest neighbors [4, 9, 14].

The Laplacian regularization framework learns f based on the manifold assumption. This assumption states that the labeling function (f) should change smoothly on the manifold. Therefore, the total summation of the absolute value of the gradient of f ($|\triangledown f|$) over the manifold should be small [4]. As we model a manifold by a graph (named G), the manifold assumption can be represented for the adjacency matrix, W, of graph G as follows [4]:

$$s_G(f) = \sum_{i=1}^{l+u}\sum_{j=1}^{l+u} w_{ij}(f_i - f_j)^2, \qquad (1)$$

where l and u indicate the number of labeled and unlabeled data, respectively, f_i is the label of vertex i, and w_{ij} corresponds to the binary weight of the edge between vertices i and j of the graph G. This term expresses whether two data points (i and j) are close enough so that they share an edge ($w_{ij} = 1$) in G, or not ($w_{ij} = 0$). In such a case, we can assign similar labels to them. So, the squared difference of labels of such data points should be small ($w_{ij}(f_i - f_j)^2$). This term is the discrete form of manifold assumption by modeling a continuous manifold $\mathcal{M}$ by a graph G. The above equation can be rewritten as follows:

$$s_G(f) = \sum_{i=1}^{l+u}\sum_{j=1}^{l+u} w_{ij}(f_i - f_j)^2 = \mathbf{f}^t L \mathbf{f}, \qquad (2)$$

where $L = W - D$ is the Laplacian matrix, D is a diagonal matrix obtained from $D_{ii} = \sum_{j=1}^{l+u} w_{ij}$, and $\mathbf{f} \in \mathbb{R}^{l+u}$ is the vector of labels. Finally, the Laplacian regularization framework infers $\mathbf{f}$ (labels of data points) by minimizing the Mean Square Error (MSE) of labeled data and the regularization term ($s_G(f)$):

$$\mathbf{f}^* = \underset{\mathbf{f}}{\operatorname{argmin}} \sum_{i=1}^{l} (f_i - y_i)^2 + \gamma \mathbf{f}^t L \mathbf{f}, \qquad (3)$$

where $0 \leq \gamma \leq 1$ is the parameter that controls the influence of the regularization term. For $\gamma = 1$, MSE and manifold assumption ($s_G(f)$) play an equal role, while for $\gamma = 0$, the manifold assumption is ignored, and labels are learned based only on MSE. The above objective function is convex and its optimal solution could be obtained by setting its derivative (with respect to f) equal to zero [4]. Since the time complexity of inverting an $n \times n$ matrix is at most $O(n^3)$, the optimal solution can be obtained in $O(n^3)$ for each dimension. Finally, since each pose $\mathbf{y} \in \mathbb{R}^d$ is d-dimensional, the above objective function can be independently solved for each dimension. Therefore, this process has the time complexity of $O(n^3 d)$.

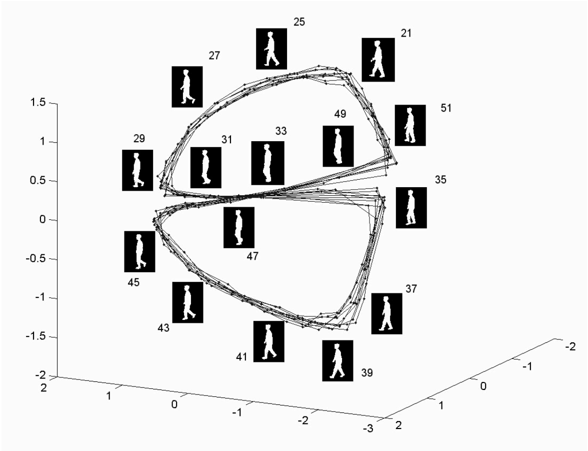


Figure 1: The embedded walking manifold [8].

3. Proposed Method

To overcome the two main challenges of 3D human pose estimation, depth ambiguity, and the lack of huge labeled data, we present a semi-supervised graph-based approach. We assume that a manifold for each activity can be approximated by a graph construction method in the input space. However, the constructed graph may contain many shortcut edges, which are connecting two points that are far away with respect to the underlying data manifold (*i.e.*, two points that have a large geodesic distance), while they are located close enough to share a common edge in the k-NN graph (*i.e.*, two points that have a small Euclidean distance). To illustrate this claim, Fig. 1 shows the manifold of a walking activity. Although the frames 25 and 41 contain two similar silhouettes, no edge connects these two silhouettes on the real embedded manifold, since their 3D poses are totally different. However, these two silhouettes might be connected as nearest neighbors in the constructed graph. Consequently, such distractive edges distort the approximated manifold, and then lead to inaccurate label estimation. By removing these shortcut edges, the approximated manifold becomes more dependable for 3D body pose estimation.

In this work, we attempt to construct more accurate, and dependable graph, not just based on similarity between each node and its neighbors in the input space, but also by seeking the similarity between temporal windows. Finally, we employ Laplacian regularization framework for learning labels.

Our work is inspired by Pourdamghani *et al.* [14] to find and remove the shortcut edges in the feature graph. However, they identified shortcut edges based on approximated spatial information in the pose space, while this approximated information is not dependable. The proposed method rather determine these destructive edges through the use of temporal information. Therefore, we verify the similarities of all the connected nodes in the feature graph with the temporal information rather than the approximated spatial information.

The sliding temporal windows are used to detect and remove the shortcut edges from the constructed graph. We assume that if two silhouettes are truly similar enough to share an edge, then their temporal neighbors obtained from the two temporal windows should be similar too. Therefore, by measuring the similarity with the sliding temporal windows, we assess whether two points connected by an edge are truly close.

Our labeled data consists of sequential frames of a given activity and unlabeled data, which is another sequence of frames from the same activity. Take $X = \{x_1, \ldots, x_n\}$ as the input data, where $n = l + u$ is the total number of data (labeled and unlabeled data). By employing the k-NN method, a graph (called $G_f(X, E)$) is constructed.

Since there is a temporal relationship between each pair of consecutive frames of a video, we can construct a temporal graph for labeled data (called $G^l_t(X^l, E^l_t)$) as follows. Each input data at time t, x^l_t, could be connected to its preceding x^l_{t-1}, and its subsequent x^l_{t+1}. Likewise, a temporal graph from unlabeled data could be constructed (called $G^u_t(X^u, E^u_t)$). Finally, these two graphs could be merged into a single graph $G_t(X, E') \in \mathbb{R}^{n \times n}$, where E' consists of the edges of graphs G^u_t and G^l_t. As mentioned earlier, the undirected weighted graph $G_f \in \mathbb{R}^{n \times n}$ is constructed by finding K nearest neighbors, N^K_i, for each data point x_i:

$$G_f^{i,j} = G_f^{j,i} = \begin{cases} 1 & x_j \in N_i^K \vee x_i \in N_j^K \\ 0 & \text{otherwise} \end{cases} \tag{4}$$

The graph G_f contains three types of edges as follows:

Type I : edges connecting data points close together in time.

Type II : edges connecting repeated similar poses when an activity is performed.

Type III : shortcut edges due to some ambiguity in the input space.

An edge is denoted as type I if its two connected vertices are reachable at most by four hops through the temporal graph G_t. We choose this number of hops as we observed that each pose becomes different from the poses that are reachable with more than four hops in G_t. To distinguish whether a specific edge G_f^{ij} is of type II or III, we consider two temporal windows of five frames, and assume that i^t and $j^{t'}$ are centered on temporal windows at time t and time t', respectively (Fig. 2).

The similarity of these temporal windows is measured by counting the number of edges that connect vertices $j^{t'+k}$

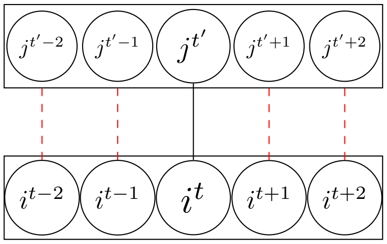


Figure 2: Two temporal windows contain temporal neighbors of vertices $j^{t'}$ and i^t. The similarity of these windows indicates whether the edge $G_f^{i,j}$ (black edge) is a shortcut.

and i^{t+k} in G_f, where $k \in [-2, 2]$ (red dashed edges in Fig. 2). Therefore, if these two temporal windows are sufficiently similar, which is indicated by the threshold θ, the edge G_f^{ij} is classified as type II. Indeed, while performing a specific activity, similar poses often occur, so it is expected these windows will be tagged as similar. Otherwise, the edge connects two ambiguous silhouettes (type III), which is indicated by dissimilar temporal windows. In this situation, the edge weight should be reduced by being set to $\alpha \in [0, 1]$. We set the value of α to 0.1, based on a 5-fold cross-validation.

The modified graph (called G_f^p) is obtained from G_f as follows:

$$G_f^{p\,(i,j)} = G_f^{p\,(j,i)} = \begin{cases} 1 & \sum_{k=-2}^{2} G_f^{(j^{t'+k}, i^{t+k})} > \theta \\ \alpha & \text{otherwise} \end{cases} \tag{5}$$

To learn the parameter θ, first the graph $g_p(Y^l, E_l)$ is constructed from labeled data by using the k-NN in the pose space, where its vertices, Y^l, are the corresponding 3D poses of the silhouettes. It is important to mention that the graph g_p is a dependable graph since this graph has no shortcut edges, and all of its connected vertices have similar labels. In addition, another graph (called $g_f(X^l, E'_l)$) is extracted from G_f, where its vertices, X^l, are the representation of the labeled data in the input space (silhouette). Similarly to G_f, this graph may have some shortcut edges. We discover the existing shortcut edges in g_f using the geodesic information that is available in g_p. When an edge is identified as a shortcut edge, the similarity of its corresponding temporal windows is measured by counting the number of edges of g_f that connect vertices within these two temporal windows (Fig. 2). This value for each shortcut edge of g_f is calculated and all of these values are accumulated in $\mathcal{S}$. In dividing $\mathcal{S}$ by the number of shortcut edges in g_f, the parameter θ is obtained. The algorithm for learning parameter θ, is shown in Algorithm 1.

To illustrate the effect of the aforementioned modifications on G_f, the proposed method is applied to a 5-NN graph that is constructed from 150 sample data points (100 samples as labeled data and 50 samples as unlabeled data) of the circular walking data set (Fig. 3). To visualize the constructed graph, we reduced the dimensionality of the corresponding pose vectors to 3 (dimensionality reduction is achieved by a kernel PCA using a Gaussian kernel). Fig. 3 depicts the red edges as the detected shortcut edges by the proposed method. In addition, in order to show the relevance of the removed edges in Fig. 3, the corresponding silhouettes to some nodes (with blue square marker), which are connected by the detected shortcut edges, are manifested.

Algorithm 1 Learning of parameter θ.

```
Require: $g_p(Y^l, E_l)$, $g_f(X^l, E'_l)$
  $\mathcal{S} = 0$                ▷ Sum of similarity from shortcut edges
  $C = 0$                    ▷ Number of shortcut edges in $g_f$
  for all $g_f^{i,j} = 1$ do
      if vertex $j$ not reachable from vertex $i$ by at most by 4 hops in graph $g_p$ then
          $\mathcal{S} \leftarrow \mathcal{S} + \sum_{k=-2}^{2} g_f^{(i^{t+k}, j^{t'+k})}$
          $C \leftarrow C + 1$
      end if
  end for
  $\theta = \mathcal{S}/C$
```

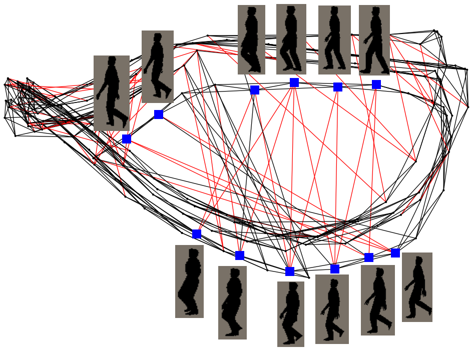


Figure 3: The 5-NN constructed graph of circular walking data. The detected shortcut edges are shown in red color. The silhouettes are correspond to the vertices with the blue square markers.

The temporal information is implicitly used for the construction of G_f^p. We can also explicitly benefit from the temporal information by exploiting the edges of G_t (called $G_f^p + G_t$). Since poses of an activity change smoothly over time, the added edges to G_f^p do not distort the manifold assumption.

4. Experimental Results

In this section, we compared the performance of the proposed method with some recent semi-supervised methods, in addition to SGPLVM [7], a generative approach that considers the temporal information. Furthermore, to investigate the impact of using the temporal graph and the removal of shortcut edges on the accuracy of pose estimation, we applied Laplacian regularization on the graphs G_f, $G_f^p + G_t$, and $G_f + G_t$. Here, we used the MSE of joint angles between the true and estimated joint angles as a measure of performance.

4.1. Datasets

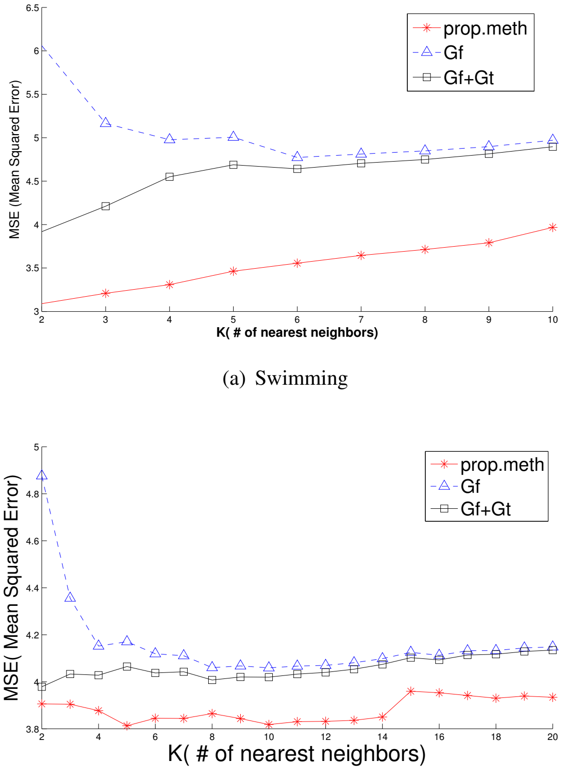
Like other authors, we considered various activities such as circular walking [2, 7], swimming, boxing, and walking forward and backward [14, 15]. The true 3D poses of these activities (except circular walking) are provided by the CMU motion capture data [1]. Moreover, the images of all activities are produced by Poser, a computer graphic package from Curious Labs. It is worthwhile to note that the 3D poses of all activities are represented by 54 dimensional vectors.

The circular walking data set of Agarwal and Triggs [2] consists of two separate sequential datasets. The first set which has 1691 frames is used as the labeled data, and the other one with 418 frames is used as the unlabeled data. To choose the labeled and unlabeled data from the swimming, boxing and walking data set [1], we have sequentially selected the labeled data from the first frame up to 20, 40, and 60 percent of the whole data, and the remainder of the data was used as the unlabeled data. It is important to note that the temporal relationship between the last frame of the labeled data, and the first frame of the unlabeled data has been removed intentionally, given that this information is not available in real world applications.

4.2. Quantitative Comparison of Graphs

As mentioned before, to show the effect of removing the shortcut edges and adding the temporal information, we applied Laplacian regularization on the graphs G_f, $G_f^p + G_t$, and $G_f + G_t$. For each activity, 60 percent of the whole data is chosen as the labeled data, and the remainder of the data set is considered as the unlabeled data.

Fig. 4 shows the MSE curves as a function of K on the walking (a) and swimming (b) datasets, where K indicates the number of nearest neighbors in k-NN. The comparison of MSE curves for G_f and $G_f + G_t$ shows that as the value of K increases, the impact of temporal edges becomes less on their MSE performance. In other words, as K increases, more irrelevant edges appear in the graphs. Hence, due to the profound negative effect of the irrelevant edges for larger values of K, the positive effect of adding temporal edges into graph G_f is decreased until the MSE of $G_f + G_t$ converges to the MSE of G_f. Moreover, the comparison of MSE curves for $G_f + G_t$ and $G_f^p + G_t$, (the proposed method) reveals the significant influence of removing shortcut edges in reducing the MSE of the proposed method. Therefore, when shortcut edges are removed and temporal edges are added to G_f, the MSE remarkably reduces.



(a) Swimming

(b) Walking

Figure 4: Mean Squared Error (MSE) of the proposed method ($G_f^p + G_t$), G_f, and $G_f + G_t$ as a function of the number of nearest neighbors on (a) swimming and (b) walking activities.

In addition, for extremely small or large values of K, the label propagation is not conducted properly. Therefore, the value of K should be chosen to be large enough to ensure sufficient connectivity between the edges while avoiding the construction of irrelevant edges. Based on our observations we choose K=5, and also let γ be 10^{-4} in all of our experiments.

4.3. Quantitative and Subjective Comparison of Methods

To see the impact of removing the shortcut edges from G_f, we compared MSE of Laplacian regularization framework on two graphs G_f, as the base graph, and $G_f^p + G_t$. In

Table 1: Mean square error comparison of the TGP, GC+RT, and G_f with the proposed method ($G_f^p + G_t$).

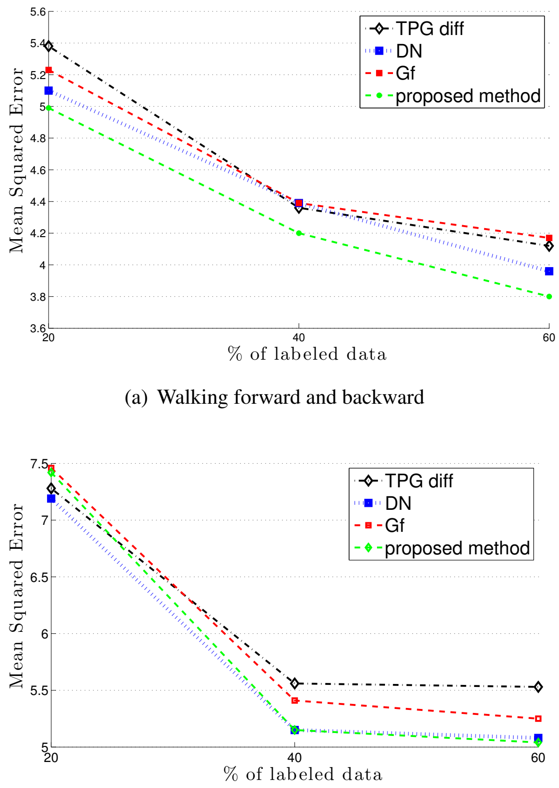
Activity (# all data)	% labeled data	TGP	GC+RT	G_f	Proposed method
Circular walking (1961)	60 %	7.28	5.30	5.25	**5.04**
	40 %	8.54	5.37	5.41	**5.15**
	20 %	21.42	7.63	7.46	**7.42**
Boxing (1400)	60 %	12.01	10.51	10.00	**9.34**
	40 %	17.91	12.04	11.69	**10.87**
	20 %	18.95	12.18	11.75	**10.96**
Swimming (1202)	60 %	5.03	4.91	4.77	**3.55**
	40 %	5.75	5.38	5.30	**4.54**
	20 %	7.10	6.67	6.65	**6.57**
Walking (1000)	60 %	4.80	4.36	4.17	**3.80**
	40 %	5.26	4.57	4.39	**4.20**
	20 %	8.59	5.65	5.23	**4.99**

addition, our method ($G_f^p + G_t$) was compared with TGP (Twin Gaussian Processes) [6], and GC+RT [14], as two recent semi-supervised methods for human pose estimation, on circular walking, swimming, boxing, and walking activities (see Table 1). As shown in the table, MSE of TGP in comparison with the other methods is considerably higher, particularly when a small number of labeled data is accessible. This shows that TGP needs a large amount of labeled data to accurately estimate 3D poses. Moreover, as Table 1 manifests, our method significantly reduced MSE in all cases of all activities since this method properly detects the shortcut edges within the base graph, G_f, and removes their destructive effect. Therefore, it can be concluded that our method estimates 3D poses more accurately than GC+RT, and the base graph G_f.

Also, in Fig. 5 we compared the proposed method with TPG diffusion [20] and Dominate Neighbor (DN) method [19], two recent state of the art methods. We used the parameters suggested in their papers and employed the Laplacian regularization on all these graphs to estimate 3D poses. In contrast to the k-NN method that simply finds k nearest neighbors for all vertices, DN is a novel method that finds dominant neighbors for each vertex, x_i, based on both pairwise similarities of x_i with other vertices, and similarities between other vertices. TPG diffusion learns new affinities by diffusing the affinities on the Tensor Product Graph (TPG).

Fig. 5 reveals that DN outperforms G_f since it finds better nearest neighbors than G_f. The TPG diffusion method, which iteratively diffuses similarities on graph G_f, may intensify the negative effect of the shortcut edges in G_f. As a result, the existence of the shortcut edges may effect the performance of the TPG diffusion method. For circular walking, DN competes with our method in performance.

Furthermore, we compared our method with a genera-



(a) Walking forward and backward

(b) Circular walking

Figure 5: Mean square error of the proposed method, TPG diffusion, and dominant neighbor (DN) as a function of percentage of labeled data on two different datasets walking (a) walking forward and backward (b) circular walking sequences.

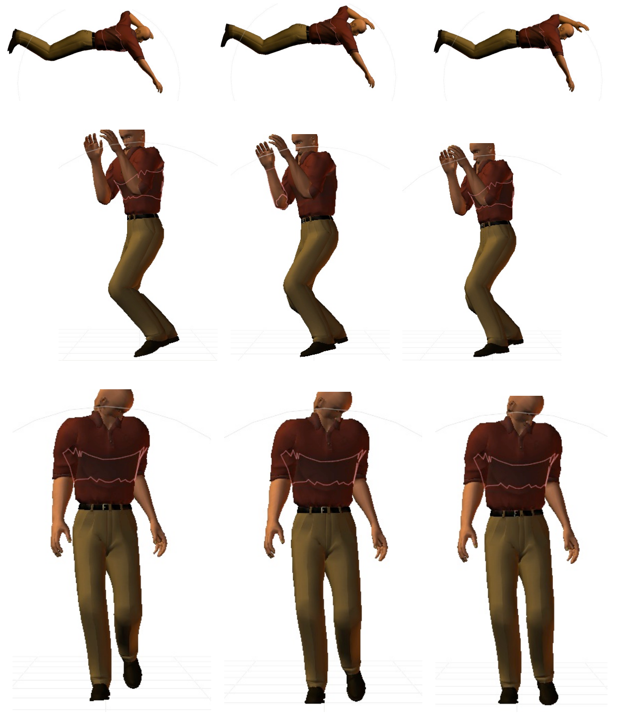


Figure 6: Subjective comparison: The first column shows the ground truth for each activity, the second and third columns show the results of Laplacian regularization on the modified graph $G_f^p + G_t$, and G_f, respectively.

tive approach, SGPLVM [7], on the circular walking activity. SPGLVM explicitly reduces the dimensionality of data points, and used temporal information to disambiguate some ambiguous silhouettes. However, the dimension reduction may misrepresent the real geometric relationships among data points. The MSE of SGPLVM on the circular walking (as reported by its authors) is 5.3, while its time complexity for learning this model's parameters and inference are $O(n^3)$ and $O(n^4)$, respectively. Using the same labeled and unlabeled data (data set), the MSE of our proposed method is 4.97, while its time complexity for learning and inference are $O(n^2)$ and $O(n^3)$, respectively. In comparison with SGPLVM, our method is not only faster at completing the learning and the prediction phases, but it also accurately estimates 3D poses of the unlabeled data.

Fig. 6 illustrates the subjective comparison of the ground truth (first column), the proposed method (second column), and the original graph G_f (third column). The first row, second row, and third row present the swimming, boxing, and walking activities, respectively. These subjective results were obtained from 60 % labeled, and 40 % unlabeled data for each activity.

5. Conclusions

In this paper, we introduced a new semi-supervised graph based method for 3D human pose estimation from a sequence of silhouettes. The proposed method takes advantage of the relationships between the labeled and unlabeled data, thereby eliminating the requirement for a large number of labeled data. Furthermore, it directly estimates the labels of data points from the approximated manifold that can be obtained from a graph construction method (*i.e.*, k-NN). Moreover, to identify and remove the shortcut edges from this graph, we employed a temporal window scheme, and compared the similarity between each pair of temporal windows. Finally, we evaluated our method on a range of activities such as walking, boxing, and swimming. Based on the experimental results, our method depicted the positive effect of removing shortcut edges from the base graphs. Additionally, for all activities, our method estimated 3D poses more accurately than TGP, GC+RT, and TPG diffusion.

Acknowledgments

This work was funded through grants from NSERC (Canada). We thank Annette Schwerdtfeger for proofreading this manuscript. The first author (Abbasi) would like to thank the Digital Media Lab, AICTC Research Center (Sharif University of Technology) for hosting her during her Master's.

References

[1] CMU motion captured dataset. `http://mocap.cs.cmu.edu/`, 2012. [Online; accessed 2012].

[2] A. Agarwal and B. Triggs. 3D human pose from silhouettes by relevance vector regression. In *In Conf. on Computer Vision and Pattern Recognition (CVPR '04)*, volume 2, pages II–882. IEEE, 2004.

[3] A. Agarwal and B. Triggs. Monocular human motion capture with a mixture of regressors. In *Conf. on Computer Vision and Pattern Recognition (CVPR '05) Workshops*, pages 72–72. IEEE, 2005.

[4] M. Belkin. *Problems of learning on manifolds*. PhD thesis, The University of Chicago, 2003.

[5] M. Belkin and P. Niyogi. Semi-supervised learning on Riemannian manifolds. *J. Mach. Learn.*, 56(1):209–239, 2004.

[6] L. Bo and C. Sminchisescu. Twin Gaussian processes for structured prediction. *Int. J. Computer Vision (IJCV)*, 87(1):28–52, 2010.

[7] C. H. Ek, P. H. Torr, and N. D. Lawrence. Gaussian process latent variable models for human pose estimation. In *Mach. learning for multimodal interaction*, pages 132–143. Springer, 2008.

[8] A. Elgammal and C. Lee. Inferring 3D body pose from silhouettes using activity manifold learning. In *Conf. on Computer Vision and Pattern Recognition (CVPR '04)*, volume 2. IEEE, 2004.

[9] A. Kanaujia, C. Sminchisescu, and D. Metaxas. Semi-supervised hierarchical models for 3D human pose reconstruction. In *Conf. on Computer Vision and Pattern Recognition (CVPR '07)*, pages 1–8. IEEE, 2007.

[10] N. Lawrence. Gaussian process latent variable models for visualisation of high dimensional data. *Adv. Neural Inf. Process. Syst. (NIPS '04)*, 16:329–336, 2004.

[11] C. Lee and A. Elgammal. Modeling view and posture manifolds for tracking. In *Int. Conf. on Computer Vision (ICCV 2007)*, pages 1–8. IEEE, 2007.

[12] M. W. Lee and I. Cohen. A model-based approach for estimating human 3D poses in static images. *IEEE Trans. on Pattern Analysis and Machine Intelligence,*, 28(6):905–916, 2006.

[13] R. Navaratnam, A. Fitzgibbon, and R. Cipolla. The joint manifold model for semi-supervised multi-valued regression. In *Int. Conf. on Computer Vision (ICCV 2007)*, pages 1–8. IEEE, 2007.

[14] N. Pourdamghani, H. R. Rabiee, F. Faghri, and M. H. Rohban. Graph based semi-supervised human pose estimation: When the output space comes to help. *Pattern. Recogn. Lett*, 33(12):1529–1535, 2012.

[15] N. Pourdamghani, H. R. Rabiee, and M. Zolfaghari. Metric learning for graph based semi-supervised human pose estimation. In *Int. Conf. on Pattern Recognition (ICPR 2012)*, pages 3386–3389. IEEE, 2012.

[16] D. Ramanan, D. A. Forsyth, and A. Zisserman. Tracking people by learning their appearance. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 29(1):65–81, 2007.

[17] R. Rosales and S. Sclaroff. Learning body pose via specialized maps. *Adv. Neural Inf. Process. Syst. (NIPS '98)*, 2:1263–1270, 2002.

[18] S. Roweis and L. Saul. Nonlinear dimensionality reduction by locally linear embedding. *Science*, 290(5500):2323–2326, 2000.

[19] X. Yang and L. J. Latecki. Affinity learning on a tensor product graph with applications to shape and image retrieval. In *Conf. on Computer Vision and Pattern Recognition (CVPR '11)*, pages 2369–2376. IEEE, 2011.

[20] X. Yang, L. Prasad, and L. J. Latecki. Affinity learning with diffusion on tensor product graph. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 35(1):28–38, 2013.

[21] Y. Zhou, X. Bai, W. Liu, and L. J. Latecki. Fusion with diffusion for robust visual tracking. In *Adv. in Neural Inf. Process. Syst. (NIPS '12)*, pages 2978–2986, 2012.