Employing Particle Filters on Riemannian Manifolds for Online Domain-Shift Object Learning and Occlusion Handling

Irene Yu-Hua Gu  Zulfiqar H. Khan
Department of Signals and Systems, Chalmers University of Technology
Gothenburg, 41296, Sweden
irenegu@chalmers.se  zulfiqar@alumni.chalmers.se

ABSTRACT
Visual object tracking from single cameras is often employed as the basic block in a multi-camera tracking environment, and its performance naturally affects the multi-camera tracking system. Online learning of object model is essential for mitigating the tracking drift for highly dynamic video objects. This paper describes a domain-shift online learning and geodesic-based occlusion handling method for enhancing the robustness of manifold object tracking, especially when a large-size object (relative to an image-size) contains significant out-of-plane changes along with some long-term partial occlusion. The main contributions of the domain-shift online learning method include: (a) Utilizing a particle filter on the manifold for online learning; (b) Bayesian formulation on the manifold, for posterior state estimation on the manifold based on nonlinear state space modeling; (c) A geodesic-based method for occlusion handling on the manifold, for preventing learning occluding objects/clutter. The online learning method uses covariance matrices of manifold candidate objects (or, particles) at each time instant rather than from a sliding-window of objects in the conventional case, hence possibility of fast online learning. The proposed method has been applied to Riemannian manifold tracking of video objects that contain large-size objects with significant out-of-plane changes accompanied with long-term partial occlusions. The method is tested, compared and evaluated on a range of videos, results have provided strong support to the robustness of the proposed method. Discussions on computational issue and application scenario to multi-camera environment are also included.

Keywords
Online domain-shift learning, geodesic-based occlusion handling, manifold particle filters, manifold Bayesian learning, Riemannian manifold, domain-shift object tracking

1. INTRODUCTION
Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from Permissions@acm.org.

Motivation for domain-shift online learning
Why use manifolds? This is a natural question before a manifold-based method is selected. One main reason is that many video objects contain out-of-plane changes. Such a dynamic object is shown to reside in nonlinear spaces (e.g., manifolds), rather than in a single vector space. Another reason of using manifold is that it is a low dimensional representation of object/signal. The key difference lies in whether definition of object basis matrices is on a manifold, or in a single vector space: the former definition leads to domain-shift object description and hence more robust domain-shift learning and tracking. Another reason is that manifolds offer low-dimensional object representation while maintaining the geometry, topology and other essential property of the object, see [9] for more details on the basics of manifolds.

Main novelties
Formulate particle filters (PFs) on the manifold: PF theory is well established in a vector space, however not on nonlin-
ear smooth manifolds. To the best of our knowledge, ours is the first successful work that applies PFs on manifolds. This leads to form the covariance matrix by using candidate objects (or particles) at the same time (rather than a time-window of objects [6]), hence fast online learning.

Propose an occlusion handling method on the manifold: a geodesic-based occlusion handling criterion is applied on the manifold during the online learning to estimate whether changes are likely caused by the target object or occluding objects/clutter. The method is particularly attractive for tracking objects that could experience long-term partial occlusion. To the best of our knowledge, ours is the first successful long-term partial occlusion handling method on the manifold.

The remainder of the paper is organized as follows. Section 2 gives a brief review of the useful formulae on Riemannian manifolds. Section 3 briefly describes the big picture of domain-shift tracking scheme that contains the process of domain-shift online learning and occlusion handling blocks. Section 4 describes the domain-shift online learning and occlusion handling in details. Test results and evaluation are included in Section 5, where discussions on computation and possible multi-camera application are also included. Finally, conclusion is given in Section 6.

2. RIEMANNIAN MANIFOLDS: REVIEWS

This section summarizes several most relevant formulae under the log-Euclidean metric on Riemannian manifolds for the sake of mathematical convenience in subsequent sections. More details are referred to [9].

Exponential mapping function \((T \rightarrow M)\): Given a starting point \(p\) (or, \(p(t=0)\)) on the manifold \(M\), and the corresponding tangent vector \(\Delta\) in its tangent plane \(T\), (1) maps the tangent vector along the geodesic to yield the end point \(q\) at a unit time on \(M\), i.e. \(q = p(1)\). The exponential mapping under the log-Euclidean metric is given by:

\[
\exp_p(\Delta) = \exp(\log p + \Delta) \tag{1}
\]

Logarithmic mapping function \((M \rightarrow T)\): Given two points \(p, q\) on \(M\), (2) results in a velocity vector \(\Delta\) in \(T\) corresponding to the geodesic from \(p\) to \(q\) on \(M\). The logarithmic map under the log-Euclidean metric is:

\[
\log_p q = \log q - \log p = \Delta \tag{2}
\]

Geodesic: Given two points \(p, q\) on \(M\), the geodesic under the log-Euclidean metric is given by,

\[
d(p, q) = \|\log_p q\|_2 = \|\log q - \log p\|_2 \tag{3}
\]

Riemannian mean: The Riemannian mean for \(\text{Symm}_n^+\) matrices under the log-Euclidean metric is calculated by,

\[
E_{\text{LE}}(p_1, \ldots, p_N) = \exp_1 \left( \frac{1}{N} \sum_{i=1}^{N} \log_1 p_i \right) \tag{4}
\]

where \(I\) is the identity matrix, with a same dimension as \(p_i\). The Riemannian mean (or, expected mean on a Riemannian manifold) is an extrinsic mean. The intrinsic mean on the manifold is the Karcher mean (or, Fréchet mean).

3. SYSTEM DESCRIPTION: BIG PICTURE

This section gives the big picture of the domain-shift tracking scheme, where the domain-shift online learning and occlusion handling process (the bottom block) described in this paper is integrated into a domain-shift tracking process (the top block), as shown in Figure 1. This paper focuses on describing domain-shift online learning and partial-occlusion handling, aimed at maintaining a timely reference object model, hence mitigating tracking drift.

4. DOMAIN-SHIFT ONLINE LEARNING

Domain-shift online learning combined with occlusion handling on smooth manifolds is desirable when dynamic objects contain non-planar, or out-of-plane motion. The method is based on a nonlinear dynamic model, Bayesian formulation, and a particle filter realization, and a geodesic-based occlusion handling on the Riemannian manifold. Our rationales behind the method are: by using a set of candidate objects (particles) at the same time instead of conventionally a time-window of objects, fast online learning of out-of-plane moving objects may be achieved; by employing a particle filter (PF) on the manifold, non-linear non-Gaussian distributed pdf on the manifold may be estimated; by including both the position and velocity of manifold point in the dynamic model, the accuracy of estimated manifold point may be improved; by using a manifold allows effective learning of object that often shifts its appearance domain, especially when significant out-of-plane pose changes occur; finally, by using a geodesic distance-based occlusion handling, online learning of occluded target object is avoided.

4.1 Nonlinear State Space Model on Manifolds

Let the object appearance at \(t\) be described by a covariance matrix \(C_t\), (i.e., a point on a Riemannian manifold) and the appearance change, or speed be \(\Delta_t\). We define the state vector as follows:

\[
s_t = [C_t \Delta_t]^T \tag{5}
\]

Let the state (i.e., object appearance) dynamics be described by the following nonlinear model,

\[
\begin{align*}
C_t &= h(C_{t-1}, \Delta_t) = \exp_{C_{t-1}}(\Delta_t) \\
\Delta_t &= \Delta_{t-1} + V_t
\end{align*} \tag{6}
\]

where \(h(\cdot)\) is a nonlinear mapping function, \(\exp_{C_{t-1}}\) is the exponential mapping with the starting point at \(C_{t-1}\), \(V_t\) is assumed to be zero-mean white, \(\Delta_{t-1}\) is constant in each
sample interval $T = t_k - t_{k-1}$, and $T = 1$ is set for mathematical convenience. We refer (6) as the nonlinear model, noting the two state space equations are defined in two different but interrelated spaces (manifold points and their associated tangent planes for two different components). The first equation in (6) is the nonlinear dynamic appearance model on the manifold where two manifold points at successive time instants are related by $\Delta t$, in the tangent plane. The second equation in (6) is a constant velocity model in the tangent plane where the acceleration is considered as white noise. The model can be considered as a 2nd-order discrete white noise acceleration model for Riemannian manifold points.

4.2 Bayesian Manifold Appearance Learning

The aim here is to perform domain-shift online learning of new state $C_t$ through Bayesian estimation of reference object model on the Riemannian manifold. This can be roughly described as predicting a set of candidates on the manifold, computing their weights by estimating the likelihood, and the posterior estimation of object appearance by combining priors and weighted predicted candidates in the MMSE sense on the manifold. This is realized with the help of a PF on the manifold. The following notations are used in the paper for different types of covariance matrices: $C_{t-1}$ is for the $j$th predicted candidate object at $t - 1$; $C_t^{obj}$ is for the posterior reference object at $t$ before occlusion handling; $C_t^{obj}$ is for the selected reference object at $t$, after occlusion handling; $C_t^{obj}$ is for the new observation (or tracked object) from the tracking process at $t$. Let the current observation at $t$ be $Z_t = C_t^{obj}$ that is provided by the tracking process, where $C_t^{obj}$ is the covariance matrix of tracked object at $t$. Let $C_t$, the first component in the state vector of (5), be estimated. The recursive Bayesian estimation under the 1st-order Markov assumption becomes,

$$p(C_t|Z_{0:t}) \propto p(Z_t|C_t) \int p(C_t|C_{t-1}, \Delta t) p(C_{t-1}|Z_{0:t-1}) dC_{t-1}$$

(7)

where $C_t$ is the state vector of object manifold appearance at time $t$, $Z_{0:t}$ is the observations up to $t$. Using the weighted sum of particles drawn from the proposal distribution, the posterior pdf estimate is approximated as:

$$p(C_t|Z_{0:t}) \approx \sum_{j=1}^{N_1} w_j(t) \delta(C_t - C_t^j)$$

(8)

where $C_t^j$ is the particle, $w_j(t)$ is the weight, $\sum_{j=1}^{N_1} w_j(t) = 1$, $N_1$ is the total number of particles. Since this PF is employed on the manifold where the nonlinear model describes state variables in two inter-connected spaces, the realization of this PF requires the interaction between manifold points and tangent planes. The following main steps are included: Prediction: Let $C_{t-1}$ be a manifold particle point at $t-1$ and $\Delta t_{t-1}$ be the velocity particle that connects $(C_{t-2}, C_{t-1})$, where $C_{t-1}$ is the end point of the geodesic starting from $C_{t-2}$. First, a set of velocity particles $\Delta t^j$ (originated from $C_{t-1}^j$) is generated in tangent planes using the previous velocity particles $\Delta t_{t-1}$ according to $\Delta t^j = \Delta t_{t-1}^j + V_1$ (see (6)), $j = 1, \cdots, N_1$. Then, a set of new manifold particles $C_t^j$ is obtained from $\Delta t^j$ through the exponential mapping, according to $C_t^j = \exp_{C_{t-1}^j}(\Delta t^j)$ (see (6)). These manifold points $C_t^j$ are considered as the predicted points at $t$. Compute likelihood and update particle weights: The likelihood is computed from the Gaussian-distributed geodesic between the observation $\tilde{C}_{t}^{obj}$ and predicted $C_t^j$, $p(C_t^{obj}|C_t^j) = \exp \left\{ -\frac{1}{\sigma_t^2} \right\}$

(9)

where $\sigma_t^2$ is measurement noise (empirical value $\sigma_t^2 = 0.1$),

$$d(C_t^{obj}, C_t^j) = \left\| \log(C_t^{obj}) - C_t^j \right\|_2$$

(10)

and subsequently normalized by $w_j(t) = w_j(t) / \sum_j w_j(t)$. Resampling is applied if $N_{res} = 1 / \sum_{j=1}^{N_1} (w_j(t))^2 < 1$ then, where $N_{res}$ is a threshold, to prevent the degeneration.

Estimate posterior manifold points: The MMSE estimate of object appearance $\hat{C}_t^{obj}$ is obtained as the expected value of weighted predicted particles on the manifold by:

$$\hat{C}_t^{obj} = \exp_t \left( \sum_{j=1}^{N_1} w_j(t) \log C_t^j \right)$$

(11)

Figure 2 shows the relationship of manifold particles, velocity particles, observation point, and the predicted and posterior estimated points between two time instants.

4.3 Geodesic-based Occlusion Handling

A geodesic-based manifold occlusion handling method is used to tackle a potential problem in online learning when the target object is partially occluded. The difficulty in handling occlusions of objects with large out-of-plane changes is caused by the ambiguity between the changes caused by the dynamic object itself (e.g. pose, deformation), while avoiding updating when changes are due to partial occlusions as this could lead to subsequent tracking drift. The occlusion handling strategy is based on the observation that relatively large differences (or less cross-correlation) between the tracked region and the reference object may occur when the object experiences occlusions as compared with object pose changes. We employ the geodesic distance between the reference object at (t-1) and the current online learned object before occlusion handling as the measure,

$$d_t(C_{t-1}^{obj}, \hat{C}_t^{obj}) = \left\| \log(C_{t-1}^{obj}) - \hat{C}_t^{obj} \right\|_2$$

(12)
The new reference object $C_{t}^{obj}$ is updated according to whether the geodesic distance in (12) is smaller than a threshold $d_{th}$,

$$
C_{t}^{obj} = \begin{cases} 
C_{t-1}^{obj} & \text{if } d_{t} \leq d_{th} \\
C_{t}^{obj} & \text{otherwise (i.e., no updating)}
\end{cases} 
$$

(13)

The rationale behind the choice is based on the observation that, the geodesic from an occluding object/clutter to the reference object is usually larger than that of a moving object to its own reference object, noting that the geodesic distance used in (12) in the online learning; b) requiring much less computation as compared with using the BC.

5. EXPERIMENTS AND RESULTS

The proposed tracker is tested on several videos for tracking and online learning of dynamic objects with significant out-of-plane (or, non-planar) changes, especially when the size of objects is relatively large as compared to the size of images. Tests videos are obtained from a moving (or static) camera either in visual band or thermal infrared (IR) band.

**Setup:** In our tests, each object bounding box size is normalized to $32 \times 32$ pixels, then partitioned into $L=16$ rectangular sub-regions. For each pixel $(x, y)$ in the $j$th sub-region $R_j$, a feature vector $v^j(x,y) = [x \ y \ f_1 \ f_2 \ \ldots \ f_8]^T$ is used, where $I$ is the image pixel intensity, $I^j_k$, $k=1, \ldots, 16$, is the real outputs of Gabor filters at orientations $\Delta \theta_n = 45^\circ$, $n=0, \ldots, 7$ and frequencies $f_i = 1/3, 1/6, 1/12, 1/24$, with isotropic spread $\sigma = 0.5 f_i$. For PFS, $N_1 = 400$, $\sigma_{s_i}^2 = 0.001$, $\sigma_r^2 = 0.1$, $N_{1,t}=50$ are used for PF-1, and $N_2 = 600$, $\sigma_{s_i}^2 = 0.25$, and $N_{2,t}=75$ are set for PF-2. Similar to most PF tracking methods, noise variances of affine bounding box are empirically determined. The initial bounding box of target object is manually selected. The vector form of covariance matrix $\text{vec}(\log(C)) = [\text{vec}(\log(C^1)) \ \ldots \ \text{vec}(\log(C^L))]^T$ is formed by concatenating $L$ sub-covariance matrices from partitioned sub-regions, same as in [17].

**Implement Riemannian manifold tracking process:** This process (top block of Figure 1) is similar to the key spirit to PF tracking in a vector space [5], where the key property is that the embedded appearance in the likelihood function is on the manifold, $p(z_{2:t} | s_{2:t}) \propto \exp \left\{ -\frac{d(C_{t}^{obj}, C_t^i)^2}{\sigma_{s}^2} \right\}$, where

$$
d(C_{t-1}^{obj}, C_t^i) = \left| \left| \log(C_{t-1}^{obj}) - C_t^i \right| \right|^2 
$$

is the geodesic on the manifold, $C_t^i$ is the $i$th particle of PF-2, and $s_{2:t}$ is the state vector of PF-2 containing 6 affine parameters of a bounding box. Tracked object is obtained from MAP estimated $s_{2:t}^{\text{opt}} = s_{2:t}^{i*}$ from the particle weights, such that $i^* = \text{argmax}_{i} (w_{2:t}^{i})$.

**Two evaluation criteria:**

1) **Structural Similarity (SSIM)** is defined between object images within tracked box and the GT (Ground Truth) box,

$$
SSIM(X,Y) = \frac{(2\mu_X\mu_Y + c_1)(2\sigma_{XY} + c_2)}{\mu_X^2 + \mu_Y^2 + \sigma_X^2 + \sigma_Y^2 + c_2 + c_2} 
$$

(14)

where $X,Y$ are two images within the tracked boxes, $\mu_X, \mu_Y$ are mean values, $\sigma_X^2, \sigma_Y^2$ are variances, $\sigma_{XY}$ is the covariance, $c_1 = (k_1L)^2$ and $c_2 = (k_2L)^2$ are small constants stabilizing the division with a very small denominator value ($k_1 = 0.01$, $k_2 = 0.03$, $L = 255$). $SSIM_{avg}$ is then obtained by averaging the SSIM over all frames in each video.

2) **Average Tracking Accuracy (ATA)** is defined as the ratio of the spatial intersection and union of GT object and the tracked object over all frames, and is the same as the PETS2009 evaluation criterion (see details in [18]).

$$
ATA = \frac{\text{STDA}}{\text{STD}} = \frac{\sum_{t=1}^{N_{\text{frames}}} |G_t^i \cap D_t^i|}{|G_t^i \cup D_t^i|} 
$$

(19)

**Results and Comparisons:** All test videos contain dynamic objects with significant out-of-plane object changes. The proposed domain-shift tracking scheme (shown in Figure 1), where domain-shift online learning and occlusion handling process is embedded, is tested. Comparisons are also made with several existing trackers, including: covariance-based tracker in [6], tracker using probabilistic model on Riemannian manifold in [13], CONDENSATION-based PF tracker in [11], Grassmann manifold-based tracker in [12], multiple instance learning (MIL-based) tracker in [16], Online AdaBoost (OAB-based) tracker in [15], FragTrack in [14], and PF-based tracker in a single vector space (PF-VS). Figure 3 shows the results obtained from the proposed tracker and comparisons with the above existing trackers on 5 videos. From the visual inspection and comparison of tracked videos and Figure 3, one can see that the proposed tracker is very robust with tighter and more accurately tracked boxes, as compared to these existing trackers on the test videos.

**Online learning and partial occlusion handling:** Tests were performed on videos containing large-size objects with significant non-planar changes accompanied with long-term partial occlusions. Test videos were semi-synthetically generated by adding a moving book image to the left then top side to partially occlude a human face video. Figure 5 shows the evaluation results for the proposed tracker with/without online learning and with/without occlusion handling, and Figure 4 show tracking results on 2 such videos. Observing the tracked video results and Figure 4, the proposed tracker is shown to be very robust in tracking under such scenarios. Further, Figure 5 has provided further support to the effectiveness of the proposed domain-shift online learning and occlusion handling method.
Figure 3: Tracking results from the proposed scheme and several existing trackers. Rows 1-2: from the proposed scheme (red solid line) and trackers in [6] (green dotted-line) and in [13] (yellow long dashed line) on videos "Chia" and "IR face1"; Rows 4,6: from the proposed scheme (red) in Rows 3,5, and from the trackers in [16] (Magenta solid-line), [15] (Green dash-line) and [14] (Cyan dash line) on videos "David" and "Girl" (where 6 frames are taken from the figure in [16]); Rows 7: from the proposed scheme (red), and from the corresponding tracker in a single vector space (blue) on video "Danni".

Figure 5: Evaluation: with/without occlusion handling (left) and with/without online learning (right). Left: on video "Danni-with-occlusion"; Right: on video "Danni". The Euclidean distance between the 4 corners of tracked and ground truth boxes.

**Evaluation:** To evaluate the performance, the criteria ATA and SSIM are applied to the proposed tracker, and several existing trackers on 6 videos. Tables 1 and 2 show the results. Comparing the results in these 2 tables, the proposed scheme clearly shows better performance on these videos.

**Discussions:**

a) Application scenarios in multi-camera tracking environment: To apply the proposed (single-camera based) scheme to tracking in smart environments containing multi-camera views, one could consider to use this scheme to replace some existing building blocks of single camera tracking in the multi-view environment, e.g. in [3], so that posterior reference object (instead of prior, or previously tracked result) may be used to update the reference object in each view, before providing to estimating multi-view reference object.  

b) Computational issue: One main barrier for manifold-based tracking and online learning is its relatively high computation. This is due to the exponential and logarithmic operators required for mapping between the points on the manifold and the velocity vectors in tangent planes. Table 3 summarizes the extra computations required by using domain-shift online learning, occlusion handling and tracking, as compared to PF tracking in a single vector space. These computations in Table 3 can be described as: a) For domain-shift online learning: $N_1$ times ($N_1$ is the number

Table 1: Evaluation based on the average tracking accuracy (ATA) in (15) between tracked objects and their corresponding (manually marked) ground truths on 6 videos, obtained from the proposed tracker and several existing trackers. The range of ATA is $[0.0, 1.0]$. The larger the value the better the performance.

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Behzad</td>
<td>.855</td>
<td>.745</td>
<td>.684</td>
<td>.762</td>
<td>.846</td>
<td>.778</td>
</tr>
<tr>
<td>Chia</td>
<td>.801</td>
<td>.711</td>
<td>.678</td>
<td>.732</td>
<td>.789</td>
<td>.747</td>
</tr>
<tr>
<td>Danni</td>
<td>.836</td>
<td>.739</td>
<td>.758</td>
<td>.785</td>
<td>.827</td>
<td>.769</td>
</tr>
<tr>
<td>Dudek</td>
<td>.798</td>
<td>.685</td>
<td>.741</td>
<td>.758</td>
<td>.813</td>
<td>.745</td>
</tr>
<tr>
<td>IR face1</td>
<td>.892</td>
<td>.805</td>
<td>.811</td>
<td>.821</td>
<td>.875</td>
<td>.809</td>
</tr>
<tr>
<td>IR face2</td>
<td>.846</td>
<td>.801</td>
<td>.795</td>
<td>.782</td>
<td>.838</td>
<td>.756</td>
</tr>
</tbody>
</table>
of particles in PF-1) of \( \exp(\cdot) \) for mapping tangent vectors \( \Delta_t^i \) to manifold particles \( C_t^i \) in (6), and \( N_1 \) times of \( \log(\cdot) \) for computing the geodesic distance in the likelihood (or, particle weight) in (9), and \( N_1 \) times of \( \log(\cdot) \) for computing the weighted Riemannian mean in (11); b) For geodesic-based occlusion handling: extra computation is negligible (only one extra \( \log(\cdot) \) is needed); c) For domain-shift tracking: \( N_2 \) times \( (N_2 \) is the number of particles in PF-2) of \( \exp(\cdot) \) for computing the geodesic distance in the likelihood (or, particle weight) in PF-2. Hence, improved tracking robustness by applying Bayesian online learning on the manifold through using a manifold PF, is at the cost of increased computations, largely due to the computation of logarithmic and exponential operators. The domain-shift online learning process is shown to use approximately 85% of the total computations in the domain-shift tracking scheme [8]. Currently, using our Matlab programs without code optimization require approximately one minute per video frame for tracking and online learning. Currently, the method is not suitable for real time applications. One should choose such domain-shift online learning and tracking scheme on videos containing challenging scenarios as mentioned in the introduction section. Further improvement should be made by utilizing dedicated FPGA to reducing the processing time.

### 6. CONCLUSION

The proposed tracking scheme using Riemannian manifold-based Bayesian online learning and geodesic-based partial occlusion handling is shown to be effective and very robust for tracking visual and infrared objects. Employing particle filters on the manifold enables the realization of manifold-based Bayesian estimation of reference object appearance under a nonlinear dynamic model. Using a set of particles at each time instant as candidate object appearances on the manifold to form region-based object covariance matrix enables is effective for fast learning of dynamic objects and for mitigating tracking drift, leading to better tracking results. The proposed scheme has visibly improved the performance in terms of reducing tracking drift and tightly tracked object bounding boxes, especially in scenarios when deformable objects are large in size as comparing with the image size, and experience significant non-planar changes. Comparisons and performance evaluations with several existing trackers have provided further support to the robustness of the proposed scheme. Discussions and suggestions on the applications of proposed scheme in multi-view environment as well as computational issues are given.

### 7. REFERENCES


