ISOMAP TRACKING WITH PARTICLE FILTERING

Nikhil Rane

Dept. of Electrical and
Computer Engineering
Clemson University
Clemson, SC 29634
nrane@clemson.edu

Stan Birchfield

Dept. of Electrical and
Computer Engineering
Clemson University
Clemson, SC 29634
stb@clemson.edu

ABSTRACT

The problem of tracking involves challenges like in-plane and out-of-plane rotations, scaling, variations in ambient light and occlusions. In this paper we look at the problem of tracking a person's head and also estimating its pose in each frame. Robust tracking can be achieved by reducing the dimensionality of high-dimensional training data and using the recovered low-dimensional structure to estimate the state of an object at every time-step with recursive Bayesian filtering. Isometric feature mapping, also known as Isomap, provides an unsupervised framework to find the true degrees of freedom in high-dimensional input data like a person's head with varying poses. After the data has been reduced to lower dimensions a particle filter can be used to track and at the same time approximate the pose of a person's head in any image sequence. Isomap tracking with particle filtering is capable of handling rapid translation and out-of-plane rotation of a person's head with a relatively small amount of training data. The performance of the tracker is demonstrated on an image sequence with a person's head undergoing translation and out-of-plane rotation.

Index Terms— Head tracking, Isomap, Particle filter, 3D pose estimation

1. INTRODUCTION

Although $N \times N$ images of an arbitrary object are of dimension N^2, sequential images of a semi-rigid object, such as a person's head, are of significantly lower dimension. Variations in the image of a head are due to a small number of factors such as translation, rotation, and scaling, as well as lighting conditions, occlusions, or facial expression. Reducing the dimensionality is crucial to capture the meaningful axes in the data space in order to effectively estimate the location and pose of the object.

In this work we focus upon tracking a person's head using six independent degrees of freedom corresponding to rotation and translation in three dimensions. In our approach we start with a training data set consisting of grayscale images of a person's head at various up-down poses and left-right poses. To find the independent degrees of freedom, we utilize a dimensionality reduction technique. There are different dimensionality reduction techniques like principal component analysis (PCA), multidimensional scaling (MDS) [1], isometric feature mapping (Isomap) [2] and locally linear embedding (LLE) [3]. PCA finds the low dimensional embedding that preserves the variance in the data while MDS finds an embedding that preserves the inter-point distances in the data. Both fail to discover the true degrees of freedom in datasets which contain non-linear structures as shown by Tenenbaum et al. [2]. Isomap is an improvement over MDS because it first computes the geodesic distances between all pairs of points in higher dimensional space and then applies MDS, yielding much better results [2].

Tracking the state of a system can be achieved in the framework of recursive Bayesian filtering. In this approach one attempts to construct the posterior probability density function (pdf) of the state of a system based on a system model, which describes the dynamic evolution of state with time, and a measurement model, which relates the noisy measurements to the state of the system. The three main steps in recursive Bayesian filtering are prediction (using a system model), measurement (using a measurement model), and update based on the measurements. The pdf obtained at the current step in time is used as a prior for the next step. The broadly used methods under the Bayesian framework are Kalman filters [4], extended Kalman filters and particle filters [5]. A Kalman filter uses linear system models while the extended Kalman filter approximates non-linear system models, but both assume the posterior probability density to be Gaussian. Particle filters, on the other hand, have no such assumption and therefore perform better when the posterior densities are non-Gaussian. Particle filtering has many variations explained by Arulampalam et al. [5]. We use the Condensation algorithm [6] because it has an advantage that the weights can be easily calculated.

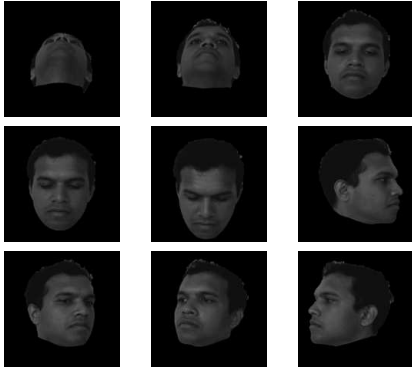Section 2 describes the Isomap algorithm and also presents

Fig. 1. Sample images from the training set.

(a)   (b)

Fig. 2. (a) Plot of the residual variance vs. dimensionality for the training data. It can be seen that the curve elbows at dimensionality 3. (b) The low-dimensional structure recovered by Isomap from the high-dimensional data. It resembles a tetrahedron.

(a)   (b)

Fig. 3. Two different two-dimensional views of the low-dimensional structure with training templates corresponding to some points placed near them to help visualize how the high-dimensional(151×151) points are embedded in 3D.

the results given by the algorithm on the training data used for this experiment. Section 3 explains the implementation of the particle filtering algorithm used for tracking. The results are reported in Section 4 with some conclusions in Section 5.

2. ISOMAP

2.1. Isomap algorithm

Isomap algorithm [2] has three steps. The algorithm takes as input the distances between all pairs from N data points in the high-dimensional input space X, measured in some domain-specific metric, e.g., Euclidean distance. The algorithm outputs coordinates in a d-dimensional Euclidean space which best represents the intrinsic geometry of the data. The three steps of the algorithm are as follows:

1. *Construct neighborhood graph:* A point is a neighbor of any other point if it lies within a fixed radius ϵ or is one of the K closest points to it. The neighborhood graph is constructed with edges equal to the distance between the points.

2. *Compute shortest paths:* In this step the geodesic distance between all points is calculated by computing the shortest paths in the neighborhood graph.

3. *Construct d-dimensional embedding:* Classical MDS is applied in this last step to obtain a low-dimensional embedding of the data. Thus the prime difference between MDS and Isomap is the use of geodesic distances in Isomap.

2.2. Results given by Isomap on the training data used for the experiment

The training data used for this experiment include 145 grayscale images of size 151×151 pixels. A few samples are shown in Figure 1. These images include a person's head varying in up-down pose and left-right pose with no variation in the ambient light and no rotation of the head in the plane of the image.
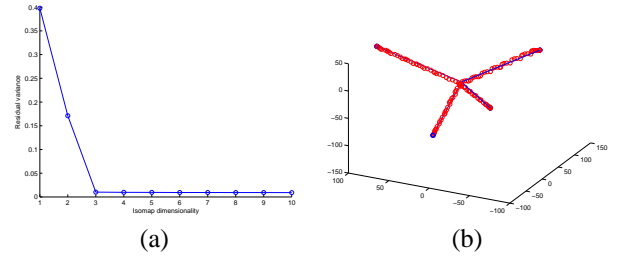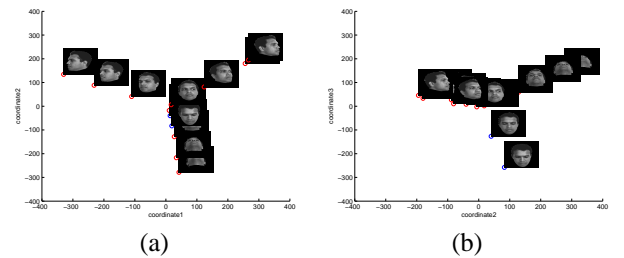
In-plane rotated templates are not included in the training set because such rotation is handled in the particle filter itself. We use SSD between all pairs of images as the distance metric for computing Isomap. Other distance measures like the Hausdorff distance [7] or chamfer distance [8] can be used to improve the performance of Isomap [9].

Figure 2a shows that the residual variance vs. dimensionality plot elbows at dimensionality of 3 which means that the high-dimensional face data can be well represented in three dimensions. Figure 2b shows the arrangement of the points in three dimensions given by Isomap. With the face data varying in up-down pose and the left-right pose, one would expect Isomap to recover only two degrees of freedom. Any face has a vertical axis of symmetry but no horizontal axis of symmetry which might explain why Isomap puts the data on the arms of a structure giving maximum distance between the extreme poses. Every arm on this structure is basically the lower dimensional representation of the head moving in one particular direction with the face looking straight ahead placed at the center. Note in Figure 2b and Figure 3 that the arms of the structure are almost linear. This shape is useful, as will be shown in Section 3.2.

3. ISOMAP TRACKING WITH PARTICLE FILTERING

This section explains how the data embedded in 3 dimensions can be used by a particle filter to track a person's head and estimate its pose at any given time frame. The particle filter [6] has three steps which are resampling, predict(drift and diffuse) and update. We use 1000 particles for this experiment. Every particle is characterized by a 7-parameter state vector comprising of 2D translational coordinates, 3D coordinates given by Isomap, in-plane rotation parameter and scaling parameter. The following subsections explain the implementation and the design of the particle filter.

3.1. Initialization of state of the particles

Since there is no prior $p(x_t|x_{t-1})$ from which to sample at time $t = 0$, we initialize the in-plane-rotation and the scale of the particles randomly within certain limits. The 2D position of the particles are initialized near the person's face in the first image of the test sequence, while the 3D Isomap coordinates are initialized at the base of the four arms (near point (0,0,0) of the Isomap coordinate system). For automatic initialization, one could simply start with a higher number of particles and gradually throw away the unsubstantial ones.

3.2. System Model

We use a linear stochastic equation [6] of the form

$$s_{t(n)} = As'_{t(n)} + Bw_{t(n)}$$

where $w_{t(n)}$ is a vector of standard normal random variates and BB^T is the process covariance noise. The deterministic component of the motion is modeled under the assumption of constant velocity [10] and the system noise is modeled as Gaussian. The arms are approximated as straight lines and the particles are made to reside only on these arms. Now the interesting question is how to handle any head pose that is absent from the training data in a test sequence? Fortunately any such pose can be roughly approximated as a combination of an out-of-plane training template and an in-plane rotation. This makes up for the missing poses in the training data since it is not really possible to have all possible poses in the training data. To make the tracking efficient an Isomap-arm-switching model is used for the particles. In this model if the particles are very close to the base of the Isomap-arm and moving with a negative velocity they can drift onto other arms with some fixed probability.

3.3. Measurement Model

There is a rectangular patch in a test image associated with every particle given by its parameters, i.e., x-y coordinates, scale, and in-plane rotation. Every training template has 3D
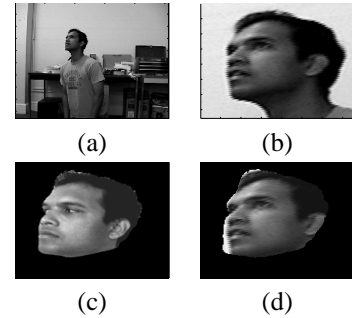

Fig. 4. (a) Image from the test sequence. (b) Image patch associated with a particle. (c) Template closest to the particle. (d) Mask of image 'c' put on image 'b'

Isomap coordinates. A particle is associated with a training template based on the minimum Euclidean distance between 3D Isomap coordinates of the training template and the 3D Isomap coordinates of the particle. The SSD between these two is computed only at the pixels where the intensity of the training template is non-zero, thus masking the background in the image patch and is used to compute the weight of the particle. Figure 4 illustrates the idea. These weights are used to form the pdf for the next state which is given by

$$p(x_{0:k}|z_{1:k}) \approx \sum_{i=1}^{N} w_k^i \delta(x_{0:k} - x_{0:k}^i)$$

3.4. Pose estimate

The weighted mean of the particles gives the estimate of the state of the system. The weighted mean of Isomap coordinates together with the in-plane rotation of the particle gives the 3D pose estimate of the person's head.

4. RESULTS

The tracker was tested on a separate image sequence with the same person's head undergoing translation and out-of-plane rotation. Tracking results are shown in Figure 5. The tracker successfully tracked the head of the person with a reasonably accurate estimation of the 3D pose. The tracker handled the out-of-plane rotation, the in-plane rotation and the rapid translation of the person's head. The tracker can also track the head in frames 50 through 60 even though the training data does not contain images which can exactly match these data. This validates our claim made in Section 3.2 about the approximation of such poses.

5. CONCLUSIONS

The results show that Isomap tracking with particle filtering is a good technique for tracking and pose-estimation of a person's head with a relatively small training data set. The other
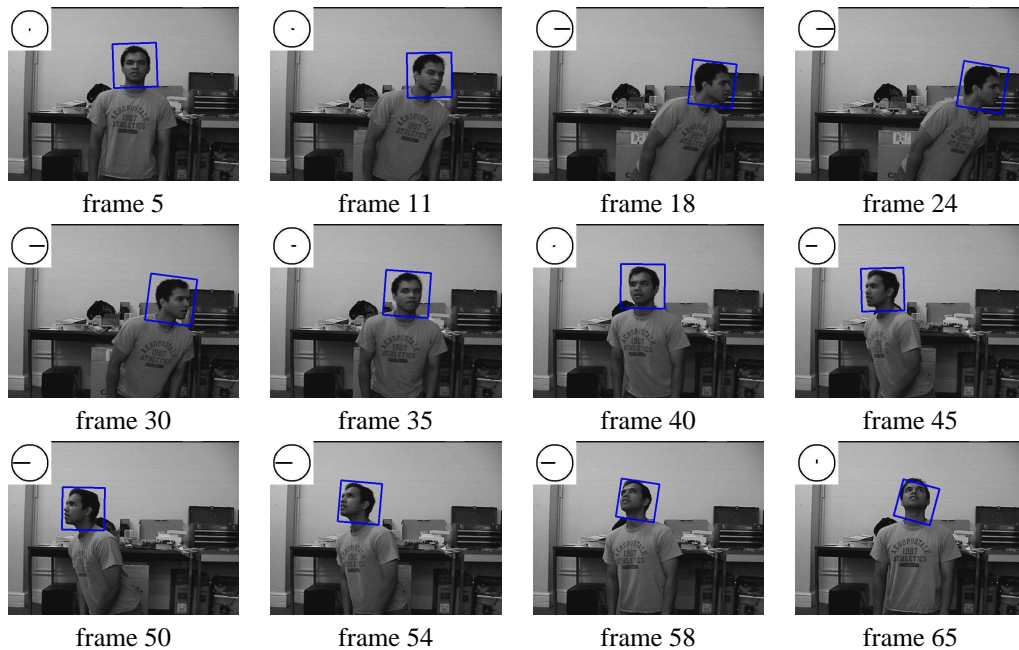
Fig. 5. Tracking results, shown as a blue square outlining the head. The vector inside the circle at the top-left corner of every frame gives the 2D out-of-plane rotation. In the circle the vertical axis represents the up-down pose and horizontal axis represents right-left pose. The length of the vector is directly proportional to the amount of rotation.

advantage of using this technique is that the particles run independently of each other except when they are resampled. Thus, this technique can be an excellent candidate for parallel computing.

This being just a preliminary attempt to use Isomap with a particle filter there are various areas of improvement. Finding a better distance measure for Isomap would make it more robust to noise and yield more accurate lower-dimensional structures. Further research needs to be done on the measurement model and weighting functions so that the tracker can handle occlusions, ambient light variations and large noise. This technique can be extended to track any kind of object if its training data is available.

6. REFERENCES

[1] T. F. Cox and M. A. A. Cox, *Multidimensional Scaling*, London: Chapman and Hall, 1994.

[2] J. Tenenbaum, V. Silva, and J. Langford, "A global geometric framework for non-linear dimensionality reduction," *Science*, vol. 290, pp. 2319–2323, 2000.

[3] S. Roweis and L. Saul, "Nonlinear dimensionality reduction by locally linear embedding," *Science*, vol. 290, pp. 2323–2326, 2000.

[4] P. Maybeck, *Stochastic Models, Estimation, and Control, Volume 1*, Academic Press, Inc., 1979.

[5] M. Arulampalam, S. Maskell, N. Gordan, and T. Clapp, "A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian filtering," *IEEE Transactions on Signal Processing*, vol. 50, no. 2, February 2000.

[6] M. Isard and A. Blake, "CONDENSATION – Conditional density propagation for visual tracking," *International Journal of Computer Vision*, vol. 29, no. 1, pp. 5–28, February 1998.

[7] D. Huttenlocher, G. Klanderman, and W. Rucklidge, "Comparing images using the Hausdorff distance," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 15, no. 9, pp. 850–863, 1993.

[8] H. G. Barrow, J. M. Tenenbaum, R. C. Bolles, and H. C. Wolf, "Parametric correspondence and chamfer matching: Two new techniques for image matching," *Proc. 5th Int. Joint Conf. Artificial Intelligence*, vol. 2, pp. 659–663, 1977.

[9] R. Souvenir and R. Pless, "Image distance functions for manifold learning," *Image and Vision Computing*, vol. 25, pp. 365–373, 2007.

[10] S. Birchfield, "An elliptical head tracker," in *Proceedings of the Thirty-first Asilomar Conference on Signals, Systems, and Computers*, Nov. 1997, vol. 2, pp. 1710–1714.