A COMPUTATIONALLY EFFICIENT 3D SHAPE REJECTION ALGORITHM

Yinpeng Chen *Hari Sundaram*
Arts Media Engineering, Arizona State University
Tempe, AZ 85281
Email: { yinpeng.chen, hari.sundaram }@asu.edu

ABSTRACT

In this paper, we present an efficient 3D shape rejection algorithm for unlabeled 3D markers. The problem is important in domains such as rehabilitation and the performing arts. There are three key innovations in our approach – (a) a multi-resolution shape representation using Haar wavelets for unlabeled markers, (b) a multi-resolution shape metric and (c) a shape rejection algorithm that is predicated on the simple idea that we do not need to compute the entire distance to conclude that two shapes are dissimilar. We tested the approach on a real-world pose classification problem with excellent results. We achieved a classification accuracy of 98% with an order of magnitude improvement in terms of computational complexity over a baseline shape matching algorithm.

1. INTRODUCTION

In this paper, we present a fast 3D shape rejection algorithm on unlabeled 3D markers. The problem is important in areas such as the performing arts [11] (e.g. dance) and rehabilitation [6], where real-time gesture tracking plays a significant role. 3D shape matching on unlabeled data is important since markers are often self-occluded or lost. The problem is made difficult since unlabeled marker data create a significant computational burden for real-time systems, due to the marker correspondence problem.

There has been prior work on shape matching algorithm for unlabeled data. Johnson and Herbert[10] presented a recognition algorithm based on computing correspondence using spin images that project 3D points to a 2D histogram. Recently, weighted graph matching [2,3,4,7] is a standard technique to align a pair of shapes represented by a set of descriptive local features. In [8], a fast contour matching using Earth Movers Distance (EMD) is presented. However, several challenging issues remain – (a) high computational complexity of extracting local features from 3D shapes, (b) high computational complexity of searching for the marker correspondences.

We propose an efficient 3D shape rejection algorithm based on hierarchical shape representation using global features. We build upon prior work pattern rejection for images [1]. We first represent the 3D markers using a Haar wavelet decomposition of the distance histogram. Then, we reject shapes that are dissimilar using a shape rejection algorithm. Our algorithm is predicated on a simple idea – if two shapes are very dissimilar, we need not compute the exact dissimilarity value to conclude that the shapes are dissimilar. We can come to this conclusion by only comparing a few Haar coefficients. This enables us to achieve a significant computational gain. We tested our algorithm on a real-world dance dataset with excellent results – we show a 98% accuracy with an order of magnitude saving over a baseline shape matching algorithm.

This paper is organized as follows. In section 2, we show how to represent a shape at multi-resolutions, and in section 3, we detail how we use the adaptive shape rejection in pose classification and discuss the computational complexity. In section 4, we show our experiment results and we present our conclusion in section 5.

2. MULTI-RESOLUTION SHAPE ANALYSIS

In this section, we discuss the multi-resolution representation of 3D dancer's shape. Each shape consists of 35 unlabeled 3D marker coordinates captured from a marker-based motion capture system. A calibrated 3D capture system (eg. VICON) usually provides labeled data, specifying the location on the body for each dancer. We focus on unlabeled data because in multi-dancers scenario, some markers are lost or incorrectly labeled due to self-occlusion or inter-occlusion. Figure 1 shows two dancer shape examples each with 35 unlabeled markers on the dancer's body.

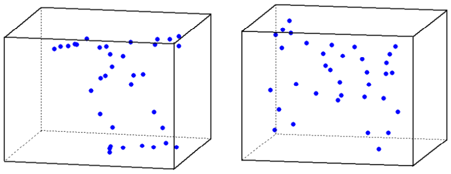

Figure 1: Two examples of dancer's pose from dataset

2.1 Feature Extraction

Let us denote the unlabeled 3D marker coordinates of a shape as $X_i=(x_i,y_i,z_i)^T$, $i=1,\dots,N$ where N is the number of markers. We create an object centric coordinate system, by moving the origin to the centroid of N markers. We extract the distance from each marker to the center denoted as r_i (shown in Figure 2 (a)) and divide r_i by a constant R_{max} to normalize them to interval [0,1]. The computational complexity to extract the normalized distance is 13*N operations. In this paper, we assume that a single real addition, subtraction or multiplication use equivalent resources. Our framework can easily handle the case when the complexity is different for different operations.

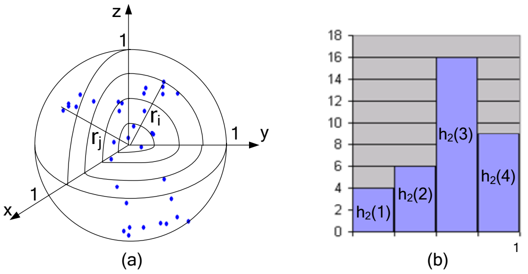

Figure 2: (a) shape in normalized 3D space that is divided into four distance bins. The shape centroid is the origin, r_i and r_j are normalized distances of i^{th} and j^{th} markers. (b) Distance histogram**.**

2.2 Multi-resolution representation

In this subsection, we discuss multi-resolution shape representation. First, we introduce distance histogram construction. Then, we show how Haar wavelet [12] basis is used to decompose the distance histogram at multi-resolutions. Finally, we discuss the computational complexity of 3D shape representation.

0-7803-9332-5/05/$20.00 ©2005 IEEE

2.2.1 Distance Histogram

The distance values r_i of the 3D markers form an unordered set. This is because the markers do not have any labels associated with them. Hence, it is difficult to find the corresponding markers between two shapes although they may be attached to the same body location. As the first step towards 3D shape representation, we transform the distance values with a distance histogram. At resolution *J*, we uniformly divide the normalized distance space [0,1] into $K=2^J$ bins. Thus, the histogram with *J* resolutions – $h_J(k)$ is represented as:

$$h_J(k) = | \{ r_i | r_i \in [\frac{k-1}{2^J}, \frac{k}{2^J}), i = 1,..,N \} | , \qquad <1>$$

where *N* is the number of markers and |·| is cardinality (set size) operator. Figure 2 (a) shows the case where the normalized 3D space is divided into 4 distance bins and Figure 2 (b) shows the distance histogram based on this division where $J=2$, $h_2(1)$, $h_2(2)$, $h_2(3)$ and $h_2(4)$ are the number of markers in the four bins.

2.2.2 Haar Wavelet Decomposition

Wavelet decompositions allow for very good image approximation with just a few coefficients [12] and work very effectively in multi-resolution image queries [9]. A Haar wavelet decomposition of the distance histogram would provide a good foundation on which to build a shape metric. Haar wavelets are also fast to compute and simple to implement. With Haar basis, the histogram with *J* resolutions can be represented as a linear combination of the Haar basis functions:

$$h_J(k) = c_0^0 \phi_0^0(k) + \sum_{j=0}^{J-1} \sum_{i=0}^{2^j-1} d_i^j \psi_i^j(k) , \qquad <2>$$

where c_0^0 and $\phi_0^0$are the scaling coefficient and scaling function respectively, d_i^j and ψ_i^j are the wavelet coefficient and wavelet function of i^{th} Haar basis at the resolution level *j*. The wavelet coefficient d_i^j can be obtained by:

$$d_i^j = \langle h_J | \psi_i^j \rangle , \qquad <3>$$

Where < | > is the inner product operator. Figure 3 shows the scaling function and Haar wavelets functions at resolution 1, 2.

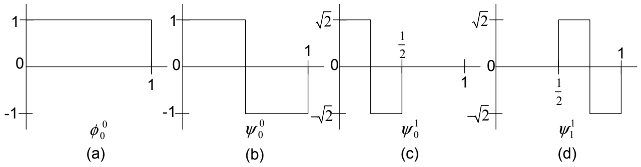

Figure 3: Haar wavelets. (a) Scaling function ϕ_0^0, (b) wavelet at resolution *J=1*, (c-d) wavelet functions at resolution *J=2*.

Since each shape consists of the same number of markers *N*, different shapes will share the same $c_0^0=N/2^J$. Hence, we only use wavelet coefficients d_i^j to represent the shape histogram. The Haar wavelet coefficient vector f_J with resolution *J* is defined as:

$$f_J = [d_0^0, d_0^1, d_1^1, ..., d_0^{J-1}, ..., d_{2^{J-1}-1}^{J-1}]^T \qquad <4>$$

The size of the coefficient vector at resolution *J* is 2^J*-1*. Note that first 2^{J-1}*-1* entries correspond to f_{J-1}, the wavelet decomposition at resolution *J-1*. The last 2^{J-1} coefficients represent the detail at resolution *J*.

2.2.3 3D Shape Representation Complexity

The 3D shape is represented using the 1D Haar wavelet decomposition of the distance histogram. The computational complexity of the representation is then due to two parts - (a) computing 1D histogram (eq.<1>), and (b) complexity of Haar wavelet decomposition (eq.<3>). The complexity of computing histogram is $J*N+2^J-1$ operations, and the complexity of Haar wavelet decomposition is $2*(2^J-1)$ operations at resolution *J*. Therefore, the overall complexity is represented as:

$$C_{J,Haar} = J \cdot N + 3 \cdot (2^J - 1) \qquad <5>$$

We can see that the computational complexity of computing histogram increases linearly in number of markers – N, and the computation complexity of Haar wavelet coefficients increase exponentially with J.

3. SHAPE REJECTION

In this section, we use the shape representation discussed in section 2 for pose classification using the nearest neighbor technique (1-NN) [5]. Here, the input shape will be classified as one of the *M* poses or will be rejected. The computation of shape distance between the input and each class is very expensive. Here we present an adaptive framework for classification using shape rejection. We can quickly reject most of classes that are dissimilar with input shape and find the class which is closest to the input.

3.1 Training

In the training phrase, we first compute the mean Haar wavelet coefficient vector at resolution *J=4* for each class:

$$f_{J,i}^{\mu} = \frac{1}{K_i} \sum_{j=1}^{K_i} f_{J,i,j} , \qquad <6>$$

where $f_{J,i}^{\mu}$ is the mean Haar wavelet coefficient vector of the i^{th} class at resolution *J*, $f_{J,i,j}$ is the Haar wavelet coefficient vector of the j^{th} training sample in the i^{th} class at resolution *J* and K_i is the number of training samples in the i^{th} class. Then a set of thresholds $\{\alpha_i^k\}$ are obtained by:

$$\alpha_i^k = \max_j \left| f_{J,i,j}(k) - f_{J,i}^{\mu}(k) \right| , \qquad <7>$$

where α_i^k is the threshold corresponding to the k^{th} entry of coefficient vector of the i^{th} class which represents the maximum distance between the k^{th} Haar wavelet coefficient of training samples - $f_{J,i,j}(k)$ and the k^{th} mean Haar wavelet coefficient $f_{J,i}^{\mu}(k)$ of the i^{th} class.

3.2 Shape rejection and distance metric

Shape rejection is based on the idea that if two shapes are very dissimilar, we need not compute the exact dissimilarity value to conclude that the shapes are dissimilar. This is achieved by only comparing a few coefficients. We adopt a supervised approach for shape rejection using Haar wavelet coefficients.

In the comparison between the input shape and the shape of the i^{th} class at resolution *J*, we first compare the Haar wavelet coefficient vector of input shape denoted as $f_{J,input}$ with the mean Haar wavelet coefficient vector of the i^{th} class - $f_{J,i}^{\mu}$. If there exists *k* ($1 \le k \le 2^J-1$) such that:

$$\left| f_{J,input}(k) - f_{J,i}^{\mu}(k) \right| > \alpha_i^k , \qquad <8>$$

the i^{th} class *will be rejected as dissimilar with the input*. Otherwise, we compute the L_1 distance of their Haar wavelet coefficient vectors as distance between the input and the i^{th} class:

$$d_J(s_{input}, s_i) = \sum_{k=1}^{2^J-1} \left| f_{J,input}(k) - f_{J,i}^{\mu}(k) \right| , \qquad <9>$$

where s_{input} is the input, s_i is the shape of the i^{th} class, $d_J(s_{input}, s_i)$ is the shape distance between s_{input} and s_i at resolution J.

Since the Haar wavelet coefficient is hierarchically ordered in the coefficient vector, the shape rejection starts with the most important coefficient to the most detailed coefficient. Hence, we conjecture that the shape rejection is processed in the most efficient way. This is validated in the experiments section.

In eq.<9>, we can see that computational complexity of the full shape distance at resolution J is 2^{J+1}*-3* operations. We define it as the static computational complexity $C_{J,d}^{\ s}$:

$$C_{J,d}^{s} = 2^{J+1} - 3 \ . \qquad <10>$$

Let us define the index of end Haar wavelet coefficient for the rejection stage of i^{th} class – $k_{end}(i)$. If the i^{th} class is rejected, we only need $k_{end}(i)$ subtractions and $k_{end}(i)$ comparisons (see eq.<8>) without computing the shape distance. Thus the computational gain is $C_{J,d}^{\ s}$*-2**$k_{end}(i)$ operations. However, if there is no rejection we waste 2^J*-1* comparisons for rejection verification before computing the shape distance at resolution J. Combining these two cases, the computational gain due to shape rejection is:

$$\Delta_{J,d}(i) = \begin{cases} C_{J,d}^{s} - 2 \cdot k_{end}(i) & \textit{if rejected} \\ -(2^J - 1) & \textit{otherwise} \end{cases} , \qquad <11>$$

where $\Delta_{J,d}(i)$ is defined as the computational gain of shape distance between the input and the i^{th} class at resolution J. We can see the computational gain is positive in rejection case except $k_{end}(i)=2^J-1$ and negative in non-rejection case or $k_{end}(i)=2^J-1$. Combining eq.<10> and eq.<11>, we obtain the adaptive computational complexity of shape distance $C_{J,d}^{\ a}$ between the input and the i^{th} class at resolution J:

$$C_{J,d}^{a}(i) = C_{J,d}^{s} - \Delta_{J,d}(i) \qquad <12>$$

3.3 Framework for Pose Classification

We now discuss the pose classification framework (ref. Figure 4). First we compute the Haar wavelet coefficient vector of the input, and then compare it with the mean Haar wavelet coefficient vector of each class and reject classes that are dissimilar with the input. If all classes are rejected, we will reject the input shape for classification. Otherwise, we compute the shape distances between the input shape and poses of classes left after rejection using <9> and using 1-NN method to classify the input into the class with minimum shape distance.

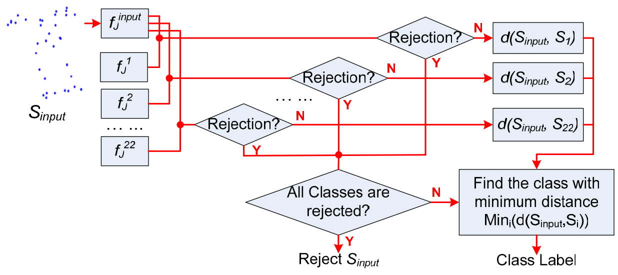

Figure 4: Framework of pose classification

3.4 Computational Complexity

We now discuss the static computational complexity of pose classification using 1-NN algorithm based on shape distance eq.<9>, without shape rejection. The static computational complexity includes feature extraction (*13*N* operations), Haar wavelet decomposition (eq.<5>), shape distances between the input and M pose classes (eq.<10>), and searching for minimum distance (*M-1* operations). Hence the static computational complexity at resolution J is:

$$C_{J,Clasificaiton}^{s} = 13N + C_{J,Haar} + M \cdot C_{J,d}^{s} + (M-1) \qquad <13>$$

Using our adaptive framework with shape rejection, the computational complexity is decreased because most of classes are rejected. Let us assume that L classes remain, the overall computational gain is:

$$\Delta_{J,Classification} = \sum_{i=1}^{M} \Delta_{J,d}(i) + (M - L) \qquad <14>$$

Where $\Delta_{J,d}(i)$ is the computational gain of shape distance (eq.<11>) between the input and the i^{th} class. Therefore, combining (eq.<13>) and (eq.<14>), we obtain the overall adaptive computation complexity of pose classification at resolution J is:

$$C_{J,Classification}^{a} = C_{J,Classification}^{s} - \Delta_{J,Classification} \qquad <15>$$

As we discussed in section 3.2, $\Delta_{J,d}(i)$ is positive in rejection case and negative in non-rejection case. However, for each input shape, the number of classes rejected is much larger than the remaining classes after rejection verification.

4. EXPERIMENTS

In our experiments, the data is captured by using sixteen-camera MotionAnalysis system with frame rate of 120Hz. This set of data, created by a world renowned choreographer Bill T. Jones, has 22 different gestures, each with specific meaning. The size of the data set is 24,600 frames. In our experiment, we use 75% frames of each gesture as the training data, and the remaining 25% for testing. We used labeled marker data to compute the best case, baseline result. The baseline results algorithm will have good accuracy due to label correspondence information.

4.1 Shape matching algorithm for labeled data

We now briefly discuss the baseline algorithm using labeled markers. Due to the labeled marker data, we now have correspondence information. The computational complexity of this algorithm is due three parts – (a) translating each marker to the object centered coordinate system (b) 2D rotation correspondence and (c) Euclidean distance computation. Note that our proposed algorithm is rotation invariant. We observed that in our dataset, there was very little intra-class shape rotation. With M classes and N markers per shape, the computational complexity (excluding rotation) per input is 10NM+6N-1=7909 operations (M=22, N=35). If rotation is included, then the computational complexity will increase about $8log_2[(K+1)/2]+1$ times, where K represents the number of discrete intervals for the 2D angles.

4.2 Results

Table 1 shows the classification results using shape rejection algorithm for unlabeled data and shape matching algorithm for labeled data. Compared with shape matching algorithm, shape rejection algorithm at resolution 4 has only tenth of computational complexity with a little accuracy tradeoff. Note that the baseline algorithm does *not* include rotation correspondence.

Table 1: Comparison between shape rejection algorithm and the baseline shape matching algorithm. The 2nd, 3rd columns are average accuracy and operation numbers per input shape over all classes.

Algorithm	Accuracy	Complexity
Shape rejection algorithm (J=4)	**98.53%**	**755**
Baseline matching algorithm	98.81%	7909

In Table 2, we observe that our rejection mechanism not only saves computational complexity, but also improves the classification accuracy. This is because in the case where the input has small distance at each coefficient and large overall distance with its own class while it has large distances at a few coefficients and small overall distance with another class, it is classified incorrectly only based on the overall shape distance. However, with shape rejection, this problem is easily solved by rejecting the class which has large distance at any coefficient with the input. It is also observed that the computational gain of adaptive algorithm is negative at resolution 1. This because that the rejection stage k_{end} is always equal to *1* at resolution 1 and $\Delta_{J,d}$ in <11> is -1.

Table 2: Pose classification results at different resolutions. The 2nd, 3rd columns are the average accuracy using adaptive shape rejection and 1-NN classification based on shape distance (eq.<9>) respectively. The 4th column is the average computational gain using adaptive shape rejection compared with 1-NN algorithm directly.

Resolution	Adaptive Accuracy	Static Accuracy	Computational Gain
1	31.62 %	27.92 %	-10
2	81.98 %	70.17 %	+43
3	95.98 %	88.24 %	+199
4	**98.53 %**	**95.19 %**	**+544**

Figure 5 shows the probability distribution over rejection stage at resolution 4. We can see that for an input shape, about 90% classes are rejected using less than 7 coefficients and full shape distance is only used for the classification of left 10% classes.

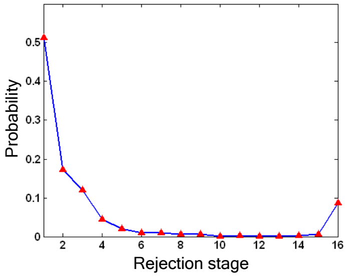

Figure 5: Pdf of the rejection stage at resolution *J=4* over all classes. Most of the shapes are rejected using a few Haar coefficients, resulting in a large computational gain.

5. CONCLUSION

In this paper, we have presented a fast 3D shape rejection algorithm on un-labeled markers. There are two key innovations (a) a multi-resolution 3D shape representation using the Haar basis. This representation is rotation invariant. (b) a 1-NN pose classification algorithm that uses *shape rejection* framework. This utilizes the Haar basis coefficient ordering, to achieve speedup. We evaluated our framework on real-world pose classification problem. Our experimental results are excellent with 98.5% percent accuracy and order of magnitude decrease in computational complexity over the baseline algorithm. In the future, we are planning to investigate shape rejection algorithms that incorporate shape complexity. We also are planning to incorporate high-level syntactical constraints into our approach.

6. REFERENCES

[1] S. Baker and S. K. Nayar (1996). *Pattern Rejection*, Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, San Francisco,

[2] S. Belongie, J. Malik and J. Puzicha (2002). *Shape Matching and Object Recognition Using Shape Contexts.* IEEE Transactions on Pattern Analysis and Machine Intelligence **24**(24): 509-522.

[3] W. Cook and A. Rohe (1999). *Computing minimum-weight perfect matchings.* INFORMS Journal on Computing **11**: 138-148.

[4] D. E. Drake and S. Hougardy (2003). *A simple approximation algorithm for the weighted matching problem.* Information Processing Letters **85**(4): 211-213.

[5] R. O. Duda, P. E. Hart and D. G. Stork (2001). Pattern classification. New York, Wiley**:** xx, 654.

[6] C. Ghez, T. Rikakis, R. L. DuBois, et al. (2000). An Auditory display system for aiding interjoint coordination. Proc. International Conference on Auditory Display. Atlanta, GA.

[7] S. Gold and A. Rangarajan (1996). *A Graduated Assignment Algorithm for graph matching.* IEEE Transactions on Pattern Analysis and Machine Intelligence **18**(4): 377-388.

[8] K. Grauman and T. Darrell (2004). *Fast Contour Matching Using Approximate Earth Mover's Distance*, Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, June.

[9] C. E. Jacobs, A. Finkelstein and D. H. Salesin (1995). *Fast Multiresolution Image Querying*, Computer Graphics, Annual Conference Series (Siggraph'95 Proceedings), 277-286,

[10] A. E. Johnson and M. Hebert (1997). *Recognizing objects by matching oriented points*, Proc. Computer Vision and Pattern Recognition (CVPR '97), 684-689,

[11] G. Qian, F. Guo, T. Ingalls, et al. (2004). A Gesture Driven Multimodal Dance System. IEEE International Conference on Multimedia and Expo. Taipei, Taiwan.

[12] E. J. Stollnitz, A. D. DeRose and D. H. Salesin (1995). *Wavelets for computer graphics: a primer.1.* Computer Graphics and Applications, IEEE **15**(3): 76-84.