FAST HUMAN POSE RETRIEVAL USING APPROXIMATE CHAMFER DISTANCE

Hui Cao, Noboru Ohnishi, Yoshinori Takeuchi[†]*, Tetsuya Matsumoto, Hiroaki Kudo*

Nagoya University
Graduate School of Information Science, [†]Information Security Promotion Agency
Furo-cho, Chikusa-ku, Nagoya, Aichi 464–8603

ABSTRACT

The estimation of 3D human pose from a single image can be implemented in the way of large-scale image retrieval. For a given input image, a few similar images are retrieved from the database consisting of human figure images annotated with 3D human poses; then the 3D poses corresponding to retrieved images serve as the pose estimates of input image. This retrieval-based method sounds too simple, but it works if two conditions are met: (i) sufficient data and (ii) good image matching algorithm. Sufficient data can be generated by means of 3D character rendering software and various human motion data. As for good image matching algorithm, here we employ the chamfer distance which has proved to be an effective tool for shape comparison in many works. However, applying the chamfer distance, as well as other good image-matching algorithms, to large-scale problem would lead to high time requirements. In order to address this computational issue, here we propose an approximate chamfer distance which is capable of achieving significant efficiency improvements with slight accuracy loss: over three hundred times faster than exact chamfer distance in current implementation.

1. CHAMFER DISTANCE

The similarity between two silhouette images can be measured using chamfer distance. Suppose we have two contours to compare, one is the query $Q = \{q\}$ and the other is a template $T = \{t\}$. The chamfer distance from T to Q is measured as the mean value of distances between points in the template to their closest points in the query:

$$d_{cham}^{(T,Q)} = \frac{1}{N_T} \sum_{t \in T} \min_{q \in Q} ||t - q||, \tag{1}$$

where N_T is the number of points in T and $|| \cdot ||$ can be any norm, eg., Euclidian or Cityblock. The chamfer distance can be efficiently computed by first calculating the distance transform (DT) of the Q using the two-pass algorithm. The value of each pixel in the DT_Q is the distance from this pixel to the closest pixel in the Q:

$$DT_Q(p) = \min_{q \in Q} ||p - q||. \tag{2}$$

Thus, the min operation in (1) becomes a simple look-up or a correlation, and Equation (1) is expressed as follows

$$d_{cham}^{(T,Q)} = \frac{1}{N_T} \sum_{t \in T} DT_Q(t) = \frac{1}{N_T} T^T DT_Q. \tag{3}$$

However, the chamfer distance $d_{cham}^{(T,Q)}$ is asymmetric, that is, $d_{cham}^{(T,Q)} \neq d_{cham}^{(Q,T)}$. To better measure the similarity between two contours, it is preferable to using the symmetric chamfer distance:

$$D_{cham}^{(Q,T)} = d_{cham}^{(T,Q)} + d_{cham}^{(Q,T)}, \tag{4}$$

which is the average of chamfer distances from bi-directions.

2. APPROXIMATE CHAMFER DISTANCE

Although the chamfer distance is efficient to compute, it would still cause a computational issue for the current task which is large-scale matching problem. The computation of DT of database images (contours) is the bottleneck of applying chamfer distance to large-scale matching problem. To address this issue, we propose an efficient approximate chamfer distance using eigen approximation. The central idea of approximate chamfer distance is thus to use low-dimensional subspace representation of DT to replace the exact DT in chamfer distance. The subspace of DT is learned to capture the major variances of DTs of all database images. The main components of learned subspace are the mean vector $\vec{m}$ and the first k eigenvectors, $\{\vec{e}_i\}_{i=1}^k$. We then compute offline and store the subspace coefficients, $\{\vec{e}_i\}_{i=1}^k$, corresponding to eigenvectors $\{\vec{e}_i\}_{i=1}^k$ for all DTs of database images. Each database DT can thus be approximately represented by the following linear combination,

$$DT_x \approx \sum_{i=1}^{k} f_{xi} \vec{e}_i + \vec{m}. \tag{5}$$

By replacing the DT_Q in the formula of chamfer distance (3) with the subspace approximation (5), it is straightforward to derive the approximate chamfer distance. The approximate

Table 1. Comparison of estimation accuracy

Methods	Approximate		Partially Approx.		Exact
	m=20	m=100	m=20	m=100	
$k=1$	57%	64%	65%	68%	70%
$k=3$	80%	84%	85%	86%	87%
$k=5$	84%	88%	89%	90%	90%
$k=7$	86%	90%	91%	91%	92%
$k=10$	87%	91%	92%	92%	93%

chamfer distance is expressed as follows:

$$\begin{aligned} d_{cham}^{(T,Q)} &\approx \frac{1}{N^T}T^T\left(\sum_{i=1}^{k} f_{qi}\vec{e}_i + \vec{m}\right) \\ &= \sum_{i=1}^{k} f_{qi}\left(\frac{1}{N^T}T^T\vec{e}_i\right) + \frac{1}{N^T}T^T\vec{m} \\ &= \vec{f}_{eigen}^{(Q)} \vec{d}_{cham}^{(T,\{m,E\})}. \end{aligned} \quad (6)$$

In (6), $\vec{f}_{eigen}^{(Q)}=[1, f_{q1}, \ldots, f_{qk}]$ denotes the vector of eigen coefficients of DT_Q after projecting onto the subspace; and $\vec{d}_{cham}^{(T,\{m,E\})} = [\frac{1}{N^T}T^T\vec{m}, \frac{1}{N^T}T^T\vec{e}_1, \ldots, \frac{1}{N^T}T^T\vec{e}_k]^T$ is the vector of eigen chamfer distances — the chamfer distances from T to mean vector and eigenvectors of DT subspace. As a result, the subspace approximation to chamfer distance $d_{cham}^{(T,Q)}$ is expressed by an inner-product of the eigen coefficients of DT_Q and the eigen chamfer distances of T.

In a similar way, we can obtain the expression of approximate chamfer distance $d_{cham}^{(Q,T)}$:

$$d_{cham}^{(Q,T)} \approx \vec{f}_{eigen}^{(T)} \vec{d}_{cham}^{(Q,\{m,E\})}, \quad (7)$$

where $\vec{f}_{eigen}^{(T)}=[1, f_{t1}, \ldots, f_{tk}]$ and $\vec{d}_{cham}^{(Q,\{m,E\})} = [\frac{1}{N^Q}Q^T\vec{m}, \frac{1}{N^Q}Q^T\vec{e}_1, \ldots, \frac{1}{N^Q}Q^T\vec{e}_k]^T$ is the vector of eigen coefficients of DT_T and eigen chamfer distances, respectively.

The benefit of above approximation is that the computation of symmetric chamfer distance is decomposed into online and offline parts. The online part includes the computations of eigen chamfer distances $\vec{d}_{cham}^{(Q,\{m,E\})}$ and eigen coefficients $\vec{f}_{eigen}^{(Q)}$, both are computed once before matching; while the offline part includes the computations of the eigen chamfer distances $\{\vec{d}_{cham}^{(T,\{m,E\})}\}$ as well as the eigen coefficients $\{\vec{f}_{eigen}^{(T)}\}$, for all database examples. Thus, the efficiency obtained from subspace approximation is that the majority of computing cost can be finished by offline computation.

We also propose a partially approximate chamfer distance which uses the approximate computation for $d_{cham}^{(Q,T)}$ while computes the $d_{cham}^{(T,Q)}$ in the way of exact chamfer distance.

Table 2. Comparison of time and memory usages

Approximate Chamfer			Exact Chamfer	
m=20	m=50	m=100	Accelerated	Normal
0.47s	0.51s	0.53s	1.07s	171.81s
6.5M	16.0M	31.8M	150.1M	0

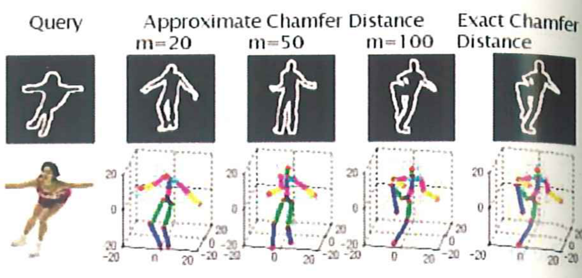

Fig. 1. An example showing the superiority of the approximate chamfer distance

2.1. Image Normalization

We have to normalize the input image before matching it to database images. From database images, we learned a multivariate linear regression model between the area of foreground region and an invariant shape descriptor: 10 Hu moments of foreground region. Depending on the regression model, we infer for input image the proper area of foreground that is likely to be aligned with database images, by which we scale the input image and then move the foreground to the center of the image.

3. EXPERIMENTAL RESULTS

The human pose database contained 14,964 human figure images rendered in various 3D poses and view angles. Three levels of subspace approximation were tested. Table 1 shows the estimation accuracy achieved in the case of partial database images as queries. The approximate chamfer distance using 100 eigenvectors and partially approximate distances at any approximation level attained close accuracy to the exact chamfer distance. We also conducted real image experiment on 50 sports images collected from internet. Sometimes poses were wrongly understood by the exact chamfer distance while the approximate chamfer distance's result is closer to the true 3D pose. An examples showing superiority of approximate distance is shown in Fig.1. Table 2 summarizes the time (for 14,964 matches) and memory usages of each distance. The approximate chamfer distance attained the best performance in terms of both time and memory usages; the exact chamfer distance is considerable slow and its accelerated version (precomputing DTs for all database images) has a high memory requirement.