Local Earth Mover’s Distance and Face Warping

S H Srinivasan
Applied Research Group
Satyam Computer Services Ltd, Bangalore
SH_Srinivasan@satyam.com

Abstract

Earth mover’s distance is one of the recently proposed distance measures in multimedia. Though attractive, earth mover’s distance is computationally complex. In this paper, we propose local earth mover’s distance which reduces the size of the linear program used by exploiting locality constraints. The speedup and approximation capabilities of local earth mover’s distance are demonstrated. Local earth mover’s distance is then applied to warping face images.

1. Introduction

In several multimedia applications, there is a need to compute distances between various objects - feature vectors, histograms, images, etc. Several distance measures have been used - from the familiar Euclidean distance for vectors to Kullback-Leibler divergence measures for distributions. Earth mover’s distance (EMD) has been proposed recently as a distance measure between histograms, or more generally between signatures [3]. EMD is based on a transportation problem formulation which in turn can be formulated as a linear programming problem. Though linear programming is in polynomial time, the widely available simplex method has exponential time worst case complexity. Several optimizations are available for transportation problems. Even with these optimizations, it will be of interest to reduce the complexity of EMD calculation. In this paper, we show how the use of locality constraints can lead to reduction in size of the linear program. The new formulation is called local earth mover’s distance (LEMD).

Our interest in EMD is in calculating warping distance between faces. When two face images are compared, there is often a need to warp one face to the other. Face warping is usually performed using optical flow or dynamic programming. In this paper, we show how EMD or LEMD can be used for warping faces.

This paper is organized as follows. Section 2 provides a brief introduction to EMD. Section 3 discusses the use of EMD for face warping. Section 4 describes local EMD. Simulation results are presented in section 5. The paper closes with conclusions.

2. Earth mover’s distance

EMD was initially proposed as a way to compute distances between histograms or signatures. For simplicity, the following discussion is in terms of histograms.

Let \((x_i, w_i), i = 1, 2, \ldots, M\) and \((y_j, u_j), j = 1, 2, \ldots, N\) be two histograms (of possibly different sizes). We can consider \(x_i\) and \(y_j\) to be bin centers and \(w_i\) and \(u_j\) to be the corresponding bin counts. The conventional histogram distances compare counts of bins having the same center. It is possible that bin centers have changed because of varying environmental conditions like illumination. Hence a better approach is to compare bins with different centers – albeit with a higher cost. The cost of comparing bins \(x_i\) and \(y_j\) is given by

\[
d_{ij} = d(x_i, y_j), \quad i = 1, 2, \ldots, M, \quad j = 1, 2, \ldots, N
\]  

where \(d(\cdot, \cdot)\) is the distance measure used between bin centers. EMD is based on the following analogy. We consider \(w_i\)s to be masses centered on \(x_i\)s. \(u_j\)s are considered to be holes centered on \(y_j\)s. We now need to move the masses to fill the holes. The cost of moving unit mass from \(x_i\) to \(y_j\) is \(d_{ij}\). Let \(f_{ij}\) be the amount of mass actually moved from \(x_i\) to \(y_j\). Then the total cost of moving is given by

\[
\sum_i \sum_j d_{ij} f_{ij}
\]  

We need to find \(f_{ij}\)s such that the cost of movement is minimum. \(f_{ij}\)s obey the following obvious constraints.

1While histograms have fixed size bins and associated counts, signatures have variable sized bins and associated weights.
compute histogram distances, we use EMD to compute the amount of pixels moved is conserved. Another popular can-

servation constraints (equation 3) which ensures that the

does not guarantee one-to-one mapping but enforces mass that several pixels are mapped to the same pixel. EMD also

mapping when calculating optical flow. Hence it is possible

4. Local EMD

An important aspect of linear programming is the number of variables. For computing the EMD between two

m × n images, let us compute the number of variables. Each

3D descriptor has size m × n (see equation 4). Hence the

number of variables involved is m^2 × n^2. If we use the original image sizes (94 × 112), the number of variables is very

large (110,838,784). For reasonable number of variables (≈ 10,000), the images have to be downsampled. For example, 9 × 11 downsampled images result in 9801 variables.

Since the number of variables is large (even for small images), it is worthwhile to develop techniques which re-

duce the number of variables. Consider the distance function given in equation 5. Since spatial distances ((i_1 - i_2)^2 + (j_1 - j_2)^2) are involved, total distance is will be large even if intensity differences ((I(i_1, j_1) - J(i_2, j_2))^2) are small for pixels which are far apart. It is unlikely that pixels far apart are matched. Hence we can ignore candidate matches between pixel positions which are far apart. Let W be the local window over which matches are possible and let w be the size of this local window. Hence f_{ij} = 0 if j does not belong to the local window. Hence we reformulate EMD as follows. Let f_{ij} be the amount of earth (or illumination) moved from pixel i in first image to pixel j in second image. It is important to note that j is the index inside the window. Hence j = 1, 2, · · ·, w. See figure 1 for the notation. The number of variables in LP is m × n × w. For 9000 variables (9216, to be precise), we can now have m = 24, n = 24, and w = 16. The image size has nearly doubled.
Since the variables are different, this also calls for a re-formulation of EMD which we call local EMD. The constraints for the linear program corresponding to local EMD between images of size \( m \times n \) are as follows (using notation from figure 1). Here \( i \) is the pixel index in the first image, \( k \) is the index in the second image. \( j \) is index inside the windows.

\[
\begin{align*}
\sum_{j \in W_i} f_{ij} & \geq 0 \\
\sum_{j \in W_i} f_{ij} & \leq w_i, \; i = 1, 2, \ldots, mn \\
\sum_{i \in W^{-1}_k, j = \rho^{-1}(i,k)} f_{ij} & \leq u_k, \; k = 1, 2, \ldots, mn \\
\sum_{i} \sum_{j \in W_i} f_{ij} & = \min \left( \sum_{i} w_i, \sum_{k} u_k \right)
\end{align*}
\]

The objective function to be minimized is given by \( \sum_{i} \sum_{j \in W_i} d_{ij} f_{ij} \)

Each pixel \( i \) in the first image has a neighborhood \( W_i \) associated in the other image. This neighborhood determines the permitted spatial positions of earth movement. The top row shows a \( 3 \times 3 \) neighborhood. The pixels in the window have two associated coordinate system – local coordinate system (index inside the window) and global coordinate system (index in the image). In the figure shown, \( j \) is the local coordinate and \( k \) is the global coordinate. The relation between \( i, j, \) and \( k \) is summed up by the function \( \rho \) such that \( \rho(i,j) = k \). The inverse function \( \rho^{-1} \) is given by \( \rho^{-1}(i,k) = j \). (Bottom row) For each pixel \( k \) in the second image, there is an associated window \( W^{-1}_k \) in the first image. This window is determined by the permitted pixel positions from which \( k \) can receive earth.

**Figure 1. Neighborhood notation for local EMD.**

### 5. Simulation results

To test the effectiveness of LEMD, we simulated random 1D images of different size. The time to solve the linear programming problem as well as the accuracy of approximation were calculated. Let \( T(n) \) and \( T_l(n) \) be the time taken for a random 1D image of size \( n \) for EMD and LEMD respectively. Let \( D(n) \) EMD value and \( D_l(n) \) be the LEMD distance. The speedup is given by \( T(n)/T_l(n) \) and the error by \( (D(n) - D_l(n))/D(n) \). We use \( \alpha = 10 \) and \( \alpha = 40 \) in all the simulations. The simulations were performed for 20 runs for each parameter setting and the averages are reported in figure 2. The simulations were done using Matlab.

We also performed the warping of faces the ORL face database [2]. This database consists of \( 92 \times 112 \) gray scale images of 40 subjects. There are 10 images per person. The size and complexity of linear programming prohibits systematic testing on all database images. Hence the images were subsampled before applying warping. The results are shown in figure 3.

### 6. Conclusion

In this paper, we have proposed the calculation of EMD taking into account locality constraints. LEMD has fewer variables and hence is considerably faster than EMD. LEMD also permits the solution of larger linear programs. Though speedup factors are application independent, approximation properties are problem dependent. In this paper, we have shown approximations errors of 1% to 10%. We have also shown the use of EMD and LEMD for face warping. Though LEMD facilitates the use of larger images for warping, the allowable image sizes are still small. Hence the warping map can be used as a starting point for alignment using other techniques like dynamic programming or optical flow.

### References

The figures show speedup and approximation error (in %) for $\alpha = 10$ and $\alpha = 40$. The size of the local window used is 9. The 1D images were randomly generated. The image size is plotted on the X-axis. It can be seen that LEMD is 0.1% worse for $\alpha = 10$ and around 11% worse for $\alpha = 40$. The speedup is independent of $\alpha$. The numbers plotted are averages of 20 runs.

Figure 2. Speedup and accuracy of approximation.

(a) Original images ($64 \times 64$) obtained by cropping two ORL images. (b) Subsampled images ($16 \times 16$). The subsampling was necessary to perform warping using available computational resources. (c) Direction of earth movement and the reconstructed (warped) image. The reconstructed image is obtained from the second image – moving the pixels according to the flow field. Size of local window is $9 \times 9$. It can be seen that the eyes and noses have moved to the corresponding position maintaining their orientation. (d) Synthesis of second image from first image using the flow field shown on the left.

Figure 3. Face warping using local EMD.