

# Using Discriminative Component Analysis to Learn Descriptors for Improved Shape Mapping

Andy Nguyen, Charlie Camp

Shape matching is an important problem in computational geometry with applications in many areas ranging from animation to medical imaging. The objective of the shape matching problem is to find a correspondence between all points on two given shapes. One general approach to this problem is to compute geometric descriptors on every point on both shapes, and then match the points based on the descriptors, subject to additional continuity constraints. Unfortunately, there are many different descriptors that have been proposed, and all of them work well in some cases but poorly in others. In this project we will obtain training data of shapes with known correspondences, and then compute the descriptors on these sets. We then use metric learning to determine the appropriate weight given to each descriptor, based on how well they match up with the known correspondences. Once the weights have been learned, we can feed them into an existing general framework for shape matching and evaluate our results there. For this last step, we compare the shapes using the eigenbasis of the Laplace-Beltrami operator by requiring that the map be function-preserving for each descriptor function fed in. While the Laplace-Beltrami basis is not a particularly good choice for excessively different shapes, it is a popular choice due to its performance on near-isometric shapes, as well as its ability to be truncated with limited signal loss, much like the Fourier basis.

## I. GENERAL PROCEDURE FOR SHAPE MATCHING

For this problem, we have many similar 3-dimensional shapes that are essentially the same object in different positions. The goal of shape mapping is to find which points on one mesh correspond to which points on another mesh. The shapes are described by 3-d triangle meshes, which consist of a set of points in 3-dimensions, as well as a set of edges that connect the points. Where necessary, a direction around each triangle can be determined by the order in which the edges are listed.<sup>3</sup> These meshes tend to contain about 12,000 points.

for each point in the mesh, we wish to create a vector containing all descriptors for that point. The most important descriptors we use are intrinsic descriptors such as the Wave Kernel Signature (WKS)<sup>4</sup>, Heat Kernel Signature (HKS)<sup>2</sup>, Gaussian Curvature, and mean curvature. The Wave Kernel Signature and Heat Kernel Signature are measured by initializing a delta function signal at the point, then measuring the value at that point while the signal propagates along the mesh surface according to the Wave Equation and Heat Equation, respectively. WKS and HKS tend to make up the majority of our descriptors, since we use the values at many different time-steps, where each time-step is treated as a distinct descriptor.

To match two meshes, we create a  $m \times n$  descriptor matrix, where  $m$  is the number of points in the mesh and  $n$  is the number of descriptors. The next step would be to match the rows of two such matrices using a least-squares method. However, since the meshes in use tend to contain on the order of 12,000 points, we first change to a much smaller Laplace - Beltrami basis and perform least-squares matching with respect to the new basis functions.

## II. USING DISCRIMINATIVE COMPONENT ANALYSIS (DCA) TO IMPROVE MAPPING

The main contribution of our project is to apply Discriminative Component Analysis to the above matching process. When the meshes are matched using their descriptor vectors, each descriptor is given an equal weight regardless of its effectiveness. We can achieve a much better matching by applying metric learning through the use of DCA. The point of DCA is to create optimal descriptor vectors by weighting and mixing the existing descriptors. We are also able to decrease the number of resulting descriptor dimensionality when using DCA. Below we describe the process used:

We begin by choosing a small subset of points called salient points, which mark the same landmark features in each mesh. For example we might choose the salient point 37 for each mesh to be the tip of the nose, while point 2 might be chosen to be the left knee. In our project we choose 40 such points. Our set of training data will then be the descriptor vectors of all of these points from all meshes, where we refer to all such vectors from a particular salient point as a chunklet.

Some chunklets are said to be discriminated from each other by negative constraints. We will aim to transform descriptor vectors such that those from discriminated chunklets are as distinct as possible and those from the same chunklet are as similar as possible. Note that not all chunklets are negatively constrained, in particular features that

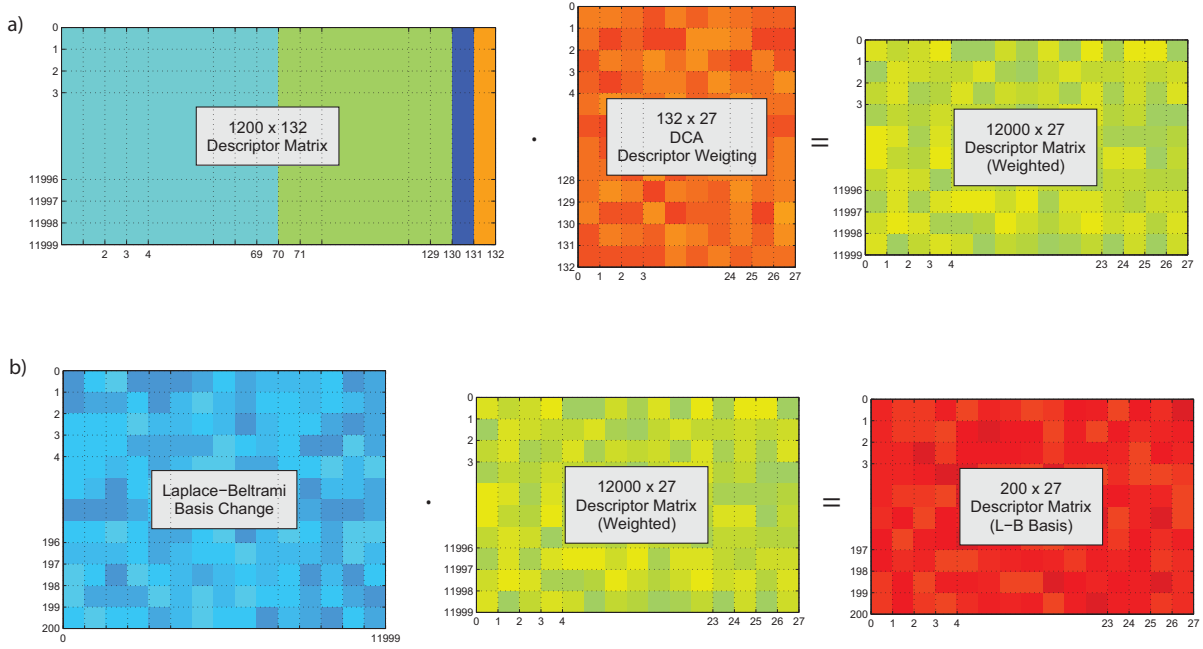


FIG. 1: a) Begin with a  $m \times n$  descriptor matrix, with  $m$  points in the mesh and  $n$  descriptors per point. The teal columns correspond to WKS descriptors at different time-steps, the green columns correspond to similar HKS descriptors, and the last two columns correspond to the two curvature descriptors. By multiplying by the calculated DCA matrix, we reduce the descriptor space to 27 optimized descriptors. b) To find correspondence mapping, we first change basis to the eigenstates of the Laplace-Beltrami operator. Once in this basis we can match the descriptor matrix to that of another mesh, then transform back to the point basis to get a correspondence mapping.

occur symmetrically in right / left pairs.

The DCA algorithm works by creating two matrices,  $C_b$  and  $C_w$ , which represent the variance between discriminated chunklets and the variance among descriptors of the same chunklet respectively,<sup>5</sup>

$$C_b = \frac{1}{n_b} \sum_{j=1}^n \sum_{i \in D_j} (m_j - m_i)(m_j - m_i)^T \quad (1)$$

$$C_w = \frac{1}{n} \sum_{j=1}^n \frac{1}{n_j} \sum_{i=1}^{n_j} (x_{ij} - m_j)(x_{ij} - m_j)^T \quad (2)$$

The matrix in equation (1) measures the average variance between negatively constrained chunklets.  $m_j$  and  $m_i$  are the mean descriptor vectors of chunklets  $i$  and  $j$ , where  $j$  sums over all chunklets and  $i$  sums over all chunklets negatively constrained to  $j$ . The matrix in equation (2) measures the average variance among points within a chunklet.

In the DCA algorithm, we first diagonalize (1) to end up with a new basis that consists of the largest-valued eigenvectors of (1). This represents the basis that best discriminates the chunklets. We transform (2) to this basis, then diagonalize the resulting matrix. By taking the lowest-value eigenvectors we obtain a transformation that maximizes variance across discriminated chunklets, and minimizes variance within a given chunklet. In this process it is common to choose only a subset of the eigenvectors at each step, resulting in a descriptor space with much lower dimensionality. In our project, we are able to reduce the dimensionality of the descriptor space from roughly 130 to about 25.

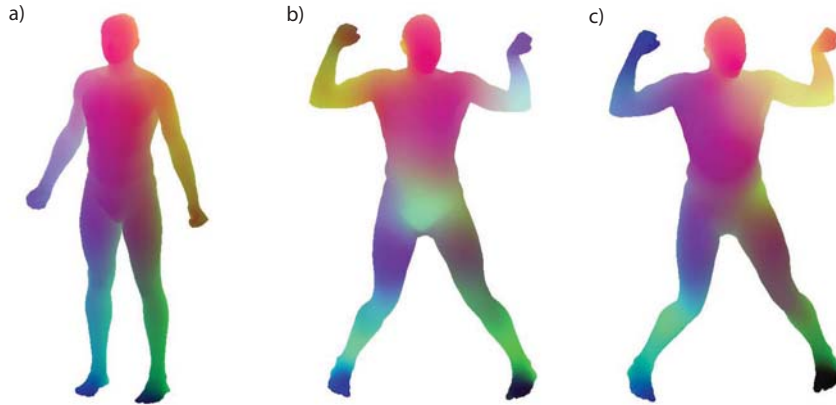


FIG. 2: Visual comparison of mapping between two meshes. a) Reference mesh. b) Mapping calculated using raw descriptor vectors. c) Mapping calculated using DCA weighting. The case with DCA weighting has noticeably better correspondence, with particular attention to the pelvic region and elbows.

The end result From DCA we acquire the matrix  $A$  that transforms our raw descriptor vectors to the optimal learned descriptor vectors. This can then be used for comparing two unknown test meshes by transforming the descriptor vectors for the entire set of points. See FIG. 1.

### III. FUNCTIONAL MAPS

Once we have two meshes with weighted descriptor matrices (in the Laplace-Beltrami basis), we can find the best correspondence through the use of Functional Maps. From the above methods we should know the descriptor functions best preserved by a good map. We can use these "probe functions" as constraints to find a map between basis functions on the two shapes<sup>1</sup>. This then become a least squares problem:

$$\min_M \|F - MG\| \quad (3)$$

Here  $F$  is a probe function written in the chosen basis (Laplace-Beltrami) on one mesh, and each corresponding column of  $G$  is the same probe function written in the chosen basis on the other mesh.

### IV. RESULTS

Mappings between test meshes were calculated both with and without DCA descriptor weighting. The average error for each mapping was calculated as the average euclidean distance of each calculated point mapping to the correct corresponding point. The error was consistently lower across the board when using DCA vs. using the raw descriptors.

Average error with and without DCA with different descriptor choices:

Using only WKS and HKS as descriptors:	Training sets: error = 0.263 without DCA, 0.200 with DCA Testing sets: error = 0.238 without DCA, 0.1915 with DCA
Using curvature with WKS and HKS:	Training sets: error = 0.3123 without DCA, 0.198 with DCA Testing sets: error = 0.3036 without DCA, 0.189 with DCA

To get an idea of the scale of the above error, the figures were approximately 2 units tall. Note that when curvature values (both Gaussian and mean curvature) were used as descriptors, the error increased noticeably. This is not unexpected, since adding certain descriptors has the possibility of increasing minimum variance within chunklets.

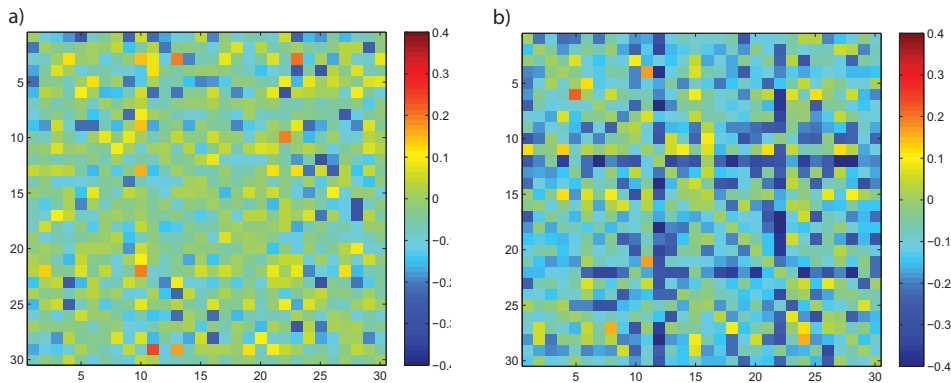


FIG. 3: Matrices containing relative error for all shape mappings amongst 30 test cases.  $ij^{th}$  entry corresponds to  $(err_{weighted} - err_{raw})$  value resulting from mapping mesh  $i$  to mesh  $j$ . Negative entries correspond to mappings where use DCA descriptor weighting resulted in a better mapping. a) Error comparison where curvature is not included. b) Error comparison where curvature is included. Note that each pixel corresponds to a comparison like in FIG. 2

Curvature in particular is known for producing this result, since the curvature of certain salient points can change drastically with different poses.

The important observation here is that the DCA learning algorithm deals with the curvature appropriately. The curvature descriptors are properly incorporated into the learned descriptors in such a way to further decrease the error, as compared even to the DCA-adjusted case without curvature. Another observation is that adding the extra 2 curvature descriptors results in 2 additional learned descriptors. This is not surprising, since curvature should contain very different information from the WKS and HKS descriptors, while many of the individual WKS and HKS descriptors are similar and will be combined with each other in the DCA learning step.

### Acknowledgments

We would like to thank Justin Solomon for providing us with his implementation of the functional maps framework.

- 
- [1] M. Ovsjanikov, M. Ben-Chen, J. Solomon, A. Butscher, L. Guibas, *Functional Maps: A Flexible Representation of Maps Between Shapes* ACM Transactions on Graphics **31**, 4 (2012).
  - [2] Jian Sun, Maks Ovsjanikov, Leonidas Guibas. *A Concise and Provably Informative Multi-scale Signature Based on Heat Diffusion*, **Eurographics Symposium on Geometry Processing (SGP)**.
  - [3] D. Anguelov, P. Srinivasan, D. Koller, S. Thrun, J. Rodgers, J. Davis, *SCAPE: shape completion and animation of people* ACM Transactions on Graphics **24**, 408-416 (2005).
  - [4] M. Aubry, U. Schlickewey, D. Cremers. *The wave kernel signature: A quantum mechanical approach to shape analysis*, **Computer Vision Workshops (ICCV Workshops), 2011 IEEE International Conference on**, 1626-1633 (2011)
  - [5] Steven C. H. Hoi, Wei Liu, Michael R. Lyu, Wei-ying Ma. *Learning distance metrics with contextual constraints for image retrieval*, **Proc. Computer Vision and Pattern Recognition**, 2072-2078 (2006)