Sensor fusion by diffusion maps

Yosi Keller,* Stéphane Lafon,* Ronald R. Coifman† and Steven W. Zucker*

Abstract

Data fusion and the analysis of high-dimensional multisensor data, are fundamental tasks in many research diciplines. In this work we propose a unified embedding scheme for multi sensory data, which is based on the recently introduced diffusion framework. Our scheme is purely data-driven and assumes no a-priory knowledge of the underlying statistical or deterministic models of the different data sources. Our approach is based on embedding separately each of the input channels and combining the resulting diffusion coordinates. In particular, as different sensors samples similar phenomena with different sampling densities, we apply the density invariant Laplace-Beltrami embedding. This is a fundamental issue in multisensor acquisition and processing, overlooked in prior approaches. In order to verify the efficiency of our approach, we apply it to multisensory statistical learning and clustering applications, such as spoken-digit recognition and multi-cue image segmentation. For both applications we experimentally show that using the unified multisensor embedding, allow better performance than the one achieved by any single sensor.

1 Introduction

The first task performed by any data processing system is data acquisition or sampling, in which measurements are collected through a number of sensors. In this work, we refer to a *sensor* as any information stream produced by an acquisition device or, more generally, any descriptor used to

*Google Inc., stephane.lafon@gmail.com

†Department of Mathematics, Yale University, {coifman-ronald, yosi.keller, steven.zucker}@yale.edu.

represent some form of data. Single-sensor systems, which process data coming from a unique information channel, have been successfully used in various context ranging from object recognition (e.g. Sonar) to the medical area (e.g. blood pressure sensors). However, it was early recognized that these systems typically suffer from incompleteness due to the fact that a single sensor is almost never sufficient to capture all of the relevant information related to a phenomenon. For instance, in medical imaging different sensors, such as X-Ray, CT, MRI and others, capture different physical properties. This issue was further studied in the context of remote-sensing (SAR, FLIR, IR and optical sensors). In particular, different sensors are subject to different limitations restricting their usability. For example, in remote sensing, optical sensors have significantly better resolution and lower SNR than Radar based SAR sensors, yet SAR sensors are immune to atmospheric conditions and can be used in any weather conditions. The multisensor approach allows to resolve ambiguities and reduce uncertainties that may arise in some situations, such as object recognition. For example, consider the work by Kidron et. al. [1] who detected image pixels within a video sequence that were related to the creation of sound, given the visual and audio data. Using only the visual data was insufficient as some of the motions in a scene were unrelated to the sound creation.

Note also that many living species rely heavily on a multisensor approach (most humans can see, hear, taste...). In particular, the fusion of audio-visual cues was shown to enhance perception [2, 3]. Last, it is often more cost-efficient to combine a variety of cheap sensors rather than to deal with an expensive single sensor.

The use of high-dimensional multisensor signals requires several tasks. First, the signals have to be embedded in a low-dimensional space that recovers the underlying manifold. When the different data sources are not synchronized and have to be aligned, this manifold can also be used for alignment [4]. In particular, as different sensors might sample the same phenomenon with different densities, the alignment requires density-invariant embeddings. In contrast, most eigenmap representations [5, 6, 7, 8] depend on the density of the points on the underlying manifold, and might be inapplicable for multisensor data integration.

A second task is the alignment and synchronization of different multisensor sources. This was extensively studied in the remote sensing and medical imaging communities. In such applications, due to the different physical characteristics of various imaging sensors, the relationship between the intensities of matching pixels is often complex and unknown a priori. The common approach to multisensor image alignment is to compute canonical representations of image features, which are invariant to the dissimilarity between the different sensors and capture the essence of the image. Theses representations include geometrical primitives such as feature points, contours and corners [9, 10, 11].

Graph theoretical schemes were also applied to this problem. A general purpose approach to high-dimensional data embedding and alignment was presented by Ham et. al [12]. Given a set of corresponding points in the different input channels, a constrained formulation of the graph Laplacian embedding is derived. First, they add a term fixing the embedding coordinates of corresponding points to predefined values. Both sets are then embedded separately, where certain samples in each set are mapped to the same embedding coordinates. Second, they describe a dual embedding scheme, where the constrained embeddings of both sets are computed simultaneously, and the embeddings of certain points in both datasets are constrained to be identical.

Kidron et.al [1] applied canonical correlation analysis to multisensor event detection. Their approach uses a parametric form of the covariance matrices to compute maximally correlated one-dimensional embeddings of the audio and video input signals. A sparsity constraint was applied to regularize the otherwise underconstrained embedding problem, where the constraint corresponds to the sparsity of the detected events.

There is also a large body of literature in engineering related to multisensor integration. These approaches can be classified into three categories [13]. First, some techniques are based on physical models of the data, as in the case of Kalman filtering. Another category corresponds to methods employing a parametric model of the data or the sensors. For instance this is the case of Bayesian inference, of the Dempster-Shafer method or Neural Networks. Such techniques often exhibit high sensitivity to the accuracy of these models [14]. The third group consists of cognitive-based methods, which aim at mimicking human inference. One of the main tools is fuzzy logic. But there again, one needs to specify subjective membership functions. It therefore appears that many of these techniques rely on prior information.

A problem related to data fusion is the fusion of multiple partitionings [15]. The focal point there is to fuse together different *partitionings*, rather than different data *sources* as in the general data fusing problem. This approach boils down to embedding the data in a one-dimensional space (the partitioning index). As this is not a metric space, a distance metric can not be defined and the work in [15] uses the co-association matrix as a binary similarity measure.

A related problem was recently studied within the computer vision community in the context of multi-cue image segmentation. These works are of particular interest, as (similar to our approach) they are based on spectral embeddings [16]. In [17] Yu presents a segmentation scheme that integrates edges detected at multiple scales. These were shown to provide complementary segmentation cues. Given the affinity matrices computed using the edges at each scale, a simultaneous segmentation is computed using a novel criterion called average cuts. Other works [18, 19], deal with the fusion of a single multiscale cue in images and can be applied directly to multisensor data

In this work we derive a unified low-dimensional representation, given a set of different input channels related to a particular phenomenon. We assume that the input signals are aligned and derive a unified representation of them, useful for statistical learning tasks and data partitioning. We compute a unified low-dimensional representation and show that it combines the information encoded in the different signals, thus, improving the parametrization and analysis of complex phenomena. We start by computing low-dimensional embeddings of each of the input signals using the diffusion framework [20, 21] and for that utilize the Laplace-Beltrami density invariant scheme [22]. The proposed scheme was first applied to statistical learning by recognizing spoken digits using audio and visual cues. We compare the results to our previous work in visual-only lip-reading [4], and show improved accuracy. Then, we turn our attention to multi-cue image segmentation, where the multisensor data is related to different image cues: RGB, contours and texture. Compared to prior works, the proposed approach does not require any deterministic model of the data or its statistics (covariance matrices etc.), and the recovered structures are purely data-driven. In particular, we resolve the density-dependence issue of the embeddings that was largely overlooked in prior works.

This paper is organized as follows: We describe the foundations of the diffusion based embeddings and introduce the unified, fused multisensor embedding in Section 2. Our scheme is then experimentally verified in Section 3, while concluding remarks and future extensions are discussed in Section 4.

2 Multi-sensor integration

In this section we present the proposed data fusion scheme. We start by describing low-dimensional spectral embeddings and then extend them to derive the density-invariant Laplace-Beltrami embedding. A more detailed description can be found in [4], and the mathematical foundations are given in [22]. Given a set $\Omega = \{x_1, ..., x_n\}$ of data points, we start by constructing a weighted symmetric graph where each data point x_i corresponds to a node. Two nodes x_i and x_j are connected by an edge with weight $w(x_i, x_j) = w(x_j, x_i)$ reflecting the degree of similarity (or affinity) between these two points. The weight function $w(\cdot, \cdot)$ describes the first-order interaction between the data points and its choice is application-driven. For instance, in applications where a distance $d(\cdot, \cdot)$ already exists on the data, it is custom to weight the edge between x_i and x_j by $w(x_i, x_j) = \exp(-d(x_i, x_j)^2/\varepsilon)$, where $\varepsilon > 0$ is a scale parameter, while other weight functions can also be used.

Following a classical construction in spectral graph theory [23] and manifold learning [24], namely the normalized graph Laplacian, we now create a random walk on the data set Ω by forming the kernel

$$p_1(x_i, x_j) = \frac{w(x_i, x_j)}{d(x_i)},$$

where $d(x_i) = \sum_{x_k \in \Omega} w(x_i, x_k)$ is the degree of node x_i. As we have that $p_1(x_i, x_j) \geq 0$ and $\sum_{j \in \Omega} p_1(x_i, x_j) = 1$, the quantity $p_1(x_i, x_j)$ can be interpreted as the probability of a random walker to jump from x_i to x_j in a single time step [23, 25]. Let P be the $n \times n$ matrix of transition of this Markov chain, then taking powers of this matrix amounts to running the chain forward in time. Let $p_t(\cdot, \cdot)$ be the kernel corresponding to the t^{th} power of the matrix P. Then, $p_t(\cdot, \cdot)$ describes the probabilities of transition in t time steps. The essential point of the diffusion framework is the idea that running the chain forward will reveal *intrinsic geometric structures* in the data set, and taking powers of the matrix P is equivalent to integrating the local geometry of the data at different scales.

An equivalent way to look at powers of P is to make use of its eigenvectors and eigenvalues: it can be showed that there exists a sequence $1 = \lambda_0 \geq |\lambda_1| \geq |\lambda_2| \geq ...$ of eigenvalues and a collection $\{\psi_0, \psi_1, \psi_2, ...\}$ of (right) eigenvectors for P:

$$P\psi_l = \lambda_l \psi_l \, .$$

These eigenvalues and eigenvectors provide embedding coordinates for the set Ω. The data points can be mapped into a Euclidean space via the embedding

$$\Psi_t : x \longmapsto \left\langle \lambda_1^t \psi_1(x), \ldots, \lambda_{m(t)}^t \psi_{m(t)}(x) \right\rangle \, , \tag{2.1}$$

where $t \geq 0$. Discussions regarding the number $m(t)$ of diffusion coordinates to employ and concerning the connection with the so-called diffusion distance are provided in [22, 26, 27].

Next, we address the issue of obtaining a density-invariant embedding. The focal point is to compute an embedding that reflects only the geometry of the data and is insensitive to the sampling density of the points. Classical eigenmap methods [5, 6, 7, 28], provide an embedding that combines the information of both the density and geometry, and the embedding coordinates heavily depend on the density of the data points. In order to remove the influence of the distribution of the data points, we renormalize the Gaussian edge weights $w_\varepsilon(\cdot, \cdot)$ with an estimate of the density. This is summarized in Algorithm 1 that was first introduced and analyzed in [22].

Next we describe the data fusion scheme, where, for the sake of clarity, we direct our discussion to the case of two input channels, while it can be easily extended to an arbitrary number of them. Suppose one has two sets of measurements related to a particular phenomenon $\Omega = \{x_1, ..., x_n\}$. Denote these sets of measurements $\Omega_1 = \{y_1^1, ..., y_n^1\}$ and $\Omega_2 = \{y_1^2, ..., y_n^2\}$, respectively, where y_i^1 and y_i^2 are high-dimensional measurements. We aim to fuse Ω_1 and Ω_2 by computing a unified low-dimensional representation $\widehat{\Omega} = \{z_1, ..., z_n\}$. Note that we assume that $\Omega_1$and Ω_2 are aligned, meaning that y_i^1 and y_i^2 relate to the same instance $x_i \in \Omega$. When this assumption is invalid, one has to apply a multi-sensor alignment scheme [12] prior to applying the fusion procedure.

We start by computing the Laplace-Beltrami embeddings of Ω_1 and Ω_2 denoted $\Phi_1^{m_1} = \{\phi_1^1, ..., \phi_n^1)\}$ and $\Phi_2^{m_2} = \{\phi_1^2, ..., \phi_n^2\}$, respectively, where m_i is the dimensionality of each embedding. Thus, ϕ_i^1

Algorithm 1 Approximation of the Laplace-Beltrami diffusion

1: Start with a rotation-invariant kernel $w_\varepsilon(x_i, x_j) = h\left(\frac{\|x_i - x_j\|^2}{\varepsilon}\right)$.

2: Let

$$q_\varepsilon(x_i) \triangleq \sum_{x_j \in \Omega} w_\varepsilon(x_i, x_j)\,,$$

and form the new kernel

$$\widetilde{w}_\varepsilon(x_i, x_j) = \frac{w_\varepsilon(x_i, x_j)}{q_\varepsilon(x_i) q_\varepsilon(x_j)}\,. \tag{2.2}$$

3: Apply the normalized graph Laplacian construction to this kernel, *i.e.*, set

$$d_\varepsilon(x) = \sum_{z \in \Omega} \widetilde{w}_\varepsilon(x_i, x_j)\,,$$

and define the anisotropic transition kernel

$$p_\varepsilon(x_i, x_j) = \frac{\widetilde{w}_\varepsilon(x_i, x_j)}{d_\varepsilon(x_i)}\,.$$

---

and ϕ_i^2 are of vectors dimensions m_1 and m_2, respectively. In order to combine these embeddings into a unified representation $\widehat{\Omega}$, we form $\widehat{\Omega} = \{z_1, ..., z_n\}$ where

$$z_i = \{\phi_i^1, \phi_i^2\}, \tag{2.3}$$

and z_i is of dimension $m + m_2$. In general, given K input sources we have

$$z_i = \{\phi_i^1, \ldots, \phi_i^K\}. \tag{2.4}$$

This boils down to combining the embedding coordinates corresponding to each sample x_i over the different input channels $\{\Omega_i\}$.

In essence, our scheme is the embedding analogue of boosting [29], where instead of adaptively integrating the output of several classifiers, we combine different embeddings. In particular, one can consider an equivalent to the *AdaBoost* scheme [29] for semi-supervised classification, where Eq. 2.4 can be replaced with

$$\widehat{z}_i = \{a^1 \phi_i^1, \ldots, a^K \phi_i^K\}, \tag{2.5}$$

$\{a^1, \ldots, a^K\}$ being the weights per embedding. In that sense, the embeddings z_i can be considered as different features, and one can apply a standard implementation of *AdaBoost* to Eq. 2.4. Yet, in this work, the focal point is to derive general-purpose coordinates regardless of a particular application. The scheme is summarized in Algorithm 2.

Algorithm 2 Multisensor embedding

1: Starting with K input sources $\Omega_k = \{y_1^k, ..., y_n^k\}$, $k = 1...K$.

2: Compute the Laplace-Beltrami embeddings of $\{\Omega_k\}$, denoted $\Phi_k^{m_k}$, where m_k is the dimensionality of the embedding of the k'th input channel.

3: Compute the unified coordinates set $\widehat{\Omega} = \{z_1, ..., z_n\}$ by appending the embeddings of each input sensor

$$z_i = \{\phi_i^1, \ldots, \phi_i^K\}, i = 1...n, \ k = 1...K.$$

3 Experimental results

The proposed scheme was experimentally verified by applying it to two tasks. First, we extend our former results in visual-only lip-reading [4] to audio-visual data. The audio-visual inputs are integrated using the multisensor fusion scheme given in Section 2 and used for spoken-digit recognition. We show that the fused multi-sensor representation provides better recognition rates. Second, we integrate several image cues (texture, RGB values, contours etc.) and show that using them in conjugation improves the segmentation results.

3.1 Spoken-digit recognition

We start by providing a short description of the experimental setup. We follow the statistical learning scheme used in [4], where the classifier was constructed in two steps. First we parameterized the embedding manifold using a large number of unlabeled samples. The embedding is then extended, using the Geometric Harmonics [4, 30], to a small set of labeled examples to create a set of *signatures* in the embedding coordinates. Then, given a test sample, we embed it by extending the manifold embedding, and find the nearest signature in the embedding space.

To this end, we recorded several grayscale movies depicting the lips of a subject reading a text in English and retained both the video sequence and the audio track. Each video frame was cropped into a rectangular of size 140×110 around the lips and was viewed as a point in $\mathbb{R}^{140\times 110}$. As far as the audio data was concerned, the sound signal was broken up into overlapping time-windows centered at the beginning of each video frame.

The video was sampled at 25 frames per second, so we divided the audio into overlapping windows of a duration of 8ms. The overlap was of 4ms, thus, each video frame corresponded to a particular (unique) audio window. In order to reduce the influence of this splitting, each audio window was multiplied by a bell-shaped function, and we then computed the DCT of the result. Last, we considered the logarithm of the magnitude of this function as being the audio features.

In terms of Section 2 and Algorithm 2, the set Ω_1 is the audio samples, while Ω_2 is the set of video frames. Note that in this applications Ω_1 and Ω_2 are naturally aligned and contain the same number of points.

The first data set (audio and visual) consisted of 6000 video frames (and as many audio windows), corresponding to the speaker reading a press article. We will refer to this data as “text data”. Next, we asked the subject to repeat each digit “zero”, “one”, ... , “nine” 40 times. This was used to construct a small vocabulary of words later employed for training and testing a simple classifier. Each spoken digit corresponded to a sequence of frames in the video data, and a sequence of time-windows for the audio data. We will refer to this data as “digit data”. We proceeded as follows for each channel: first, the data points corresponding to the text data were used to learn the geometry of speech data as we formed a graph with Gaussian weights $\exp(\frac{\|x_i - x_j\|^2}{\varepsilon})$ on the edges, for an appropriately chosen scale $\varepsilon > 0$. We then renormalized the Gaussian weights using the Laplace-Beltrami normalization described in Algorithm 1. In order to obtain a low-dimensional parametrization we computed the diffusion coordinates on this new graph. Therefore we ended up with two embeddings, $\Phi_1^{m_1}$ and $\Phi_1^{m_2}$, corresponding to the audio and visual data.

The next step involved the digits data. We computed the diffusion coordinates for all of the samples in the digits data, by applying the Geometric Harmonics scheme [4, 30] and extending the diffusion coordinates computed on the text data.

In order to train a classifier for digit identification, we randomly selected 20 sequences of each digit, the remaining sequences being used as a test set. Each digit word can now be viewed as a

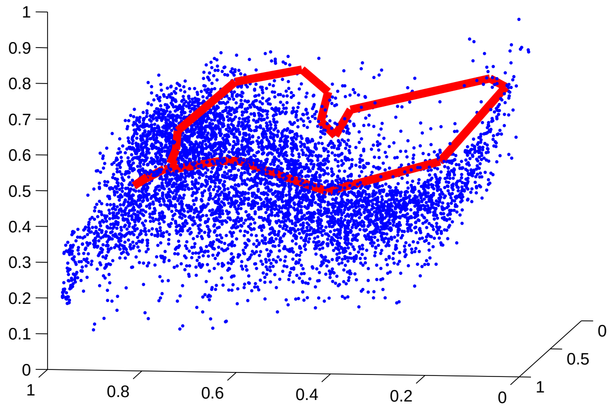


Figure 1: The visual data in the first 3 diffusion coordinates. We also represented a trajectory corresponding to an instance of the word "one".

set of points in the diffusion space, and the word recognition problem now amounts to identifying the most similar set in the diffusion space (see Fig. 1). We can now build a classifier based on comparing a new set of points to a collection of labeled sets in the training set. In order to compare sets in the diffusion space we used the symmetric Hausdorff distance between two sets Γ_1 and Γ_2, defined as

$$d_H(\Gamma_1, \Gamma_2) = \max\left\{ \max_{x_2 \in \Gamma_2} \min_{x_1 \in \Gamma_1} \{\|x_1 - x_2\|\}, \max_{x_1 \in \Gamma_1} \min_{x_2 \in \Gamma_2} \{\|x_1 - x_2\|\} \right\}. \tag{3.1}$$

As the Hausdorff distance overlooks the difference in dynamics between Γ_1 and Γ_2, it is robust to sampling rate changes. This is essential in speech recognition, where even the same speaker, might pronounce the same words at different speeds.

The recognition rates of this classifier for the visual-only data were already reported in [4], where 15 eigenvectors were used for embedding. Hence, we re-ran this experiment with 10 eigenvectors, and the results are shown in Table 1. Similarly, the classification rates corresponding to the audio-only data, and 10 eigenvectors are presented in Table 2.

	“0”	**“1”**	**“2”**	**“3”**	**“4”**	**“5”**	**“6”**	**“7”**	**“8”**	**“9”**
zero	**0.90**	0	0	0.01	0	0	0.08	0	0	0
one	0	**0.99**	0	0	0	0	0	0	0.01	0
two	0.04	0.01	**0.90**	0.03	0.02	0	0	0	0	0
three	0	0	0.01	**0.94**	0	0	0.01	0.02	0.01	0
four	0.01	0	0	0.05	**0.93**	0	0	0	0	0
five	0	0	0	0	0	**0.81**	0.01	0.16	0	0.01
six	0.07	0	0	0.01	0	0	**0.87**	0.03	0.01	0.01
seven	0.03	0	0	0.04	0	0.07	0.05	**0.74**	0.04	0.02
eight	0	0	0	0	0.02	0.03	0	0.03	**0.75**	0.16
nine	0	0	0	0	0	0	0	0.04	0.14	**0.82**

Table 1: Digits recognition rates for a classifier based on the visual-only data. The results are averaged over 50 random trials and the data was embedded onto a 10 dimensional diffusion space. Each row depicts the classification rate of a given digit over then the 10 possible classes (digits).

In order to illustrate the advantage of combining both data channels using the proposed multisensor integration scheme, we present the results obtained using Algorithm 2 (see Table 3). More precisely, we appended the first 5 eigenvectors of the audio data with the top 5 eigenvectors of the video data, and then constructed a new 10 dimensional representation of the data. Finally, as before, a classifier was trained and tested on the unified embedding. A summary of the recognition rates of the different schemes and features is reported in Table 4.

Clearly, the proposed multisensor approach outperformed the single-channel classifiers, exemplifying the achieved synergy. More precisely, it seems to get the best of the predictive powers of the audio and visual classifiers. In fact, this is a straight consequence of the concatenation of the audio and visual diffusion features. For instance, the digit “one” is successfully classified using the visual channel. As suggested in [4], typical frame sequences corresponding to the word “one” con-

	"0"	"1"	"2"	"3"	"4"	"5"	"6"	"7"	"8"	"9"
zero	**0.75**	0	0.04	0	0.01	0.01	0.06	0.08	0.05	0
one	0	**0.94**	0	0	0	0.03	0	0	0	0.02
two	0.02	0	**0.87**	0.04	0.01	0	0.01	0	0.03	0.02
three	0.01	0	0.03	**0.90**	0.02	0.01	0	0	0.01	0.01
four	0.01	0	0	0.02	**0.96**	0	0	0	0	0.01
five	0.01	0.01	0	0.06	0	**0.86**	0	0.01	0.01	0.03
six	0	0	0	0	0.01	0	**0.93**	0.05	0	0
seven	0.05	0	0	0	0	0	0.14	**0.81**	0.01	0
eight	0.02	0	0.04	0.02	0	0.02	0	0.07	**0.80**	0.03
nine	0	0.01	0	0.01	0.01	0.04	0	0	0.01	**0.92**

Table 2: Digits recognition rates for a classifier based on the audio-only data. The results are averaged over 50 random trials and the data was embedded onto a 10 dimensional diffusion space. Each row depicts the classification rate of a given digit over then the 10 possible classes (digits).

tain pictures with an open mouth and no visible teeth. This type of frame almost never appears in the other digit sequences. As a consequence, trajectories for the word "one" will be well separated from other digit trajectories in the visual diffusion space. In contrasts, for the audio-only features, the separation is slightly less evident, and there is some confusion with "five" and "nine". When appending cues, the separation remains high. Notice also that these accurate results were obtained despite the fact that we used only 5 eigenvectors from each channel in the combined scheme, and not 10 as in the single channel results (Tables 1 and 2).

3.2 Image segmentation

The sensor fusion scheme was also applied to multi-cue image segmentation. As features we used combinations of Interleaving Contours (IC) [31], the L_2 metric between RGB values and Gabor filters based texture descriptors [32]. The Gabor filters used 3 scales and 8 orientations. For each

	"0"	"1"	"2"	"3"	"4"	"5"	"6"	"7"	"8"	"9"
zero	**0.90**	0.00	0.00	0.00	0.00	0.00	0.06	0.04	0.00	0.00
one	0.00	**0.99**	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.01
two	0.00	0.00	**0.96**	0.01	0.02	0.00	0.00	0.00	0.00	0.00
three	0.00	0.00	0.00	**0.99**	0.00	0.00	0.00	0.00	0.00	0.01
four	0.00	0.00	0.00	0.04	**0.96**	0.00	0.00	0.00	0.00	0.00
five	0.00	0.00	0.00	0.00	0.00	**0.97**	0.00	0.00	0.02	0.01
six	0.06	0.00	0.00	0.00	0.00	0.00	**0.90**	0.04	0.00	0.00
seven	0.03	0.00	0.00	0.00	0.00	0.00	0.03	**0.93**	0.00	0.00
eight	0.00	0.00	0.00	0.00	0.00	0.01	0.00	0.01	**0.95**	0.03
nine	0.00	0.01	0.00	0.00	0.00	0.01	0.00	0.00	0.02	**0.96**

Table 3: Classification results for the scheme combining both channels, over 50 random trials. The combined graph was built from a feature representation of the data based on appending the first 5 eigenvectors of the audio channel with the first 5 eigenvectors of the video stream.

pixel, the metrics were computed in an area of 5×5 around it. Such that given a feature $f(i,j)$ computed over the image I, the distance between the pixels $I(i,j)$ and $I(i_1,j_1)$ with respect to the feature is given by

$$D\left(I\left(i,j\right),I\left(i_1,j_1\right)\right)=\begin{cases}\left\|f(i,j)-f(i_1,j_1)\right\|_{L_2} & \sqrt{(i-i_1)^2+(j-j_1)^2}\leq 2\\ \infty & otherwise\end{cases}. \quad (3.2)$$

Equation 3.2 is used to sparsify the affinity matrix of I, otherwise, its eigenvectors computation become computationally exhaustive for common image sizes. Applying Eq. 3.2 might create spurious additional parametrizations related to the spatial coordinates. For instance, consider the vertical and horizontal lines in all of the segmentations in Fig. 2. We refrained from using the Nyström Method [33], that would have resolved this issue, in order to simplify the testing procedure and as this phenomenon is well understood.

For every input image, we computed several embeddings and the integrated representation was computed by the procedure described in Section 2. We emphasize, that for each image, the same

Channel type	**“0”**	**“1”**	**“2”**	**“3”**	**“4”**	**“5”**	**“6”**	**“7”**	**“8”**	**“9”**
Audio	0.75	0.94	0.87	0.90	**0.96**	0.86	**0.93**	0.81	0.80	0.92
Visual	**0.90**	**0.99**	0.90	0.94	0.93	0.81	0.87	0.74	0.75	0.82
Combined	**0.90**	**0.99**	**0.96**	**0.99**	**0.96**	**0.97**	0.90	**0.93**	**0.95**	**0.96**

Table 4: A summary of the classification accuracy of the spoken digits dataset, using different cues and embedding schemes. The audio-visual data combined using the proposed scheme outperforms the single-channel classifiers.

embedding vectors were used both for the single and multi-cue segmentations. In all of the simulations we used 5 eigenvectors from each feature. For all images we present the segmentation results of applying k-means clustering to each of the original embeddings and the fused coordinates. This follows the Modified-NCut (MNCut) image segmentation scheme [34]. The scheme was implemented in Matlab and used the built-in kmeans and SVD implementations. Note that the regular Graph-Laplacian was used for the segmentation and not the density-invariant Laplace-Beltrami.

(a) Interleaving contours (b) RGB (c) Fused coordinates

Figure 2: Applying the proposed scheme to the *Tiger* image. (a) Segmentation computed using the Interleaving contours edge based features. (b) Segmentation results based on L_2 differences in RGB values. (c) Using the fused coordinates we achieve a visually better pleasing result.

Figure 2 depicts the segmentation results of the *Tiger* image taken from the Berkeley segmentation database. The images were segmented using the IC and RGB features and the results are shown Figs. 2a and 2b, respectively. The segmentation in Fig. 2c exemplifies that using the fused coordinates provided better results than using just one of the cues (IC and RGB).

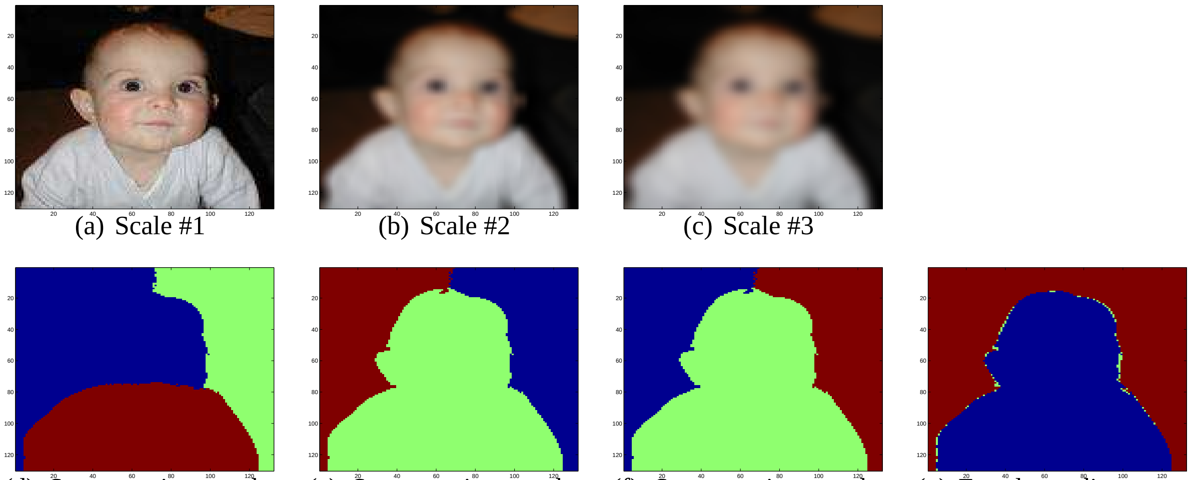
(a) Texture (b) RGB (c) Combined

Figure 3: Applying the proposed scheme to the *Lizard* image. (a) Segmentation achieved using the texture features. Note the over segmentation in the are behind the Lizard's head. (b) Segmentation results based on L_2 differences in RGB values. Note the over-segmentation above the Lizard's leg. (c) Using the fused coordinates we achieve a visually better pleasing result.

Different features were used in Fig. 3. The IC feature is inappropriate for analyzing highly-textured images, as it results in over-segmentation. Thus, we used the RGB and texture features. The texture based segmentation (Fig. 3a) results in over-segmentation of the lizard's body, while overlooking the cut between the front and background rocks on the left side of the image. Similarly, using the RGB descriptor also results in over-segmentation. In contrast, the combined segmentation is better eye pleasing.

Finally, we applied the fusion scheme to multi-scale image segmentation. The different image scales (shown in Figs. 4a-4c) were generated by a Gaussian kernel, the IC feature was computed in each scale, and the multi-scale embeddings were fused using the proposed scheme. This resulted in a segmentation that combined the salient cluster boundaries in the image over the different scales, allowing to overlook some of the spurious single-scale segmentations, such as the left eye in Fig. 4a and the throat area in 4b. In [17] a multiscale segmentation was computed via the computation of an "average cut". There, the single-scale Markov matrices were fused, rather than the embedding vectors. In our scheme, there is no difference between a multiscale fusion and the fusion of any combination of the other cues.

To conclude, by fusing the different image features, we were able to achieve better segmentation results. In essence, this approach resembles biological vision systems by combining different cues and emphasizing salient multi-features edges. The scheme is flexible and once the embed-



(a) Scale #1 (b) Scale #2 (c) Scale #3

(d) Segmentation results at Scale #1 (e) Segmentation results at Scale #2 (f) Segmentation results at Scale #3 (g) Fused coordinates results

Figure 4: Applying the multisensor embedding to multiscale image segmentation. The Interleaving contours edge based feature was applied to each of the image in the first row ((a)-(c)). The second row depicts the corresponding segmentation results. (g) show the improved segmentation achieved using the fused coordinates.

dings of each feature are computed, one can combine the embeddings in any possible way without having to recompute them.

4 Conclusions and future work

In this work we presented a unified multisensor data embedding scheme, based on the diffusion framework. The fusion was achieved by combining the embeddings of different input channels. We applied the scheme to audio-visual lip reading and image segmentation that are typical examples of multisensor pattern recognition and classification. In both cases, the results achieved by using fused coordinates prevailed over those of the single sensor.

We embedded each data source separately and then appended the embeddings to produce the fused representation. Although this approach is straightforward and allows to combine different channels easily, it is possible that different channels are correlated. Then, one can find a lower dimensional representation by considering the unified coordinates as the features of a signal and re-embedding them to further reduce the dimensionality.

The image segmentation results, suggest that in certain applications, one can utilize a variety of features in different resolution scales. Thus, due to the large number of possible input channels, it might be beneficial to compute adaptive weights that maximize a certain criterion. For instance, in semi-supervised classification problems, one can train the weights of the combined representation for optimal classification over a training set by using the *AdaBoost* algorithm.

References

[1] E. Kidron, Y. Schechner, and M. Elad, "Pixels that sound," in *Proceedings, IEEE Conference on Computer Vision and Pattern Recognition*, vol. I, June 2005, pp. pp. 88–96.

[2] J. Driver, "Enhancement of selective listening by illusory mislocation of speech sounds due to lip-reading," *Nature*, no. 381, pp. 66–68, 1996.

[3] Y. Gutfreund, W. Zheng, and E. I. Knudsen, "Gated visual input to the central auditory system," *Science*, no. 297, pp. 1556–1559, 2002.

[4] S. Lafon, Y. Keller, and R. R. Coifman, "Data fusion and multi-cue data matching by diffusion maps," *Accpeted for publication PAMI*.

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

[6] M. Belkin and P. Niyogi, "Laplacian eigenmaps for dimensionality reduction and data representation," *Neural Computation*, vol. 6, no. 15, pp. 1373–1396, June 2003.

[7] D. Donoho and C. Grimes, "Hessian eigenmaps: new locally linear embedding techniques for high-dimensional data," *Proceedings of the National Academy of Sciences*, vol. 100, no. 10, pp. 5591–5596, May 2003.

[8] Z. Zhang and H. Zha, "Principal manifolds and nonlinear dimension reduction via local tangent space alignement," Department of computer science and engineering, Pennsylvania State University, Tech. Rep. CSE-02-019, 2002.

[9] H. Li, B. S. Manjunath, and S. K. Mitra, "A contour-based approach to multisensor image registration," *IEEE Transactions on Image Processing*, vol. 4, no. 3, pp. 320–334, March 1995.

[10] R. Sharma and M. Pavel, "Registration of video sequences from multiple sensors," in *Proceedings of the Image Registration Workshop*. NASA GSFC, 1997, pp. 361–366.

[11] A. Gueziec, X. Pennec, and N. Ayache, "Medical image registration using geometric hashing," *IEEE Computational Science & Engineering, special issue on Geometric Hashing*, vol. 4, no. 4, pp. 29–41, October-December 1997.

[12] J. Ham, D. Lee, and L. Saul, "Semisupervised alignment of manifolds," in *Proceedings of the Tenth International Workshop on Artificial Intelligence and Statistics, Jan 6-8, 2005, Savannah Hotel, Barbados*, pp. 120–127.

[13] E. Waltz and J. Llinas, *Spectral graph theory*. Artech House, Boston, 1990.

[14] J. Sasiadek, "Sensor fusion," *Annual reviews in control*, vol. 26, pp. 203–228, 2002.

[15] A. Fred and A. K. Jain, "Combining multiple clusterings using evidence accumulation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 27, no. 6, pp. 835–850, June 2005.

[16] Y. Weiss, "Segmentation using eigenvectors: A unifying view," in *ICCV '99: Proceedings of the International Conference on Computer Vision*, vol. 2. Washington, DC, USA: IEEE Computer Society, 1999, p. 975.

[17] S. X. Yu, "Segmentation using multiscale cues." in *2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2004), 27 June - 2 July 2004, Washington, DC, USA*, 2004, pp. 247–254.

[18] T. Cour, F. Bénézit, and J. Shi, "Spectral segmentation with multiscale graph decomposition." in *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2005), 20-26 June 2005, San Diego, CA, USA*, 2005, pp. 1124–1131.

[19] S. X. Yu, "Segmentation induced by scale invariance," in *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2005), 20-26 June 2005, San Diego, CA, USA*, vol. 1, 2005, pp. 444 – 451.

[20] R. Coifman, S. Lafon, A. Lee, M. Maggioni, B. Nadler, F. Warner, and S. W. Zucker, "Geometric diffusions as a tool for harmonic analysis and structure definition of data: Diffusion maps," *Proceedings of the National Academy of Sciences*, vol. 102, no. 21, pp. 7426–7431, May 2005.

[21] R. Coifman, S. Lafon, A. Lee, M. Maggioni, B. Nadler, F. Warner, and S. Zucker, "Geometric diffusions as a tool for harmonics analysis and structure definition of data: Multiscale methods," *Proceedings of the National Academy of Sciences*, vol. 102, no. 21, pp. 7432–7437, May 2005.

[22] R. Coifman and S. Lafon, "Diffusion maps," *Applied and Computational Harmonic Analysis*, 2005, to appear.

[23] F. Chung, *Spectral graph theory*. CBMS-AMS, May 1997, no. 92.

[24] M. Belkin, P. Niyogi, and V. Sindhwani, "Manifold regularization: a geometric framework for learning from examples," University of Chicago, Tech. Rep. TR-2004-06, 2004.

[25] J. Shi and J. Malik, "Normalized cuts and image segmentation," *IEEE Tran PAMI*, vol. 22, no. 8, pp. 888–905, 2000.

[26] B. Nadler, S. Lafon, R. Coifman, and I. Kevrekidis, "Diffusion maps, spectral clustering and the reaction coordinates of dynamical systems," *Applied and Computational Harmonic Analysis*, 2005, to appear.

[27] S. Lafon and A. B. Lee, “Diffusion maps: A unified framework for dimension reduction, data partitioning and graph subsampling,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, p. Accpeted for publication, 2005.

[28] M. Balasubramanian, E. L. Schwartz, J. B. Tenenbaum, V. de Silva, and J. C. Langford, “The Isomap algorithm and topological stability,” *Science*, vol. 295, no. 5552, p. 7, 2002.

[29] T. Hastie, R. Ribshirani, and J. H. Friedman, *The elements of statistical learning: data mining, inference and prediction*. Springer, 2002.

[30] R. Coifman and S. Lafon, “Geometric harmonics,” *Applied and Computational Harmonic Analysis*, 2005, to appear.

[31] M. Meila and J. Shi, “Learning segmentation by random walks.” in *NIPS*, 2000, pp. 873–879.

[32] J. Malik, S. Belongie, T. Leung, and J. Shi, “Contour and texture analysis for image segmentation,” *International Journal of Computer Vision*, vol. 43, no. 1, pp. 7–27, 2001.

[33] C. Fowlkes, S. Belongie, F. Chung, and J. Malik, “Spectral grouping using the nyström method.” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 26, no. 2, pp. 214–225, 2004.

[34] M. Maila and J. Shi, “A random walks view of spectral segmentation,” in *AI and STATISTICS (AISTATS) 2001*, 2001.