

Diffusion Processes for Retrieval Revisited

Michael Donoser and Horst Bischof
Institute for Computer Graphics and Vision
Graz University of Technology
{donoser,bischof}@icg.tugraz.at

Abstract

In this paper we revisit diffusion processes on affinity graphs for capturing the intrinsic manifold structure defined by pairwise affinity matrices. Such diffusion processes have already proved the ability to significantly improve subsequent applications like retrieval. We give a thorough overview of the state-of-the-art in this field and discuss obvious similarities and differences. Based on our observations, we are then able to derive a generic framework for diffusion processes in the scope of retrieval applications, where the related work represents specific instances of our generic formulation. We evaluate our framework on several retrieval tasks and are able to derive algorithms that e. g. achieve a 100% bullseye score on the popular MPEG7 shape retrieval data set.

1. Introduction

Retrieving the most similar instances to a provided query element, from a potentially large database, is an important field of research in computer vision with many applications like image search or database retrieval. Traditionally, this is based on analyzing pairwise affinity values, which measure the similarity between two elements. The affinity values are then used to rank the most similar elements first, where obviously performance is mainly defined by the provided pairwise measures. Such a basic retrieval approach has the main limitation that the structure of the underlying data manifold is completely ignored. For this reason, recently there was a growing interest in providing context sensitive similarities, where the derived measures exploit the context between all elements of the database to improve retrieval scores. In general, all these methods have in common, that they aim at capturing the geometry of the underlying manifold.

Most methods in this field follow the same principle. First, the manifold, defined by the provided affinity matrix, is interpreted as a weighted graph, where each element is represented by a node, and edges connect all nodes with corresponding edge weights proportional to the pairwise affin-

ity values. Then the pairwise affinities are re-evaluated in the context of all other elements, by diffusing the similarity values through the graph. The most common diffusion processes are based on random walks, where a transition matrix defines probabilities for walking from one node to a neighboring one, which are fixed proportional to the provided affinities. By repeatedly making random walk steps on the graph, affinities are spread on the manifold, which in turn improves the obtainable retrieval scores.

Let us consider a toy example to illustrate the usefulness of such diffusion processes for retrieval tasks. Figure 1 shows a toy pattern, where two query points are defined, and as the scope of retrieval is to return the most similar examples from the database, all other elements are labeled according to the larger affinity to one of the two query points. As can be seen without diffusion the underlying manifold is not considered and retrieval performance is insufficient. In comparison, after diffusing the similarities through the manifold and capturing the intrinsic global manifold structure, we get significantly improved retrieval results.

In general, diffusion processes iteratively update the provided affinity matrices, until some kind of convergence is achieved. Nevertheless, many approaches have a closed-form solution, where one can directly calculate the result without an iterative process. Unfortunately, closed form solutions require complex operations, like calculating the inverse of huge matrices, which makes them impracticable for large-scale retrieval applications. Therefore, in this paper we mainly focus on iterative diffusion processes, where we further speed up the process, by sparsifying the graph structure, e. g. by considering graphs of a fixed topology like a k -nearest neighbor graph. Diffusion is then constrained to the newly obtained graph representation.

To summarize, diffusion processes have shown to be an indispensable tool for improving retrieval performance. In this paper, we first aim at providing a thorough overview of the state-of-the-art in this field in Section 2, where we highlight and discuss similarities and differences between related work. Additionally, we outline relations to a game theoretical process and derive a novel formulation for diffu-



Figure 1: Illustration of usefulness of diffusion processes for retrieval. Affinities before (left) and after (right) the diffusion are used to assign each element to the highlighted query samples according to their pairwise affinities. As can be seen, pairwise affinities alone are not sufficient to capture the intrinsic structure of the data manifold. Applying a diffusion process spreads affinities and allows to significantly improve retrieval performance (best viewed in color).

sion based on evolutionary dynamics. Finally, it turns out, that all these methods can be summarized into one, generic diffusion framework, which we describe in Section 3. Related work then represents specific instances of our generic definition and this allows us to propose several new algorithms, by simply combining the most promising ideas of the related papers. We finally provide an evaluation, focusing on highlighting the ideas that have the highest influence on the retrieval scores. In such a way, we are able to derive a formulation that e. g. obtains a 100% bullseye score on the popular MPEG7 data set, as it is shown in Section 4.

2. Related Work

We start by reviewing the state-of-the art in the field of retrieval, with a specific focus on methods based on diffusion processes. Diffusion processes in general start from a provided pairwise $N \times N$ affinity matrix \mathbf{A} , which relates N different elements to each other. First step, is to interpret the matrix \mathbf{A} as a graph $\mathcal{G} = (V, E)$, consisting of N nodes $v_i \in V$, and edges $e_{ij} \in E$ that link nodes to each other, fixing the edge weights to the provided affinity values A_{ij} . Diffusion processes then spread the affinity values through the entire graph, based on the defined edge weights. This process is interpretable as a random walk on the graph, where a so-called transition matrix defines probabilities for walking from one node to a neighboring node. Instead of tracking where an individual random walk goes to, usually the probability distribution over vertices after a certain number of steps is of interest. As we will see in the following discussion of related work, diffusion processes mainly differ in (a) initialization, (b) definition of the transition matrix and (c) definition of the diffusion process. Our core findings on this are finally summarized in Table 1.

We start with the definition of a random walk on a graph $\mathcal{G} = (V, E)$. For this, we have to define the random walk transition matrix as

$$\mathbf{P} = \mathbf{D}^{-1} \mathbf{A}, \quad (1)$$

where \mathbf{D} is an $N \times N$ square matrix defined by

$$d_{ij} = \begin{cases} \deg(i) & \text{if } i = j \\ 0 & \text{otherwise} \end{cases}, \quad (2)$$

where $\deg(i)$ is the degree of a vertex, i. e. the sum over its edge weights. Thus, \mathbf{P} is a row-stochastic matrix (rows sum up to 1), containing the transition probabilities for a random walk in the corresponding graph. Due to the structure of the state transitions, it is possible to define the probability vector \mathbf{f}_t of being at a specific node after t steps of random walks as

$$\mathbf{f}_t = \mathbf{f}_0 \mathbf{P}^t, \quad (3)$$

where \mathbf{P}^t is the power of the matrix \mathbf{P} , i. e. \mathbf{P} raised to the positive integer t , and \mathbf{f}_0 is the $1 \times N$ dimensional vector of the initial distribution. Thus a single step of the diffusion process is defined by the simple update rule

$$\mathbf{f}_{t+1} = \mathbf{f}_t \mathbf{P}. \quad (4)$$

Note, that this iterative method relates to the Power Iteration method, which computes the principal eigenvector of a matrix. Indeed, the iterative process converges to the left eigenvector of the transition matrix \mathbf{P} with corresponding eigenvalue 1.

The Random Walk model was later extended to one of the first and most successful retrieval methods: the Google Page Rank system [17]. This system was originally designed for objectively ranking webpages, by effectively measuring the human interest in related pages, considering the underlying hyperlink structure. Thus, for Random Walk diffusion, each webpage is represented as a node and the edge weights are fixed in relation to the number of outgoing links per page. Main insight of the Page Rank system is that a web user does not always purposefully click to navigate, sometimes he/she jumps to a potentially random page. Thus, the standard random walk is modified, where at each time step t a random walk step is done with probability α , whereas a random jump to an arbitrary node is made with probability $(1 - \alpha)$. This leads to following update strategy

$$\mathbf{f}_{t+1} = \alpha \mathbf{f}_t \mathbf{P} + (1 - \alpha) \mathbf{y}, \quad (5)$$

where \mathbf{y} defines the probabilities of randomly jumping to the corresponding nodes. Iterating this update step until convergence returns the desired webpage ranking in the final distribution $\mathbf{f}_{t \rightarrow \text{inf}}$. For example, \mathbf{y} was selected as uniform vector for webpage ranking in [17] and this approach is referred to as *Global Page Rank* (GPR). Nevertheless, \mathbf{y} can be selected in any manner, and enables personalization to specific random jump preferences, which is then denoted as *Personalized Page Rank* (PPR).

Please note, that if we aim to analyze diffusion for many different and independently personalized page ranks, the

update can be written in matrix notation as

$$\mathbf{W}_{t+1} = \alpha \mathbf{W}_t \mathbf{P} + (1 - \alpha) \mathbf{Y}, \quad (6)$$

where the rows of the $N \times N$ matrix \mathbf{W} now contain the ranks of interest and the $N \times N$ matrix \mathbf{Y} is simply a stacking of all personalized row vectors \mathbf{y} into a matrix.

In [23] a method denoted as Ranking on Manifolds (ROM) was proposed, which is heavily related to the standard Page Rank approach. This method also iteratively updates an initialization vector (the ranking values for a query). Main difference to Page Rank is the slightly adapted transition matrix defined as

$$\mathbf{P}_{\text{NC}} = \mathbf{D}^{-1/2} \mathbf{A} \mathbf{D}^{-1/2}, \quad (7)$$

which we denote as \mathbf{P}_{NC} because of its analogy to the normalized Graph Laplacian used in normalized cut segmentation. This matrix is symmetric (in contrast to the Random Walk transition matrix \mathbf{P}). The diffusion process is the same as in the Personalized Page Rank (see Equation 5), where the sequence converges again to a non-trivial solution, where \mathbf{P} and \mathbf{P}_{NC} have the same eigenvalues and eigenvectors related by $\mathbf{x}_P = \mathbf{x}_{\text{NC}} \mathbf{D}^{1/2}$, where \mathbf{x}_P is the principal eigenvector of \mathbf{P} and \mathbf{x}_{NC} of \mathbf{P}_{NC} , respectively.

A related diffusion process originates from the field of semi-supervised learning denoted as label propagation (LP) [24], where the goal is to spread information from labeled to unlabeled data. In the scope of image retrieval, this means that the only labeled point is the query instance itself. Label Propagation applies the standard random walk diffusion step (Equation 4), but in contrast injects query information in a specific manner by fixing $f(i) = 1$ after each diffusion step. Running this diffusion process for infinite time converges to a constant vector and assigns each node the same label (since only the query is labeled). Hence, in this case the evolution of the vector is of interest and the process has to be stopped at the right moment (mostly after t iterations), where adapting t allows to perform kind of multi-scale analysis.

Graph Transduction (GT) [2] directly focuses on applying Label Propagation for retrieval tasks. Key idea of this approach is a normalization of the input affinity matrix \mathbf{A} , that is adapted specifically for each query. Thus, graph transduction cannot be written in matrix form as shown in Equation 6, since the matrix \mathbf{A} differs for all query elements. As a consequence this approach is not scalable to large data sets. More interesting, normalization is based on an analysis of the nearest neighbors (NN) of the query, which constrains the diffusion process locally and yields to different transition matrices for each query element. Nevertheless, by fixing a neighborhood-graph and by using the same normalization for all queries, Graph Transduction could be easily reformulated in matrix notation.

In [21] the idea of constraining the diffusion graph to local neighborhoods is further investigated (LCDP). Authors also use a kNN-Graph to define the locality, but in contrast to [2] additionally adapt the diffusion process. Key idea is to define the transition probabilities in a novel manner, setting them to high values if and only if all (!) paths between the kNN of two nodes are short. The paper describes that this idea can be encoded into the diffusion process by adapting the update step to

$$\mathbf{W}_{t+1} = \mathbf{P} \mathbf{W}_t \mathbf{P}^T. \quad (8)$$

Interestingly, in [22] exactly the same diffusion step is derived from a different point of view. It is denoted as Tensor Graph Diffusion (TGD) and aims at integrating relations of higher order than pairwise affinities into the diffusion process. This is achieved by building a new graph representation obtained by the tensor product of the standard graph with itself. In such a way an edge weight in the new graph representation relates four tuples of vertices to each other and aims at revealing intrinsic higher-order relations between the elements analyzed. The diffusion process on such a graph again converges to a non-trivial solution. Nevertheless, this form of diffusion is impracticable since the graphs are really huge. For this reason, authors show that the diffusion process on the tensor graph is identical to a modified diffusion process on the standard graph, which turns out to be the same as in [21] (see Equation 8).

In [20] another variant to define a neighborhood graph for diffusion, denoted as shortest path propagation (SPP), was presented, where shortest paths were independently analyzed for each query element. This again increases the runtime significantly and achieves only slightly increased retrieval scores. Another neighborhood graph variant was introduced in [22], a so-called dominant set (DS) approach. For each query, a dominant set is identified by evolutionary dynamics from game theory, which identifies the most closely related neighbors. Such a dominant set approach is also considered in [16] for defining a global neighborhood graph. Again, just for retrieving the improved neighborhood graph an independent diffusion process has to be applied, which significantly increases runtime.

The methods discussed above, all improve retrieval scores, but somehow lack the notion of a global similarity metric. In contrast, diffusion maps [4] are also based on a diffusion process on a graph and additionally induce a global similarity metric. Also for diffusion maps the random walk transition matrix \mathbf{P} is used to propagate affinities through the manifold. The final diffusion map distances (for a fixed t) can be found in closed form, by eigenvalue weighted Euclidean distances between eigenvectors of \mathbf{P} . Since for $t \rightarrow \infty$ an equilibrium is reached, where all diffusion distances equal zero, the dynamics of the diffusion process are of main interest.

Recently, in [8] a Self-Smoothing Operator (SSO) was introduced which is quite related to diffusion maps. It is equivalent to Page Rank, simply using the affinities for a query point as initialization. This approach was extended in [19] by a slight adaption of the diffusion step denoted as Self Diffusion (SD), where a closed form solution for specific t values was provided. The paper further describes a way to estimate an optimal t value and a way to reduce complexity by a matrix factorization approach.

Interestingly also the recently quite popular evolutionary Replicator Dynamics (RD) can be interpreted as diffusion process on a matrix. These dynamics are already applied in several computer vision applications, e. g. for matching image segments and points [1], for finding common spatial visual patterns [14] or for clustering [15], showing promising results in all applications. RD are a first order evolutionary dynamic from the field of game theory defined by the following diffusion process

$$f_{t+1}^i = f_t^i \frac{(\mathbf{A} \mathbf{f}_t)_i}{\mathbf{f}_t^T \mathbf{A} \mathbf{f}_t}, \quad (9)$$

where f_t^i is the i -th element of the vector \mathbf{f} . As an important, additional constraint the vector \mathbf{f} has to lie on the simplex Δ defined as

$$\Delta = \{ \mathbf{f} \in \mathbb{R}^N : f^i \geq 0 \text{ and } \mathbf{1}^T \mathbf{f} = 1 \}, \quad (10)$$

where $\mathbf{1}$ is a vector of ones, i. e. $\sum f^i = 1$. The dynamics start with a random initialization \mathbf{f}_0 which also has to lie on the simplex. The simplex Δ is invariant under the replicator dynamics formulation, which means that every trajectory starting on the simplex will remain on the simplex. For more details and convergence proofs of these dynamics see e. g. [18]. As can be seen the iterative process highly relates to diffusion processes if written in matrix notation as

$$\mathbf{W}_{t+1} = \mathbf{W}_t \odot \mathbf{A} \mathbf{W}_t \text{ and } \mathbf{W}_{t+1} = \mathbf{D}^{-1} \mathbf{W}_{t+1}, \quad (11)$$

where \odot is the Hadamard matrix product. The first step in this update equation spreads the similarities, whereas the second part ensures that the newly obtained matrix is again row-stochastic, i. e. each row lies on the simplex. Although, the matrix \mathbf{A} is not a valid transition matrix it encodes the same global information as the commonly used \mathbf{P} and is therefore, also applicable for diffusion.

Also methods from the field of clustering are frequently based on such diffusion processes, and we would like to mention the most related ones. In [11] power iteration clustering (PIC) was proposed. This method is also based on the observation that the largest eigenvector of the random walk transition matrix \mathbf{P} is always a constant vector with eigenvalue 1. Power Iteration is used to find this largest eigenvector, where again the evolution of the vector is of

interest, in this case to infer reasonable clusters. In Authority Shift Clustering (ASC) [3] the personalized page rank matrix is considered in the diffusion process and emerging clusters during evolution are analyzed to define a hierarchical clustering result.

Finally, a few methods try to improve provided affinity matrices without diffusion processes. For example, in [5] a meta similarity approach was proposed. Core idea is to characterize similarities between elements, by comparing similarities of the corresponding kNN sets. The process is quite simple: (a) estimate the kNN and build a novel vector representation, where only the similarities to the kNN are set and all others are fixed to 0, (b) use L1 distances between the novel representations to measure their similarity. Obviously, the correct choice of K is quite important in such a process. In [9] the mutual-kNN graph was exploited to update similarities without any type of diffusion. The main idea is to build a specific graph structure and simply let the shortest path through the graph define the new affinities between the elements.

To summarize, a lot of different diffusion principles have been proposed in the last years. All discussed methods are heavily related to each other, without thoroughly specifying similarities and differences. Thus, the idea of the paper is to unite all these approaches under a common framework. In such a way, we are also able to derive novel algorithms, uniting advantages of the different concepts. Our framework is described in detail in the next section.

3. Improving Retrieval by Diffusion

Based on the findings on related work in the field of retrieval, we are able to formulate a generic framework for diffusion processes, where all methods discussed in the previous section represent single instances of our framework. Our generic formulation then allows to define and validate a set of novel diffusion processes, that are thoroughly evaluated in the experimental section. Let us first define some basic notations used throughout this section.

We assume that we have given a set $\mathcal{X} = \{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_N\}$ of N elements with $\mathbf{x}_i \in \mathcal{R}^D$ and measures of pairwise similarities A_{ij} between each point pair. The most common way to define such similarities is to use a Gaussian Kernel defined as

$$A_{ij} = \exp\left(-\frac{\|\mathbf{x}_i - \mathbf{x}_j\|^2}{2\sigma^2}\right), \quad (12)$$

where σ is a parameter to fix. All affinity values together form the $N \times N$ input affinity matrix \mathbf{A} defined as

$$\mathbf{A} = \begin{pmatrix} A_{11} & \cdots & A_{1N} \\ \vdots & \ddots & \vdots \\ A_{N1} & \cdots & A_{NN} \end{pmatrix}. \quad (13)$$

Method	Abbr.	Initialization \mathbf{W}^0	Transition \mathbf{T}	Diffusion
Global Page Rank [17]	GPR	\mathbf{u}	\mathbf{P}	$\mathbf{f}_{t+1} = \mathbf{f}_t \mathbf{T}$
Personalized Page Rank [17]	PPR	\mathbf{u}	\mathbf{P}	$\mathbf{f}_{t+1} = \alpha \mathbf{f}_t \mathbf{T} + (1 - \alpha) \mathbf{y}$
Ranking on Manifolds [23]	ROM	\mathbf{u}	\mathbf{P}_{NC}	$\mathbf{f}_{t+1} = \alpha \mathbf{f}_t \mathbf{T} + (1 - \alpha) \mathbf{y}$
Label Propagation [24]	LP	\mathbf{y}	\mathbf{P}	$\mathbf{f}_{t+1} = \mathbf{f}_t \mathbf{T}$ and $f(i) = 1$
Graph Transduction [2]	GT	\mathbf{y}	\mathbf{P}	$\mathbf{f}_{t+1} = \mathbf{f}_t \mathbf{T}$ and $f(i) = 1$
Locally Constrained DP [21]	LCDP	\mathbf{A}	\mathbf{P}_{kNN}	$\mathbf{W}_{t+1} = \mathbf{T} \mathbf{W}_t \mathbf{T}^T$
Tensor Graph Diffusion [22]	TGD	\mathbf{A}	\mathbf{P}_{DS}	$\mathbf{W}_{t+1} = \mathbf{T} \mathbf{W}_t \mathbf{T}^T + \mathbf{I}$
Shortest Path Propagation [20]	SPP	\mathbf{y}	\mathbf{P}_{SP}	$\mathbf{f}_{t+1} = \mathbf{f}_t \mathbf{T}$
Self Smoothing Operator [8]	SSO	\mathbf{A}	\mathbf{P}	$\mathbf{W}_{t+1} = \mathbf{W}_t \mathbf{T}$
Self Diffusion [19]	SD	\mathbf{A}	\mathbf{P}	$\mathbf{W}_{t+1} = \mathbf{W}_t \mathbf{T} + \mathbf{I}$
Replicator Dynamics [18]	RD	\mathbf{u}	\mathbf{A}	$\mathbf{f}_{t+1} = \mathbf{f}_t \odot \mathbf{T} \mathbf{f}_t$ and $\mathbf{f}_{t+1} = \mathbf{f}_{t+1} / \mathbf{f}_{t+1} $
Power Iteration Clustering [11]	PIC	\mathbf{s}	\mathbf{P}	$\mathbf{f}_{t+1} = \mathbf{T} \mathbf{f}_t$ and $\mathbf{f}_{t+1} = \mathbf{f}_{t+1} / \mathbf{f}_{t+1} $
Authority Shift Clustering [3]	ASC	\mathbf{P}_{PPR}	\mathbf{P}_{PPR}	$\mathbf{W}_{t+1} = \mathbf{W}_t \mathbf{T}$

Table 1: Overview of related work in the field of diffusion processes. All methods start from defined initialization \mathbf{W}^0 (first column) and iteratively propagate similarities through the underlying manifold using defined transition matrix \mathbf{T} (second column) in the defined diffusion scheme (third column). Notation is as follows: \mathbf{A} is the input affinity matrix, \mathbf{u} is the unit vector, \mathbf{y} is the zero-vector with only the query element fixed to 1, \mathbf{s} is the stationary vector as defined in [11], \mathbf{P} is the standard random walk transition matrix, \mathbf{P}_{kNN} is the random walk transition matrix constrained to the k-nearest neighbors, \mathbf{P}_{NC} is the symmetric normalized cut transition matrix, \mathbf{P}_{SP} is the shortest path transition matrix as defined in [20] and \mathbf{P}_{PPR} is the personalized page rank transition matrix.

The goal of retrieval is to consider each element \mathbf{x}_i as independent query and rank all other points according to their pairwise affinities to the specified query point \mathbf{x}_i . As baseline, the input affinity matrix \mathbf{A} could be directly analyzed to perform retrieval. Nevertheless, as outlined in the previous sections, such an approach neglects the underlying data manifold and does not lead to satisfactory results. Thus, we next define a generic diffusion framework, which converts a given affinity matrix \mathbf{A} into a novel, diffused version \mathbf{A}^* . Retrieval is then based on the obtained matrix \mathbf{A}^* .

Analyzing the related work, it can clearly be seen that diffusion processes mainly consist of three important steps: (A) Initialization, (B) Definition of the transition matrix and (C) Definition of the diffusion process. We are able to identify several variants for all three steps by analyzing the methods discussed in the last section. Our framework allows four types of initialization (A1)-(A4) (Table 2), six different types of transition matrices (B1)-(B6) (Table 3) and finally three main diffusion variants (C1)-(C3) (Table 4). The corresponding tables list the different choices and outlines related work that fits into the different variants.

Thus in overall we get $4 \times 6 \times 3 = 72$ possibilities to define diffusion processes. For all combinations we start from the initialization \mathbf{W}^0 , and apply iterative updates using the transition matrix \mathbf{T} in the defined diffusion process, as long as the average number of changing elements in the obtained rankings between two subsequent iterations is below ϵ (we fix ϵ to 0.01 in all experiments), which finally

yields the improved matrix \mathbf{A}^* . Please note, that some of the combinations have closed form solutions, which are not further considered, since they require slow and complex operations like the inverse of huge matrices. Therefore, we focus on the iterative diffusion processes that converge to the same solution and that are directly applicable for large scale analysis. Please note, that only a few of these 72 variants have been applied before in the scope of retrieval applications: (A1)+(B1)+(C1): SSO [8], SD [19], GT [2] – (A1)+(B5)+(C2): TGD [22] – (A1)+(B4)+(C2): LCDP [21], and for example applying evolutionary dynamics (C3) is for the first time considered for retrieval.

4. Experiments

Goal of our experiments is a thorough comparison of the achievable performance of the 72 diffusion variants that can be applied based on the generic framework described in the last section. Specifically, we want to derive statements about the importance of several proposed extensions. We implemented our diffusion framework in Matlab and code is provided at <http://vh.icg.tugraz.at>. Experiments analyze three different data sets in the scope of retrieval: *MPEG7* (shape), *YALE* (faces) and *ORL* (faces).

MPEG7 [10] is one of the most popular data sets for demonstrating retrieval performance. It consists of 1400 binary images showing silhouettes from 70 different categories. *MPEG7* has seen a tremendous progress in the re-

ID	Initialization \mathbf{W}^0	Methods
(A1)	Affinity Matrix \mathbf{A}	SSO [8], SD [19], TGD [22], LCDP [21]
(A2)	Identity Matrix \mathbf{I}	LP [24], GT [2], SPP [20]
(A3)	Transition Matrix \mathbf{P}	ASC [3]
(A4)	Transition Matrix \mathbf{P}_{kNN}	ASC [3]

Table 2: Four different types of initialization and related work based on the different approaches.

ID	Transition Matrix \mathbf{T}	Methods
(B1)	Transition Matrix \mathbf{P}	SSO [8], SD [19], GPR [17], PPR [17], LP [24], GT [2], PIC [11]
(B2)	Personalized Page Rank Transition Matrix \mathbf{P}_{PPR} ,	ASC [3]
(B3)	Normalized Cut Transition Matrix \mathbf{P}_{NC}	ROM [23]
(B4)	K-Nearest Neighbors Transition Matrix \mathbf{P}_{kNN}	LCDP [21]
(B5)	Dominant Set Neighbors \mathbf{P}_{DS}	TGD [22]
(B6)	Affinity matrix \mathbf{A}	RD [18]

Table 3: Six different types of transition matrices and related work based on the different approaches.

ID	Update Scheme	Methods
(C1)	$\mathbf{W}_{t+1} = \alpha \mathbf{W}_t \mathbf{T} + (1 - \alpha) \mathbf{I}$,	SSO [8], SD [19], ROM [23], ASC [3]
(C2)	$\mathbf{W}_{t+1} = \mathbf{T} \mathbf{W}_t \mathbf{T}^T$	LCDP [21], TGD [22]
(C3)	$\mathbf{W}_{t+1} = \mathbf{W}_t \otimes \mathbf{T} \mathbf{W}_t$ and $\mathbf{W}_{t+1} = \mathbf{D}^{-1} \mathbf{W}_{t+1}$	RD [18]

Table 4: Three different update schemes and related work based on the different approaches.

cent years, concerning (a) the shape matching algorithms applied for defining the pairwise affinities and (b) the improvements in the field of diffusion processes. We use the best performing shape matching algorithm of [7] to define the pairwise distances. Retrieval accuracy is measured by ranking all other elements for each query and calculating the average number of occurrences of elements of the same category as the query, within the 40 first ranked elements. This measure is denoted as the bullseye score and its upper limit is 100%, which means that for each query the 20 instances of the same category are ranked within the first 40. If directly analyzing the distance matrix of [7] without applying diffusion we get a baseline bullseye score of 93.55%.

YALE face dataset B [6] is a standard benchmark for face clustering, which consists of face images of various persons seen under different poses and illumination conditions. We extract the same subset as [8] to define a retrieval task, where 15 subjects are shown in 11 different conditions. Each image is down-sampled and then normalized to 0-mean and 1-variance. Distances are directly defined by pairwise Euclidean distances between the vectorized representations. Retrieval score is again measured by a bullseye score, analyzing the 15 most similar instances. The baseline bullseye score is 69.48% for this data set.

ORL has 40 subjects with 10 grayscale images per subject, where pose, illumination, and expression are diverging. The same representation as for *YALE* data set is used to build the distance matrix, and the bullseye score considering 15 closest neighbors is used to evaluate retrieval. The baseline bullseye score is 62.35% for this data set.

4.1. Evaluation of Retrieval Performance

As a first experiment we identify the most promising combination out of our pool of 72 variants, by analyzing the best achievable bullseye score after diffusion. For this we thoroughly evaluated all combinations on all three data sets considered, where we fixed the parameter K relative to the number of instances within the same class (i. e. 10 for *MPEG7*, 5 for *YALE* and *ORL*). We independently optimized over σ values for each combination, to demonstrate the best obtainable retrieval performance. According to the Page Rank approach, we fixed α to 0.85. Tables 5 (*MPEG7*), 6 (*YALE*) and 7 (*ORL*) list bullseye scores in a compactified manner.

First of all, results show that applying diffusion, to spread affinities through the manifold, in all cases significantly improves the obtainable scores: for *MPEG7*: 93.55% \rightarrow 100%, for *YALE* 69.48% \rightarrow 77.30% and for

ORL 62.35% \rightarrow 77.42%. Furthermore, constraining the diffusion process to the local neighbors by adapting the transition matrix \mathbf{P} to the k -nearest neighbors \mathbf{P}_{kNN} (B4) or to the dominant set \mathbf{P}_{DS} (B5) seems to bring the largest boost in performance, especially for the diffusion variants (C1) and (C2). Thus, \mathbf{P}_{kNN} seems the way to go, because \mathbf{P}_{DS} requires significantly more time (since for obtaining the dominant set an independent diffusion step is required), while in essence results are not substantially different. Furthermore, all update schemes provide reasonable results on all data sets, where the update scheme (C2) seems to yield the best performance. Nevertheless, scheme (C2) requires an additional matrix multiplication per iteration.

Initialization, in general is not so important, since methods converge to the same solution. Only, if using the game theoretical diffusion (C3) one has to initialize either using the affinity matrix \mathbf{A} (A1) or the transition matrix \mathbf{P} (A3), other initializations do not provide reasonable results for (C3). In overall, starting from the kNN-page rank transition matrix \mathbf{P}_{kNN} (A4) seems to lead to fastest convergence and best performance, which up to now has only be considered in the clustering approach of [3].

To summarize, as most promising combination, we choose to initialize the diffusion process by the transition matrix \mathbf{P}_{kNN} (A4), to constrain the transition matrix to the k -nearest neighbors (B4) and to apply the higher-order diffusion process (C2). This approach consistently yields the best results, while having low computation time, which enables large-scale retrieval applications. This configuration is used for the experiments in the next section.

4.2. Influence of Locality

As we demonstrated in the last section, constraining the diffusion process locally is a promising direction to improve retrieval scores. Thus, in this experiment we analyze the influence of different local neighborhood sizes on the obtainable results, by thoroughly testing a range of K values, defining the number of nearest neighbors considered. Obtainable scores for the most promising diffusion variant (see previous section) are shown in Figure 2.

As can be seen, selecting a reasonable K value is important to obtain optimal retrieval results, nevertheless for all $K \geq 3$ performance is significantly improved in comparison to the baseline. On MPEG7 the choice is insignificant, from $K = 9$ upwards the diffusion process always leads to a 100% bullseye score, since this data set is already saturated. On the other two data sets YALE and ORL, tuning of K allows to improve performance, where on ORL an optimal bullseye score of 77.30% is achieved for $K = 5$, which decreases down to 73.40%, if using the number of neighbors from the same class (10). For YALE, the optimal score is 77.08% for $K = 5$ which decreases to 73.39% for the correct $K = 11$. Due to this influence of K on re-

MPEG7	C1	C2	C3
B1	99.89 (A3)	99.93 (A3)	99.91 (A3)
B2	99.91 (A3)	99.94 (A3)	99.89 (A1)
B3	99.89 (A3)	99.90 (A4)	99.91 (A3)
B4	99.99 (A1)	100.00 (A4)	99.90 (A3)
B5	99.99 (A1)	100.00 (A4)	99.91 (A3)
B6	99.88 (A3)	99.88 (A3)	99.91 (A3)

Table 5: Scores on the MPEG7 data set. Bullseye rating for the first 40 elements is listed. Best initialization method is listed in brackets. Baseline: 93.55%

YALE	C1	C2	C3
B1	71.90 (A4)	72.23 (A4)	71.07 (A1)
B2	70.96 (A4)	71.24 (A1)	71.07 (A1)
B3	71.46 (A4)	71.29 (A4)	71.18 (A1)
B4	73.39 (A3)	77.30 (A2)	71.24 (A1)
B5	74.38 (A3)	76.20 (A2)	71.46 (A1)
B6	70.91 (A3)	70.30 (A3)	71.74 (A1)

Table 6: Scores on the YALE data set. Bullseye rating for the first 15 elements is listed. Best initialization method is listed in brackets. Baseline: 69.48%

ORL	C1	C2	C3
B1	71.88 (A4)	73.12 (A4)	65.65 (A1)
B2	73.58 (A4)	73.75 (A4)	67.20 (A3)
B3	72.12 (A4)	72.95 (A4)	65.62 (A1)
B4	75.98 (A4)	77.42 (A2)	70.93 (A1)
B5	74.12 (A3)	75.50 (A4)	68.73 (A3)
B6	71.28 (A4)	71.40 (A4)	65.65 (A1)

Table 7: Scores on the ORL data set. Bullseye rating for the first 15 elements is listed. Best initialization method is listed in brackets. Baseline: 62.35%

sults, researching methods to automatically find an optimal K for constrained diffusion processes in an efficient manner seems to be a promising future direction of research.

4.3. MPEG7 Results

MPEG7 is the most frequently used data set for evaluating retrieval methods. Thus, in Table 8 we show the recent developments on MPEG7, listing different state-of-the-art methods and the progress on the retrieval score over time. As can be seen, we are able to demonstrate for the first time a 100% bullseye score on this challenging data set, i. e. for each query after diffusion, all 20 instances of the same category are ranked within the first 40 returned instances.

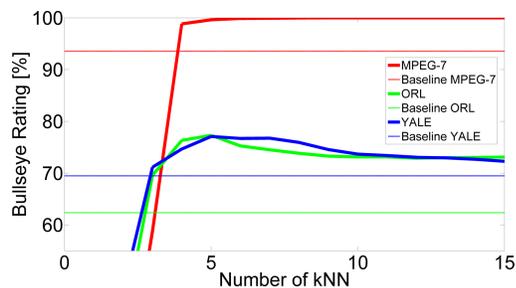


Figure 2: Influence of adapting the local neighborhood on the obtainable retrieval scores (best viewed in color).

Method	Bullseye score (in%)
IDSC [12]	85.40
IDSC + GT [2]	91.61
IDSC+mkNN [9]	93.40
ASC [13]	88.30
ASC+LCDP [13]	95.96
ASC+TN+TPG [22]	96.47
AIR [7]	93.67
AIR+TN+TPG [22]	99.99
AIR+Proposed	100.00

Table 8: Comparison of retrieval performance on MPEG7.

5. Conclusion

In this paper we introduced a generic diffusion framework evaluated in the scope of retrieval applications. Our diffusion methods start from different initializations, and use different combinations of transition matrix and diffusion process variants, to iteratively spread affinities through the underlying data manifold. We further outlined that the state-of-the-art in this field represents specific instances of our diffusion framework. We provided a thorough evaluation, highlighting that constraining the diffusion locally achieves the most promising boost in performance, where automatically selecting a reasonable local neighborhood size is still an open issue. The best performing instance of our generic framework for example achieved a 100% bullseye score on the popular MPEG7 data set.

Acknowledgements Authors gratefully acknowledge financial support from the FP7-ICT EU project Nr. 601139 CultAR (Culturally Enhanced Augmented Realities).

References

[1] A. Albarelli, S. Bulò, A. Torsello, and M. Pelillo. Matching as non-cooperative game. In *ICCV*, 2009.

[2] X. Bai, X. Yang, L. Latecki, W. Liu, and Z. Tu. Learning context-sensitive shape similarity by graph transduction. *PAMI*, 32:861–874, 2010.

[3] M. Cho and K. M. Lee. Authority-shift clustering: Hierarchical clustering by authority seeking on graphs. In *CVPR*, 2010.

[4] R. Coifman and S. Lafon. Diffusion maps. *Applied and Computational Harmonic Analysis: Special issue on Diffusion Maps and Wavelets*, 21:5–30, 2006.

[5] A. Egozi, Y. Keller, and H. Guterman. Improving shape retrieval by spectral matching and meta similarity. *IEEE Transactions on Image Processing*, 5:1319–1327, 2010.

[6] A. Georghiades, P. Belhumeur, and D. Kriegman. From few to many: Illumination cone models for face recognition under variable lighting and pose. *PAMI*, 23:643–660, 2001.

[7] R. Gopalan, P. Turaga, and R. Chellappa. Articulation-invariant representation of non-planar shapes. In *ECCV*, 2010.

[8] J. Jiang, B. Wang, and Z. Tu. Unsupervised metric learning by self-smoothing operator. In *ICCV*, 2011.

[9] P. Kontschieder, M. Donoser, and H. Bischof. Beyond pairwise shape similarity analysis. In *ACCV*, 2009.

[10] L. Latecki, R. Lakamper, and U. Eckhardt. Shape descriptors for non-rigid shapes with a single closed contour. In *CVPR*, 2000.

[11] F. Lin and W. W. Cohen. Power iteration clustering. In *ICML*, 2012.

[12] H. Ling and D. Jacobs. Shape classification using the inner-distance. *PAMI*, 29(2):286–299, Feb. 2007.

[13] H. Ling, X. Yang, and L. J. Latecki. Balancing deformability and discriminability for shape matching. In *ECCV*, 2010.

[14] H. Liu and S. Yan. Common visual pattern discovery via spatially coherent correspondences. In *CVPR*, 2010.

[15] H. Liu and S. Yan. Robust graph mode seeking by graph shift. In *ICML*, 2010.

[16] H. Liu, X. Yang, L. J. Latecki, and S. Yan. Dense neighborhoods on affinity graph. *IJCV*, 2012.

[17] L. Page, S. Brin, R. Motwani, and T. Winograd. The Page-Rank citation ranking: Bringing order to the Web. Technical report, Stanford, 1998.

[18] M. Pelillo. Matching free trees with replicator equations. In *NIPS*, 2001.

[19] B. Wang and Z. Tu. Affinity learning via self-diffusion for image segmentation and clustering. In *CVPR*, 2012.

[20] J. Wang, Y. Li, X. Bai, Y. Zhang, C. Wang, and N. Tang. Learning context-sensitive similarity by shortest path propagation. *Pattern Recognition*, 44(10-11):2367–2374, 2011.

[21] X. Yang, S. Koknar-Tezel, and L. Latecki. Locally constrained diffusion process on locally densified distance spaces with applications to shape retrieval. In *CVPR*, 2009.

[22] X. Yang, L. Prasad, and L. J. Latecki. Affinity learning with diffusion on tensor product graph. *PAMI*, 35, 2013.

[23] D. Zhou, J. Weston, A. Gretton, O. Bousquet, and B. Schölkopf. Ranking on data manifolds. In *NIPS*. 2004.

[24] X. Zhu. *Semi-Supervised Learning with Graphs*. PhD thesis, Language Technologies Institute, Carnegie Mellon University, 2005.