

스펙트럴 그래프 기반 Commute Time 임베딩 특성 분석

한 희 일[†]

요 약

본 논문에서는 파형 신호와 이미지 등에서 패치를 추출하고 이를 패치 그래프로 구성한 다음, 이로부터 각 패치 간의 컴्यू트 타임을 구하여 이에 기반한 임베딩 기법을 구현하고, 가장 널리 이용되는 PCA(principal component analysis) 임베딩 결과와 비교 분석한다. 임베딩에서 차원을 줄일 경우 원 임베딩과 축소된 차원의 임베딩 간에는 오차가 크지 않도록 차원을 결정하는 것이 일반적이다. 하지만 본 논문에서 구현한 임베딩 방식은 삼차원 이하로 줄여 오차가 80~90%를 상회하여도 축소된 차원의 임베딩 공간에서 각 신호 고유의 기하 구조를 생성하므로 패턴 분류나 기계 학습 등의 응용 목적에 활용 가능성을 실험으로 확인한다.

Analysis of Commute Time Embedding Based on Spectral Graph

Hee-il Hahn[†]

ABSTRACT

In this paper an embedding algorithm based on commute time is implemented by organizing patches according to the graph-based metric, and its performance is analyzed by comparing with the results of principal component analysis embedding. It is usual that the dimensionality reduction be done within some acceptable approximation error. However this paper shows the proposed manifold embedding method generates the intrinsic geometry corresponding to the signal despite severe approximation error, so that it can be applied to the areas such as pattern classification or machine learning.

Key words: Commute time(컴्यू트 타임), Embedding(임베딩), Manifold learning(다양체 학습), Spectral graph(스펙트럴 그래프), Graph Laplacian(그래프 라플라시안)

1. 서 론

인공지능, 데이터 마이닝, 컴퓨터 비전 등의 분야에서 기계 학습(machine learning)의 다양한 알고리즘은 패턴 분류나 함수 등을 추정하기 위하여 방대한 양의 데이터 처리를 요구한다. 또한, 알고리즘의 성능을 높이기 위해서는 다양한 예의 데이터를 확보하여 이를 학습에 활용하여야 한다. 이러한 과정은 알고리즘의 처리 속도를 저하시킬 뿐만 아니라 최적의 결과를 얻는데 어려움을 초래하기도 한다. 이러한

문제를 해결하기 위하여 다양한 기술이 개발되고 있다. 수많은 데이터로 구성된, 처리하고자 하는 데이터 세트를 고차원 공간의 데이터로 간주하고 이를 저차원 공간의 점으로 표현하여 차원을 줄이는 작업(dimensionality reduction)은 기계 학습에서 가장 중요하고 필수적으로 해결하여야 할 분야에 속한다. 예전부터 주어진 데이터의 정보량을 줄이기 위하여 다양한 압축 방식이 제안되었지만, 데이터를 다양체 위의 점으로 간주하여 이를 저차원 공간으로 임베딩함으로써 기하학적으로 차원을 줄이려는 연구는 최근

※ 교신저자(Corresponding Author) : 한희일, 주소 : 경기도 용인시 모현면 왕산리 한국외국어대학교 정보통신공학과(449-791), 전화 : 031)-330-4356, FAX : 031)333-4256, E-mail : hihahn@hufs.ac.kr

접수일 : 2013년 9월 4일, 수정일 : 2013년 11월 26일

완료일 : 2013년 11월 29일

[†] 정희원, 한국외국어대학교 정보통신공학과

※ 본 연구는 2013학년도 한국외국어대학교 학술연구비 지원으로 수행되었음.

들어 활발히 진행되고 있다[1,2].

데이터의 차원을 줄이기 위한 가장 보편적이고 고전적인 방법으로는 PCA(principal component analysis)가 있다. 일련의 데이터가 주어지면, PCA는 데이터 세트가 최대 분산을 갖는 방향을 찾아서 적절한 수의 방향벡터로 구성된 공간에 데이터 세트를 사영 시킴으로써 차원을 줄일 수 있다. 대부분의 데이터가 데이터 세트의 선형 부분공간(linear subspace)에 존재할 때 PCA는 우수한 성능을 보이지만 그렇지 않으면 성능이 급격히 저하되는 단점이 있다. 이를 해결하기 위하여 다양한 알고리즘이 제안되고 있는데, 이를 다양체 학습(manifold learning)이라고 부른다. 다시 말해서, PCA가 선형 공간에서 잘 동작하도록 최적화되어 있듯이 다양체 학습은 비선형 공간에서 데이터의 차원을 효과적으로 줄이도록 설계된 것으로 볼 수 있다. ISOMAP [2]은 다양체 학습 알고리즘 중에서 가장 먼저 제안된 것으로 널리 응용되고 있으며, MDS(multidimensional scaling) [3]를 확장한 것으로 볼 수 있다. 이 방식은 우선 주어진 입력 데이터 간의 측지 거리(geodesic distance)를 구하여야 하는데, k-NN(k-nearest neighbor) 그래프에서 각 노드 간의 최단 거리를 측지 거리로 간주한다. 그런 다음, MDS를 이용하여 그람 행렬(Gram matrix)을 구함으로써 저차원 유클리드 공간의 해당되는 점을 찾는다. 이 방식은 파라미터 공간(local chart)이 컨벡스(convex)임을 가정하므로 그렇지 않을 때에는 오차의 크기가 급격히 증가하는 단점이 있다. LLE(locally linear embedding) [4]는 ISOMAP과 거의 동시에 제안된 것으로, 다양체가 매우 부드럽고 작은 패치 내에서는 선형성이 보장된다는 가정을 이용한다. ISOMAP이 각 점 간의 거리를 보존하는(isometric) 맵인 반면에 LLE는 거리를 보존하지는 않지만 이들 간의 각을 보존하는(conformal) 특징이 있다. LLE는 그래프 라플라시안을 이용하는데, 이를 헤시안(Hessian) 행렬로 대체한 hLLE(Hessian LLE)는 거리를 보존함으로써 LLE의 단점을 극복하고 있다 [5]. LE(Laplacian eigenmap) [6,7]는 주어진 데이터를 각 노드로 하는 그래프를 구성하고 각 노드를 유클리드 공간으로 임베딩하는 스펙트럴 그래프 방식을 채택하고 있다. 이 방식은 그래프 라플라시안 행렬에서 구한 고유치(eigenvalue)와 고유벡터(eigenvector)를 이용하여 그래프의 구조적인 특성을 분석

한다. LE는 데이터를 클러스터링하는 기능은 우수하지만 각 점 간의 거리에 대한 정의가 명확하지 않아 임베딩 방식으로는 널리 활용되고 있지 않다. 본 논문에서는 거리 개념을 보완하기 위하여 컴퓨터 타임을 이용하여 거리를 정의하고 임베딩 맵을 구한다. 현재까지 발표된 임베딩 관련 연구에서는 임베딩에서 구한 거리 정보나 좌표를 응용 목적에 적용하지만 임베딩 공간에서의 기하 구조를 분석하거나 활용하는 예는 거의 찾아 볼 수 없다. 또한, 임베딩에서 차원을 줄일 경우 본래의 임베딩과 축소된 차원의 임베딩 간에는 오차가 크지 않도록 차원을 결정한다. 예를 들어, 주어진 데이터에서 구한 공분산 행렬(covariant matrix)에서 PCA를 이용하여 차원을 줄일 때에는 오차 에너지가 10-20 % 이내에서 유지되도록 차원을 결정하는 것이 일반적이다. 이러한 방법으로는 차원을 크게 줄일 수 없을 뿐만 아니라 축소된 차원의 임베딩 공간에서의 데이터 분포 등을 시각화할 수 없는 단점을 노출한다. 이를 개선하기 위하여 본 논문에서는 이미지나 파형 신호 등에서 서로 겹치도록 패치(patch)를 추출하고 이들을 그래프로 구성한 다음 컴퓨터 타임으로 각 노드 간의 거리를 정의하고 이에 해당되는 임베딩 맵을 구하면, 이차원이나 삼차원으로 차원을 크게 줄여 오차 에너지가 80-90%를 상회하여도 임베딩 공간에서 각 신호 고유의 기하 구조로 변환 가능함을 보인다. 즉, 삼차원 이하로 차원을 크게 줄이면 원래의 컴퓨터 타임 거리 정보는 크게 손실되지만 그 신호에 내재한 고유의 기하 구조를 생성해 내므로 클러스터링이나 패턴 분류 등을 수행하기 위한 기준으로는 정보 압축을 충실히 수행할 수 있는 장점이 있다.

본 논문의 구성은 다음과 같다. 2장에서는 스펙트럴 그래프 이론을 리뷰하고 본 논문에서 구현한 패치 그래프를 3장에 설명한다. 4장에서는 그래프로 구성된 랜덤 워크에서 구한 확산 거리(diffusion distance) [8]를 이용하여 컴퓨터 타임을 구하는 방법을 간략히 소개하고 이에 따라 저차원 유클리드 공간으로 임베딩하는 방법을 설명한다. 파형 신호와 이미지 등에서 패치를 추출하고 이를 그래프로 구성한 다음, 이로부터 각 패치 간의 컴퓨터 타임을 구하여 이에 기반한 임베딩 기법을 구현하고, 가장 널리 이용되는 PCA 임베딩 결과와 비교 분석한 결과 등을 5장에 제시한다. 마지막으로 6장에서는 결론을 맺고 향후

연구 진행방향에 대하여 논의한다.

2. 스펙트럴 그래프 이론

그래프를 기본적으로 이해하기 위해서는 그래프 라플라시안 행렬의 고유값이 중심적인 역할을 한다고 볼 수 있는데, 이 고유값들을 그래프의 스펙트럼이라고 부른다. 스펙트럴 그래프 이론의 주 목적은 그래프의 스펙트럼으로부터 그래프의 구조와 주요 특성을 연역해 내는 것이다. 영상 분할, 데이터 클러스터링 등의 분야에서 널리 이용되고 있는 이 이론은 그래프 라플라시안 행렬에서 구한 고유치와 고유벡터를 이용하여 그래프의 구조적인 특성을 분석한다. 그래프를 분석하는데 있어서 가장 중요한 일은 각 노드를 연결하는 에지를 통하여 시간에 따라 정보가 어떻게 전파되는지를 파악하는 것이다. 본 논문의 목적은 데이터를 클러스터링하거나 임베딩하는데 컴퓨터 타임이 어떻게 이용되는지를 연구하는 것이다. 이를 위하여 이 절에서는 스펙트럴 그래프 이론을 간략히 리뷰한다.

그래프가 주어지면 우선 각 노드 간의 유사도가 결정되어야 한다. 이에 대한 설명은 뒤에서 다시 논의하기로 한다. 두 노드 u 와 v 간의 유사도 $w(u,v)$ 를 알면 각 노드의 정도(degree) $d_u = \sum_v w(u,v)$ 를 구할 수 있다. 여기서 $w(u,v)=w(v,u)$ 이라고 가정한다. 이 때, 그래프 라플라시안은 다음과 같이 정의된다 [9].

$$L(u,v) = \begin{cases} d_u & u = v \\ -w(u,v) & \text{otherwise} \end{cases}$$

L 은 대칭 행렬이고, 이의 수학적 분석 등을 용이하게 하기 위하여 다음과 같이 정규화하여 이용한다.

$$L_{sym} = D^{-\frac{1}{2}} L D^{-\frac{1}{2}} = I - D^{-\frac{1}{2}} W D^{-\frac{1}{2}}$$

여기서 D 는 d_u 를 u 번째 원소로 정의한 대각 행렬이다. $d_u=0$ 이면 노드 u 는 다른 노드와 연결성을 갖고 있지 않음을 나타내는데, 이 때 $D^{-1}(u,u)=0$ 으로 정의한다. L_{sym} 도 대칭 행렬이므로 그 고유값은 모두 실수이고 0을 포함한 양수이다. 이를 확인하기 위하여 다음과 같이 레일리 몫(Rayleigh quotient)을 이용할 수 있다[9].

$$\begin{aligned} \frac{\langle g, L_{sym}g \rangle}{\langle g, g \rangle} &= \frac{\langle g, D^{-1/2} L D^{-1/2} g \rangle}{\langle g, g \rangle} = \frac{\langle f, Lf \rangle}{\langle D^{1/2} f, D^{1/2} f \rangle} \\ &= \frac{\frac{1}{2} \sum_{u,v} (f(u)-f(v))^2 w(u,v)}{\sum_v f(v)^2 d_v} \end{aligned}$$

여기서 g 는 열 벡터로 간주하고 $g = D^{1/2} f$ 이다. 위 식으로부터 모든 고유값은 0을 포함한 양수임을 알 수 있다. 또한 $\mathbf{1} = [1, \dots, 1]^T$ 으로 정의하면 $D^{1/2} \mathbf{1}$ 은 고유값이 0일 때 L_{sym} 의 고유벡터임을 확인할 수 있다. 그래프의 노드 수가 n 일 때, L_{sym} 의 고유값은 $0 = \lambda_1 \leq \dots \leq \lambda_n$ 으로 나타낼 수 있다. 이에 더하여,

$$\lambda_2 = \min_{f \perp D \mathbf{1}} \frac{\frac{1}{2} \sum_{u,v} (f(u)-f(v))^2 w(u,v)}{\sum_v f(v)^2 d_v}$$

이고, 이 때의 고유 벡터는 $g = D^{1/2} f$ 으로 주어지는데, 이는 리만 다양체(Riemannian manifold)의 라플라스-벨트라미(Laplace-Beltrami) 작용소의 고유값에 해당된다[7,9].

$$\lambda_M = \infty \frac{\int_M |\nabla f|^2}{\int_M |f|^2}$$

3. 스펙트럴 그래프 구성

각 쌍의 데이터 간의 거리(또는 유사도 등)가 주어지면, 동일한 그룹에 속하는 데이터 사이에는 거리가 작고 서로 다른 그룹 간에는 거리가 크도록 데이터를 나누면 간단히 이 데이터들을 클러스터링할 수 있다. 이를 더 정교하게 하기 위하여 각 데이터를 노드로 지정하고, 두 노드 간의 유사도가 주어진 값보다 크면 두 노드를 연결하고 연결 강도에 양수를 할당하는 그래프를 구성한다.

본 논문에서는 파형 신호나 이미지 등에서 추출한 패치를 그래프의 노드로 이용한다. 패치는 오래 전부터 패턴 분류, 잡음 제거, 신호 검출 등에 널리 이용되고 있다[10,11]. 패치는 일반적으로 다음과 같이 신호에서 추출한 벡터로 정의된다.

$$\mathbf{x}_n = (x[n], x[n+1], \dots, x[n+d-1])^T \in \mathbb{R}^d \quad (1)$$

이미지에서는 $\sqrt{d} \times \sqrt{d}$ 영역을 추출한 다음 이를 열로 나열하여 벡터화시킴으로써 패치를 구한다.

여기서 각 패치는 서로 겹치게 함으로써 이 들 간에 상관관계를 갖도록 한다. 이와 같이 추출한 패치들을 모두 모은 패치세트 $\{\mathbf{x}_n, n=1, \dots, N\}$ 가 d 차원 상의 비선형 다양체를 이산화한 점의 집합으로 가정할 때, 다양체 위의 두 점이 서로 근방에 위치하면 이 들 간의 측지거리는 유클리드 거리로 근사화할 수 있다. 하지만 두 점이 서로 멀리 떨어져 있으면, 다양체의 곡률로 인하여 측지거리와 유클리드 거리 사이에는 큰 오차가 발생하여 측지 거리를 측정하는 것이 사실 상 불가능하다. 본 논문에서는 이러한 특성을 반영하여 k -NN(k -nearest neighbor)방식을 택한다. 이 방식은 \mathbf{x}_v 가 \mathbf{x}_u 에 가장 가까운 k 개의 인근 패치에 속하거나 \mathbf{x}_u 가 \mathbf{x}_v 에 가장 가까운 k 개의 인근 패치에 속하면 \mathbf{x}_u 와 \mathbf{x}_v 를 연결시킨다. 이 때, 패치 그래프에서 패치 \mathbf{x}_u 와 \mathbf{x}_v 간의 가중치 $w(u, v)$ 는 다음과 같이 구한다[12].

$$w(u, v) = \begin{cases} -\frac{\delta(\mathbf{x}_u - \mathbf{x}_v)^2}{\sigma^2} & \mathbf{x}_u, \mathbf{x}_v: \text{connected} \\ 0 & \text{otherwise} \end{cases}$$

여기서 $\delta(\mathbf{x}_u - \mathbf{x}_v) = \left\| \frac{\mathbf{x}_u}{\|\mathbf{x}_u\|} - \frac{\mathbf{x}_v}{\|\mathbf{x}_v\|} \right\|$ 이다.

$\delta(\mathbf{x}_u - \mathbf{x}_v)$ 는 그 크기로 정규화된 두 패치 \mathbf{x}_u 와 \mathbf{x}_v 간의 거리를 나타내고, σ 는 $\delta(\cdot)$ 에 따른 $w(u, v)$ 의 크기를 제어한다. 패치를 그 크기로 정규화하면 $\mathbf{x}_u \in R^d$ 를 S^{d-1} 으로 매핑함으로써 차원을 1 만큼 줄이는 효과를 얻을 수 있다. 여기서 S^{d-1} 는 $d-1$ 차원 구(sphere)를 나타낸다. 또한 σ 가 $\delta(\cdot)$ 보다 크면 $w(u, v) \approx 1$ 이지만, 그렇지 않으면 $w(u, v)$ 는 급속히 0으로 감소하게 되므로 결국 σ 는 패치 그래프의 확산속도를 제어하는 효과가 있다. 반면에 σ 가 매우 작으면 $\delta(\cdot)$ 가 작은 값을 갖더라도 $w(u, v) \approx 0$ 이므로 잡음에 매우 민감한 영향을 받는다. 따라서 본 논문에서는 σ 를 충분히 큰 값으로 지정한다.

3.1 그래프 라플라시안 행렬

그래프는 기본적으로 무향(undirected)이고 가중 행렬 W 는 대칭 즉, $w(u, v) = w(v, u)$ 이라고 가정한다. 그래프 라플라시안 행렬 L 은 $D - W$ 으로 정의되는데, 여기서 D 는 각 원소가 $d_u = \sum_v w(u, v)$ 인 대각행렬이다. L 은 positive semidefinite이고 그 고유치는

항상 0 이상의 값을 갖는다. 또한, 그래프가 완전 연결일 때, 0의 고유값이 존재하고 이에 해당되는 고유 벡터는 $\mathbf{1} = [1, \dots, 1]^T$ 이다. 일반적으로 행렬 L 을 정규화하기도 하는데, L_{sym} 또는 다음과 같은 방법으로 정규화시킨다.

$$L_{rw} = D^{-1}L = I - D^{-1}W$$

L_{rw} 의 고유값과 고유벡터가 각각 λ, v 즉, $L_{rw}v = \lambda v$ 이면 L_{sym} 의 고유치는 L_{rw} 의 것과 동일한 λ 이고 해당 고유벡터는 $D^{1/2}v$ 즉, $L_{sym}D^{1/2}v = \lambda D^{1/2}v$ 이 성립한다. 또한 $Lv = \lambda Dv$ 이면 L_{rw} 의 고유값과 고유 벡터는 각각 λ, v 이다. 이러한 특성을 이용하면 정규 또는 비정규 라플라시안 행렬을 이용하여 그래프를 다양하게 해석할 수 있는 장점이 있다.

4. 컴뮷트 타임 거리

그래프에서의 랜덤워크는 노드에서 노드로 랜덤하게 이동하는 랜덤 프로세스를 말한다. 여기서 노드 \mathbf{x}_i 에서 노드 \mathbf{x}_j 으로 이동할 전이 확률(transition probability)은 $p_{ij} = w(i, j)/d_i$ 으로 주어질 수 있으므로 랜덤 워크의 전이 확률 행렬 $P = (p_{ij})_{i, j=1, \dots, N}$ 는 다음과 같이 정의된다.

$$P = D^{-1}W = I - L_{rw}$$

따라서, L_{rw} 의 고유값이 λ 이면 P 의 고유값은 $1 - \lambda$ 이며, 이 때 L_{rw} 와 P 는 동일한 고유 벡터를 갖는다. 다시 말하면, L_{rw} 의 최소 고유값은 P 의 최대 고유값과 일치하고, 이에 해당되는 고유벡터는 그 그래프의 특성을 해석하는데 매우 중요한 역할을 한다. 그래프 상의 랜덤 워크에서 평균 히팅 타임(average hitting time) $h(\mathbf{x}_i, \mathbf{x}_j)$ 는 노드 \mathbf{x}_i 에서 출발하여 노드 \mathbf{x}_j 으로 도달하는데 걸리는 평균 시간을 나타낸다. 두 노드 \mathbf{x}_i 와 \mathbf{x}_j 간의 컴뮷트 타임 $c(\mathbf{x}_i, \mathbf{x}_j)$ 은 랜덤 워크가 \mathbf{x}_i 에서 \mathbf{x}_j 으로 이동한 다음 다시 \mathbf{x}_i 으로 되돌아오는데 소요되는 평균시간으로 정의된다. 즉, $c(\mathbf{x}_i, \mathbf{x}_j) = h(\mathbf{x}_i, \mathbf{x}_j) + h(\mathbf{x}_j, \mathbf{x}_i)$ 이다. 그래프에서 최단 거리(측지 거리)와는 달리, 두 노드 간의 컴뮷트 타임은 두 노드를 연결하는 경로가 많을수록 감소한다. 따라서, 동일한 클러스터에 속한 노드 간

의 컴퓨터 타임은 작은 값을 갖는 반면에, 서로 다른 클러스터에 속한 노드 사이에는 매우 큰 컴퓨터 타임을 가지는 특성이 있다. 컴퓨터 타임은 다음과 같이 주어지는 확산거리(diffusion distance)와 밀접한 관련이 있다.

$$D_t^2(x_i, x_j) = \|p_t(x_i, \cdot) - p_t(x_j, \cdot)\|^2 = \sum_{k=2}^n \gamma_k^{2t} (\psi_{ik} - \psi_{jk})^2$$

여기서 ψ_{ik} 는 ψ_k 의 i 번째 원소이고 γ_k 와 ψ_k 는 P 의 k 번째 고유값과 고유 벡터 즉, $P\psi_k = \gamma_k\psi_k$ 이다. 이는 $L_{rw}\psi_k = (1-\gamma_k)\psi_k$ 를 만족시킨다. 이와 같이 확산 거리가 주어지면, 두 노드 x_i 와 x_j 간의 컴퓨터 타임 $c(x_i, x_j)$ 는 다음과 같이 구할 수 있다[12,13].

$$c(x_i, x_j) = vol \sum_{t=0}^{\infty} D_{t/2}^2(x_i, x_j) \quad (2)$$

$$= vol \sum_{k=2}^n \frac{1}{1-\gamma_k} (\psi_{i,k} - \psi_{j,k})^2$$

$$= vol \sum_{k=2}^n \frac{1}{\hat{\lambda}_k} \left(\frac{v_{i,k}}{\sqrt{d_i}} - \frac{v_{j,k}}{\sqrt{d_j}} \right)^2$$

여기서 v_{ik} 는 L_{sym} 의 고유벡터 v_k 의 i 번째 원소이고 $vol = \sum d_i$, $\hat{\lambda}_k = 1-\gamma_k$ 이다. 또한, 고유값 $\hat{\lambda}_k$ 는 다음과 같이 정렬되어 있다고 가정한다.

$$0 = \hat{\lambda}_1 \leq \hat{\lambda}_2 \leq \dots \leq \hat{\lambda}_n < 2 \quad (3)$$

그림 1-(c)는 1,376 개의 패치로 구성된 그래프에서 구한 L_{sym} 의 고유값을 오름 차순으로 정렬하여 그린 그림이다. 이 그림을 통하여 식 (3)을 실험적으로 확인할 수 있다. 여기서 주의할 점은 $\hat{\lambda}_1 \neq 0$ 이면 그래프가 완전 연결되어 있지 않은 상태이므로 NN(nearest neighbor)의 수를 늘려 $\hat{\lambda}_1 = 0$ 가 되도록 확인하여야 한다.

4.1 컴퓨터 타임 임베딩

컴퓨터 타임은 데이터를 임베딩하는데 활용할 수 있다. 여기서 거리공간 X 에서 거리공간 Y 의 부분공간 S 으로 매핑하는 위상동형(homeomorphism)이 존재하면 이를 임베딩이라고 부른다. 즉, 각 노드를 S 에서 좌표화하는 작업을 말한다. 위 식 (2)를 관찰하면 $\sqrt{c(x_i, x_j)}$ 는 R^{n-1} 상의 두 벡터 간의 거리로 해석될 수 있다. 다시 말해서 x_i 를 다음과 같이 임베딩하면

$$\Psi: x_i \rightarrow \sqrt{vol} \left(\frac{v_{k,2}}{\sqrt{\hat{\lambda}_2 d_i}}, \dots, \frac{v_{k,N}}{\sqrt{\hat{\lambda}_N d_i}} \right) \quad (4)$$

$\sqrt{c(x_i, x_j)}$ 는 그래프의 두 노드 x_i 와 x_j 간의 유클리드 거리로 간주될 수 있는데, 이를 컴퓨터 타임 거리라고 부른다. 여기서, L_{sym} 의 고유벡터 v_k 의 인덱스를 2부터 시작하는 이유는 v_1 에 해당되는 고유값 $\hat{\lambda}_1$ 이 0의 값을 갖기 때문이다. 그런데 임베딩 공간의 차원은 노드 수에 따라 증가하므로 어느 정도의 오차를 허용하는 범위 내에서 차원을 κ 으로 줄이면 다음과 같이 정의된 임베딩을 이용할 수 있다[13].

$$\Phi: x_i \rightarrow \sqrt{vol} \left(\frac{v_{k,2}}{\sqrt{\hat{\lambda}_2 d_i}}, \dots, \frac{v_{k,\kappa+1}}{\sqrt{\hat{\lambda}_{\kappa+1} d_i}} \right) \quad (5)$$

특히, 임베딩 차원 κ 를 3 이하로 줄이면 시각화가 가능하므로 본 논문에서는 $\kappa = 3$ 으로 고정하여 임베딩한다. 예를 들어, 그림 1-(a)에 주어진 바와 같이 1,400 개 샘플 길이의 chirp 신호로부터 $d = 25$ 인 패치를 식 (1)과 같이 구성하여 식 (5)의 방식으로 컴퓨터 타임 임베딩하면 그림 1-(b)의 결과를 얻을 수 있다. 각 패치에서 그레이던트 벡터의 크기를 각각 구하고 이를 오름 차순으로 정렬하여 중간 값보다 작으면 파란 점으로, 크면 빨간 점으로 나타낸다. 즉, 파란 점은 부드러운 저주파 영역의 패치에 해당되는 반면, 빨간 점은 고주파 영역을 나타낸다. 이는 파형 신호를 패치화함으로써 삼차원 공간 상의 곡선 또는

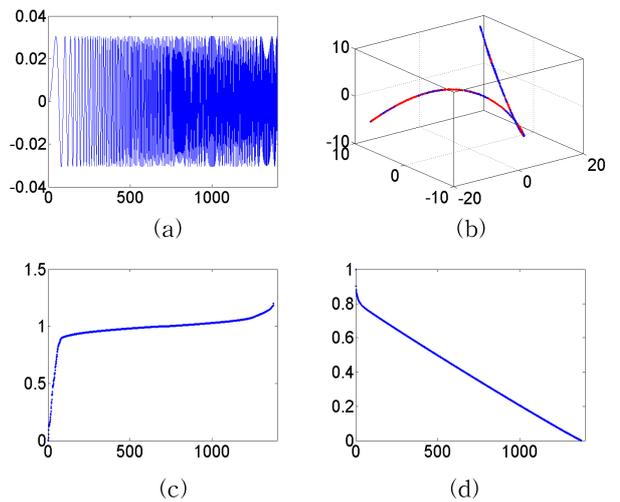


그림 1. chirp 신호의 컴퓨터 타임 임베딩 (a) chirp 신호 (b) 임베딩 결과 (c) 정렬된 고유값의 변화(d) 축소된 차원 κ 에 따른 근사화 오차의 변화

기하구조로 표현 가능함을 보여 준다. 그림 1-(d)는 식 (5)로 근사화하였을 때의 오차를 나타낸 것이다. 이 그림에서 알 수 있듯이, 임베딩 차원을 3으로 줄이면 오차율이 85%를 상회하므로 수치적으로는 근사화된 임베딩이 컴류트 타임 거리를 제대로 보존하지는 못하지만 각 신호에 고유한 기하구조를 생성해 내므로 클러스터링이나 패턴 분류 등을 수행하기 위한 기준으로는 정보 압축을 충실히 수행할 수 있는 장점이 있다.

4.2 컴류트 타임 임베딩의 수학적 분석

컴류트 타임 임베딩은 다음과 같은 목적함수를 최소화하는 매핑을 구하는 것이다[9,12].

$$\epsilon = \frac{\sum_{i,j} \|y_i - y_j\|^2}{\sum_{i,j} y_i^T y_j d_i} = tr\left(\frac{Y^T LY}{Y^T DY}\right)$$

여기서 tr 은 행렬의 대각 성분을 모두 합한 값을 나타낸다. 임베딩 문제는 다음 식을 만족하는 해를 구하는 것으로 귀결된다.

$$\hat{Y} = \operatorname{argmin} tr\left(\frac{Y^T LY}{Y^T DY}\right)$$

위 식에서 괄호 안의 식을 $Z = D^{1/2} Y$ 으로 치환하면

$$\frac{Y^T LY}{Y^T DY} = \frac{Z^T D^{-1/2} L D^{-1/2} Z}{Z^T Z} = \frac{Z^T L_{sym} Z}{Z^T Z}$$

이므로 Z 가 L_{sym} 의 가장 작은 고유값에 대응되는 고유 벡터일 때 $tr\left(\frac{Y^T LY}{Y^T DY}\right) = tr\left(\frac{Z^T L_{sym} Z}{Z^T Z}\right)$ 의 최소값을 얻을 수 있다. 다시 말해서 $Y = D^{-1/2} V$ 일 때 ϵ 은 최소값을 갖는다. 여기서 $V = [v_2, \dots, v_{\kappa+1}]$ 으로 v_m 은 L_{sym} 의 m 번째 고유 벡터이다. 이는

$$Y = \sqrt{\operatorname{vol}} D^{-1/2} V(\hat{\Lambda}^\dagger)^{1/2} \tag{6}$$

으로 재정의하여도 동일한 결과를 얻는다. 여기서 $\hat{\Lambda}$ 는 L_{sym} 의 고유값 $\hat{\lambda}_m$ 을 대각 원소로 하는 대각 행렬이고 $\hat{\Lambda}^\dagger$ 는 $\hat{\Lambda}$ 의 Moore-Penrose 역으로 다음과 같이 정의된다.

$$\hat{\Lambda}^\dagger = \operatorname{diag} \begin{cases} \frac{1}{\hat{\lambda}_i} & \hat{\lambda}_i \neq 0 \\ 0 & \hat{\lambda}_i = 0 \end{cases}$$

즉, 식 (6)을 $\frac{Y^T LY}{Y^T DY}$ 에 대입하면

$$\begin{aligned} \frac{Y^T LY}{Y^T DY} &= \frac{\operatorname{vol}(\hat{\Lambda}^\dagger)^{1/2} V^T D^{-1/2} L D^{-1/2} V(\hat{\Lambda}^\dagger)^{1/2}}{\operatorname{vol}(\hat{\Lambda}^\dagger)^{1/2} V^T D^{-1/2} D D^{-1/2} V(\hat{\Lambda}^\dagger)^{1/2}} \\ &= \frac{(\hat{\Lambda}^\dagger)^{1/2} V^T L_{sym} V(\hat{\Lambda}^\dagger)^{1/2}}{(\hat{\Lambda}^\dagger)^{1/2} V^T V(\hat{\Lambda}^\dagger)^{1/2}} \end{aligned}$$

인데, $V^T V = I$ 이고 $V^T L_{sym} V = \hat{\Lambda}$ 이므로

$$\frac{Y^T LY}{Y^T DY} = \frac{(\hat{\Lambda}^\dagger)^{1/2} \hat{\Lambda} (\hat{\Lambda}^\dagger)^{1/2}}{\hat{\Lambda}^\dagger} = \hat{\Lambda} \text{ 이다. 따라서, 식 (6)의}$$

Y 는 결국 식 (5)의 벡터 행렬 식과 일치하고, ϵ 의 최소값은 $tr(\hat{\Lambda})$ 를 갖는다.

5. 실험 및 토론

본 논문에서는 파형 신호와 이미지에서 패치를 추출하고 이들로부터 패치그래프를 구성하여 실험한다. 모든 패치의 차원은 25차로 고정하고 이러한 고차원 데이터를 시각화하기 위하여 삼차원 공간으로 사영시킨다. 우선, 신호의 상관관계에 따른 컴류트 타임 임베딩의 특성을 알아 보기 위하여 다양한 대역폭의 저역 필터로 가우시안 잡음 신호를 전 처리한 다음 임베딩의 변화 과정을 그림 2에 제시하였다.

이 그림에서 알 수 있는 바와 같이, 파형 신호의 상관관계가 증가함에 따라 그 신호 고유의 기하 구조 즉, 곡선 형태가 생성된다. 그 원인은 패치들 간의 상관 관계가 낮을 수록 이들의 컴류트 타임 거리가 랜덤한 값을 가져 흩트러진 점의 형태로 임베딩되지만, 상관 관계가 증가함에 따라 이들의 컴류트 타임 거리가 비교적 고르게 분포되어 연속적인 임베딩을 갖기 때문이다. 이는 랜덤한 신호일수록 저차원 신호로 압축할 수 없음을 의미한다.

고차원 데이터 x_1, \dots, x_n 가 주어졌을 때, 이를 저차원 공간으로 임베딩하는 가장 보편적인 방법은 주성분 분석(PCA)을 적용하는 것이다. 그림 3은 700 개의 샘플로 구성된 정현파 신호, 1,000 개 샘플의 음성 신호, 1,500 개 샘플의 바이올린 음에 대한 PCA 임베딩과 컴류트 타임 임베딩을 각각 보여주고 있다. 여기서, 정현파 같은 주기 신호는 주기에 따라 패치가 반복되어 나타나므로 파형의 길이는 기본적으로 한 주기를 포함하면 충분하다. 정현파 신호에서는 25 차원 벡터 크기의 패치를 676 개 추출하여 위와 동

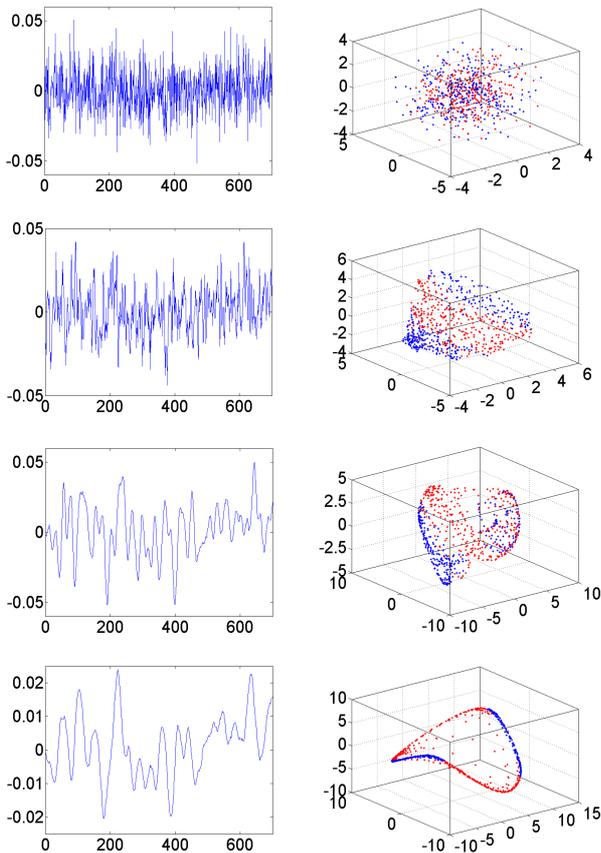


그림 2. 가우시안 잡음 신호의 대역폭에 따른 컴퓨터 타임 임베딩의 변화. 1행: 원 신호, 2행: 차단 주파수 0.1, 3행: 차단 주파수 .05, 4행: 차단 주파수 0.03

일한 방법으로 실험하였다. PCA 임베딩에서는 새로운 임베딩 공간에서 랜덤하게 분포하여 삼차원 공간으로 표현하는 것이 무의미하다. 다시 말하면, 이 패치들이 놓여 있는 다양체는 선형적인 구조가 아닌, 곡률이 있는 복잡한 기하 구조를 갖고 있어서 25 차원 벡터를 삼차원 벡터로 압축하기에는 큰 무리가 있음을 나타낸다. 반면에 컴퓨터 타임 임베딩에서는 부드러운 곡선 구조를 갖고 있는데 파란 점과 빨간 점이 각각 클러스터링되어 있는 모습을 볼 수 있다. 여기서 빨간 점들은 정현파의 단조 증가나 단조 감소 영역에서 추출한 것이고 파란 점들은 극점을 포함하는 영역에서 추출된 것이다. 컴퓨터 타임 임베딩 결과를 위에서 아래 방향으로 사영시켜 이차원으로 임베딩하면 원의 구조임을 충분히 예측할 수 있으므로 이차원 벡터로 압축하여도 기본적인 정보를 그대로 보존하고 있음을 알 수 있다. 그 외의 음성 파형, 바이올린 악기 음 신호 등에서도 위와 마찬가지로 PCA 임베딩보다 컴퓨터 타임 임베딩에서 단순하고 명확

한 기하 구조를 갖고 있음을 확인할 수 있다.

이 그림에서 특이한 점은 정현파와 바이올린 신호의 컴퓨터 타임 임베딩은 기본적으로 동일한 기하 구조를 갖고 있는데, 이는 현악기 소리의 특성이 기본적으로 정현파적이기 때문일 것으로 예상할 수 있다. 비올라나 첼로 소리에 대해서도 매우 유사한 구조를 갖고 있음을 확인할 수 있다. 또한, 음성 파형 '아'와 '우'의 임베딩은 서로 다른 고유의 기하 구조를 갖고 있다. 이는 음성 인식, 화자 인식 등의 패턴 분류를 기하학적으로도 접근 가능함을 예시한다고 판단할 수 있다.

그림 4는 이미지에서 5×5 패치를 추출하여 25 차원의 벡터로 변환한 다음 PCA와 컴퓨터 타임 임베딩을 수행하여 삼차원 공간으로 사영시켜 임베딩한 결과를 보여 주고 있다. 이 그림에서도 마찬가지로 파란 점은 부드러운 저주파 영역의 패치에 해당되는 반면, 빨간 점은 에지나 텍스처 등의 고주파 영역을 나타낸다. PCA 임베딩에서 파란 점은 원점 부근에 집중적으로 나타나지만, 빨간 점은 그 주변에 흐트러져 있다. 이에 비하여 그림 4의 2행에 제시한 컴퓨터 타임 임베딩은 파란 점이 어떤 곡면을 따라 정렬된 반면, 빨간 점은 PCA 임베딩에 비하여 매우 조밀하게 분포되어 있음을 확인할 수 있다.

이 실험을 통하여 파형 신호나 이미지 등에서 추출한 패치를 이용하여 컴퓨터 타임 임베딩하면 파형 신호에서 추출한 패치는 일차원 곡선을 따라 임베딩되고 이미지 패치는 이차원 곡면을 따라 임베딩되어 그 신호에 고유한 기하 구조를 갖고 있음을 알 수 있다. 그리고 컴퓨터 타임 임베딩이 PCA 보다 정보 압축 능력이 탁월하여 데이터의 차원을 크게 줄일 수 있음을 기대할 수 있다. 즉, 임베딩 차원을 급격히 줄여서 오차율이 크게 증가하여도 본 논문에서 제안한 컴퓨터 타임 임베딩은 각 신호에 고유한 기하구조를 생성해 내므로 클러스터링이나 패턴 분류 등을 수행하기 위한 기준으로는 정보 압축을 충실히 수행할 수 있는 장점이 있다.

이 실험을 통하여 파형 신호나 이미지 등에서 추출한 패치를 이용하여 컴퓨터 타임 임베딩하면 파형 신호에서 추출한 패치는 일차원 곡선을 따라 임베딩되고 이미지 패치는 이차원 곡면을 따라 임베딩되어 그 신호에 고유한 기하 구조를 갖고 있음을 알 수 있다. 그리고 컴퓨터 타임 임베딩이 PCA 보다 정보 압축 능력이 탁월하여 데이터의 차원을 크게 줄일 수 있음을 기대할 수 있다. 즉, 임베딩 차원을 급격히 줄여서 오차율이 크게 증가하여도 본 논문에서 제안한 컴퓨터 타임 임베딩은 각 신호에 고유한 기하구조를 생성해 내므로 클러스터링이나 패턴 분류 등을 수행하기 위한 기준으로는 정보 압축을 충실히 수행할 수 있는 장점이 있다.

6. 결 론

본 논문에서는 고차원 유클리드 공간에 임베딩된 저차원 비선형 다양체 위에 놓인 데이터의 차원을 줄이기 위한 다양체 학습 기법에 대하여 알아보았다.

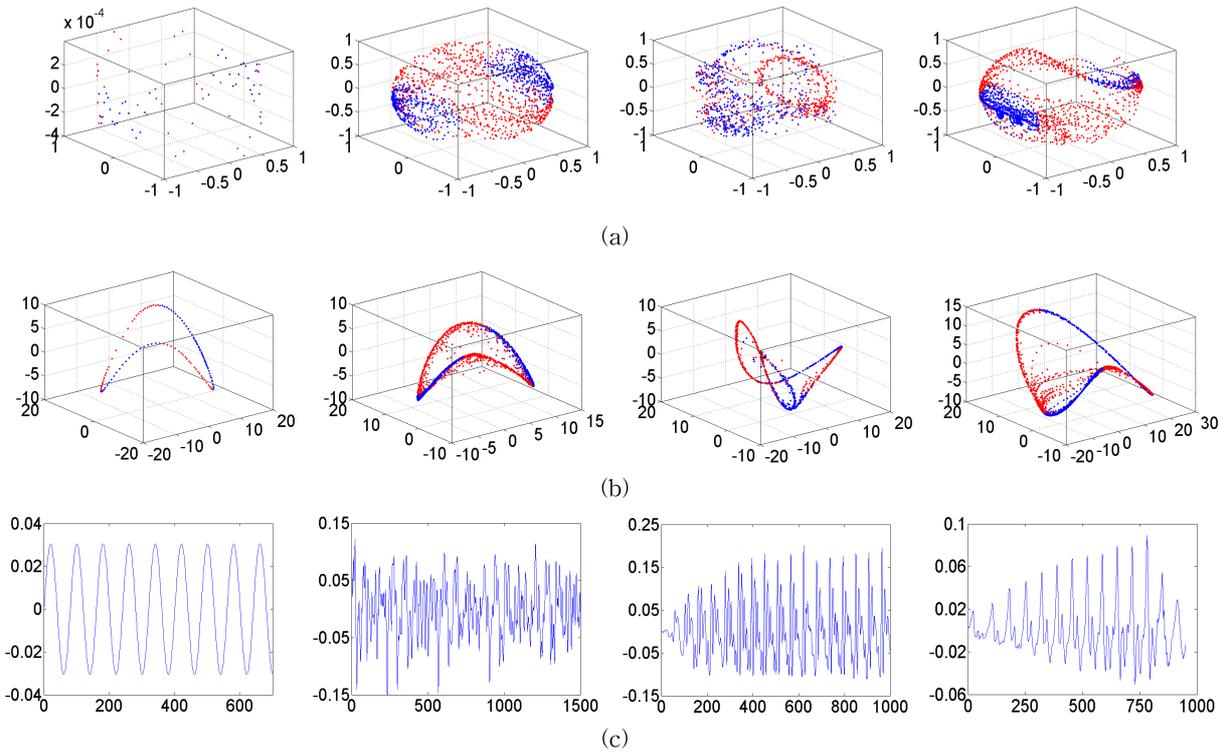


그림 3. 파형 신호에 대한 PCA 및 커뮤트 타임 임베딩 (a) PCA 임베딩, (b) 커뮤트 타임 임베딩, (c) 입력 신호
 1열: 정현파, 2열: 바이올린 음, 3열: 음성 파형('아'), 4열: 음성 파형('우')

이를 위하여 파형 신호와 이미지 등에서 패치를 추출하고 이를 그래프로 구성한 다음, 이로부터 각 패치

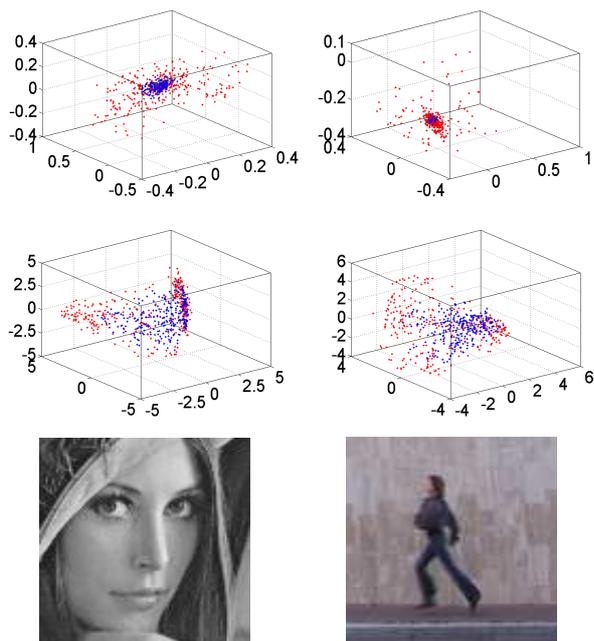


그림 4. 이미지에 대한 PCA 및 커뮤트 타임 임베딩. 1행: PCA 임베딩, 2행: 커뮤트 타임 임베딩, 3행: 입력 이미지(100×100)

간의 커뮤트 타임을 구하여 이에 기반한 임베딩 기법을 구현하고, 가장 널리 이용되는 PCA 임베딩 결과와 비교 분석하였다. PCA 임베딩에서는 입력 데이터가 선형 부공간에 집중되어 분포되어 있으면 이에 해당되는 공분산 행렬에서 구한 주 고유벡터 방향으로 사영시킴으로써 효과적으로 차원을 줄일 수 있으나, 그 데이터가 놓여 있는 공간이 비선형이면 저차원 공간으로의 임베딩은 정보 손실이 커서 사실상 무의미하다. 이를 해결하기 위하여 여러 가지의 다양체 학습 기법이 제안되고 있는데, 커뮤트 타임 임베딩이 가장 우수한 성능을 나타내는 것으로 알려져 있다. 본 논문에서는 파형 신호나 이미지에서 패치를 추출하여 패치 그래프를 구성한 후 커뮤트 타임 임베딩하면, 삼차원 이하로 크게 압축하여 근사화 오차가 80~90%를 상회하여도 그 신호에 고유한 기하 구조를 생성해 내므로 패턴 분류 등의 응용 분야에 효과적으로 적용 가능함을 보여 주었다.

이 기법을 통하여 입력 데이터 열이 구성하는 다양체의 기하 구조를 근사적으로 확인할 수는 있으나 이를 응용 분야에 접목시키기에는 해결하여야 할 난제가 많다. 예를 들어, 그래프의 노드 수에 따라 다양

체의 구조가 변하기 때문에 주어진 신호에 적합한 노드 수를 결정하고, 각 노드에 대한 인근 노드를 지정하는 방법 등에 대한 개선이 필요하다. 무엇보다도 중요한 것은 신호의 특성에 따라 노드 간의 유사도를 측정하는 기법이 달라져야 하는데 이에 대한 기초 연구가 선행되어야 할 것이다. 향 후에는 이에 대한 연구와 함께 계산량을 줄이는 작업을 시행할 계획이다.

참 고 문 헌

[1] H. Hahn, "Proposing a Connection Method for Measuring Differentiation of Tangent Vectors at Shape Manifold," *Journal of Korea Multi-media Society*, Vol. 16, No. 2, pp. 160-168, 2013.

[2] J.B. Tenenbaum, V. deSilva, and J.C. Langford, "A Global Geometric Framework for Nonlinear Dimensionality Reduction," *Science*, Vol. 290, pp. 2319-2323, 2000.

[3] T.F. Cox and M.A.A. Cox, *Multidimensional Scaling*, Chapman and Hall/CRC, 2001, USA.

[4] S.T. Roweis and L.K. Saul, "Nonlinear Dimensionality Reduction by Locally Linear Embedding," *Science*, Vol. 290, pp. 2323-2326, 2000.

[5] D.L. Donoho and C. Grimes, "Hessian Eigenmaps: New Locally Linear Embedding Techniques for High Dimensional Data," *Proc. the National Academy of Sciences*, pp. 5591-5596, 2003.

[6] M. Belkin and P. Niyogi, "Laplacian Eigenmaps and Spectral Techniques for Embedding and Clustering," *Advances in Neural Information Processing Systems*, Vol. 14, pp. 585-591, 2001.

[7] M. Belkin and P. Niyogi, "Laplacian Eigenmaps for Dimensionality Reduction and Data Representation," *Neural Computation*, Vol. 15, No. 6, pp. 1373-1396, 2003.

[8] R.R. Coifman and S. Lafon, "Diffusion Maps," *Applied and Computational Harmonic Analysis*, Vol. 21, pp. 5-30, 2006.

[9] F. Chung, *Spectral Graph Theory*, American Mathematical Society, 1997, USA.

[10] K.S. Ni and T.Q. Nguyen, "A Model for Image Patch-Based Algorithms," *ICIP2008*, pp. 2588-2591, 2008.

[11] M. Zontak and M. Irani, "Internal Statistics of a Single Natural Image," *CVPR2011*, pp. 977-984, 2011.

[12] K.M. Taylor, *The Geometry of Signal and Image Patch-Sets*, Ph D Thesis of University of Colorado, 2011.

[13] H. Qiu and E.R. Hancock, "Clustering and Embedding using Commute Times," *IEEE Trans. PAMI*, Vol. 29, No. 11, pp. 1873-1890, 2007.



한 희 일

1980년 3월~1984년 2월 서울대학교 제어계측 공학과 공학사
 1984년 3월~1986년 2월 서울대학교 제어계측 공학과 공학석사

1992년 8월~1995년 12월 University of Arizona 전기및 컴퓨터 공학과 공학박사
 1987년 1월~1998년 3월 한국전자통신연구원, 선임연구원
 1998년 3월~현재 한국외국어대학교 정보통신공학과 교수
 관심분야: 신호처리, 영상처리, 컴퓨터비전, 패턴인식