Event-Centric Twitter Photo Summarization

by

Chung-Lin Wen

M.S., National Taiwan University (2009)
B.B.A., National Taiwan University (2006)

Submitted to the the Program in Media Arts and Sciences,
School of Architecture and Planning
in partial fulfillment of the requirements for the degree of

Master of Science

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

June 2014

© Massachusetts Institute of Technology 2014. All rights reserved.
Event-Centric Twitter Photo Summarization

by

Chung-Lin Wen

Submitted to the Program in Media Arts and Sciences,
School of Architecture and Planning
on May 9, 2014, in partial fulfillment of the
requirements for the degree of
Master of Science

Abstract

We develop a novel algorithm based on spectral geometry that summarize a photo collection into a small subset that represents the collection well. While the definition for a good summarization might not be unique, we focus on two metrics in this thesis: representativeness and diversity. By representativeness we mean that the sampled photo should be similar to other photos in the data set. The intuition behind this is that by regarding each photo as a "vote" towards the scene it depicts, we want to include the photos that have high "votes". Diversity is also desirable because repeating the same information is an inefficient use of the few spaces we have for summarization. We achieve these seemingly contradictory properties by applying diversified sampling on the denser part of the feature space.

The proposed method uses diffusion distance to measure the distance between any given pair in the dataset. By emphasizing the connectivity of the local neighborhood, we achieve better accuracy compared to previous methods that used the global distance. Heat Kernel Signature (HKS) is then used to separate the denser part and the sparser part of the data. By intersecting the denser part generated by different features, we are able to remove most of the outliers, i.e., photos that have few similar photos in the dataset. Farthest Point Sampling (FPS) is then applied to give a diversified sampling, which produces our final summarization.

The method can be applied to any image collection that has a specific topic but also a fair proportion of outliers. One scenario especially motivating us to develop this technique is the Twitter photos of a specific event. Microblogging services have become a major way that people share new information. However, the huge amount of data, the lack of structure, and the highly noisy nature prevent users from effectively mining useful information from it. There are textual data based methods but the absence of visual information makes them less valuable. To the best of our knowledge, this study is the first to address visual data in Twitter event summarization. Our method’s output can produce a kind of “crowd-sourced news”, useful for journalists as well as the general public.

We illustrate our results by summarizing recent Twitter events and comparing them with those generated by metadata such as retweet numbers. Our results are of at least the
same quality although produced by a fully automatic mechanism. In some cases, because metadata can be biased by factors such as the number of followers, our results are even better in comparison. We also note that by our initial pilot study, the photos we found with high-quality have little overlap with highly-tweeted photos. That suggests the signal we found is orthogonal to the retweet signal and the two signals can be potentially combined to achieve even better results.

Thesis Supervisor: Ramesh Raskar
Title: Associate Professor
Event-Centric Twitter Photo Summarization

by

Chung-Lin Wen

Thesis Advisor .................................................................

Ramesh Raskar
Associate Professor
Program in Media Arts and Sciences

Thesis Reader .................................................................

Deb Roy
Associate Professor
Program in Media Arts and Sciences

Thesis Reader .................................................................

Cesar Hidalgo
Assistant Professor
Program in Media Arts and Sciences
**Acknowledgments**

I would like to express my appreciation to my advisor, Prof. Ramesh Raskar, for his support during my time at MIT. The time spent in the Camera Culture group is the most unique and exciting experience in my life.

I would also like to thank my readers, Prof. Deb Roy and Prof. Cesar Hidalgo, for providing me constructive critiques during the proposal and the completion of this thesis.

In addition, Dr. Dan Raviv, the post doc I work closely with on this project, also offers very helpful advice. Without his support, this project could not have been accomplished.

I also enjoy my discussion with office mate Nikhil Naik very much. It is very valuable to have people working on similar direction in a very diversified lab.

Finally, I would like to thank my family, Yuki and my parents. Yuki, my dearest wife, always support me with endless love and care. My parents always offer me their encouragement and support. I feel blessed to have you as my family and I would like to take this opportunity to express my deepest appreciation.
# Contents

1 Introduction 15
   1.1 Brief review of image collection summarization techniques 19
   1.2 Brief review of Twitter event summarization techniques 21

2 Problem statement 23

3 Spectral Geometry 25
   3.1 Diffusion maps 25
      3.1.1 Similarity graph 26
      3.1.2 Markov process 27
      3.1.3 Sparsified weight matrix 29
      3.1.4 Diffusion maps and diffusion distance 29
   3.2 Heat Kernel Signature and Auto Diffusion Function 31
   3.3 Farthest Point Sampling 33

4 Image Collection Summarization Algorithm 35
   4.1 Data cleaning 35
   4.2 Feature generation and affinity matrix 37
   4.3 Diffusion maps and diffusion distance 38
   4.4 Outliers removal 39
   4.5 Farthest point sampling 40
   4.6 Feature selection and parameter tuning 40
   4.7 Implementation details 41
5 Results and Evaluation

5.1 Evaluation .................................................. 56

5.2 Discussion .................................................. 61
  5.2.1 A crowd-sourcing approach ......................... 62
  5.2.2 Textual data and metadata integration ............. 65
  5.2.3 Near-duplicate photos and its influence .......... 66

6 Conclusion .................................................... 69
List of Figures

1-1 The image search result on Twitter via the query term "#Oscars2014" as of April, 2014. Only a few of them are directly related to the event. ........................................ 17

3-1 The difference between linear and non-linear clustering. Demonstrated by a swiss roll dataset. [Photo credit: scikit-learn.org] ............................................. 26

3-2 TED image scatter plot according to the first two dimension in $Y$ ............. 30

3-3 An example of HKS value. Blue colors indicate lower values while red colors indicate higher values. ................................................................. 32

4-1 System overview .......................................................... 36

4-2 Duplicated Images in the input photo collection collected from Twitter API 36

4-3 Example of outliers and non-outliers ............................................. 39

5-1 Part of the photos from the Media Lab City Car dataset .......................... 44

5-2 The diffusion distance to the first data point in the Media Lab City Car dataset 45

5-3 Data points in the Media Lab City Car dataset in diffusion maps embedding projected to 3D. The color indicates the frame numbers. ......................... 45

5-4 Media Lab City Car data set summarized in ten images .......................... 46

5-5 Part of the photos from TED event dataset ............................................. 47

5-6 Images from TED dataset with lowest and highest HKS value .................. 48

5-7 Summarizing TED 2014 in five photos .............................................. 51

5-8 Part of the photos from the Oscar data set ............................................. 51

5-9 Intersection of lowest 15% HKS value applied on both normalized color and histogram and GIST ................................................................. 54
5-10 Summarizing Oscar 2014 in five photos ........................................ 55
5-11 Photos most retweeted with hashtag ”#TED2014” ............................... 56
5-12 Photos most retweeted with hashtag ”#Oscars2014” ............................. 57
5-13 The result of our method and baseline methods on Oscar data set #2 ...... 60
5-14 A prototype for Rep or Not. A crowd-sourcing approach for image collec-
tion summarization problem .............................................................. 61
5-15 Almost identical images in the Oscar data set ..................................... 66
List of Tables

5.1 The result of comparing proposed method with baseline on two events using Amazon Mechanical Turk .......................... 59
5.2 The result of comparing proposed method with baselines on a larger data set of Oscar photos using Amazon Mechanical Turk ............. 59
Chapter 1

Introduction

With the advance of mobile devices and social networks, the way people exchange information has fundamentally changed. As for news, a few big news media companies are no longer the only sources of information. For instance, people nowadays go to all kinds of social media, micro-blogging sites or blogs to look for the information they are interested in. The ability to broadcast interesting information has been effectively distributed to ordinary people outside the news companies.

Twitter is one of the most popular micro-blogging platforms that enormous number of users have adopted to share information for its realtime-ness compared to other media. The unbiased view of Twitter also made it very ideal for social movement. For instance, the Arab Spring. While potentially containing useful information, the micro-blogging platforms raise several challenges. First, potentially useful information is buried in a significant amount of tweets that are irrelevant, such as mundane daily life sharing or broadcast private discussions. Second, for a detected event, the amount of information might still well exceed the human perceptual capacity to process it in a reasonably short time. Hence, how to effectively detect and summarize events is one of the critical steps toward effective data mining on micro-blogging platforms.

Already much effort has been put into the event detection or summarization fields in research communities [2, 3, 23, 24, 26, 28, 37]. These studies used textual data or metadata, such as the content of the tweets or the number of tweets, as the key features. While these studies aim at detecting and describing events that are going on, an essential part of the
event information is still missing, namely, the visual information. Humans are visual by nature. This is also the reason why newspapers and magazines spend much effort and budget on photographs or graphics.

Visual information is not only essential for presenting events to users, but also valuable for event summarization. First, it is not uncommon for people to use different phrases to describe the same idea. On the other hand, the same set of phrases can be used to describe differing ideas. Second, in a highly casual platform like Twitter, people tend to tweet about a topic with a photo but without using more specific vocabulary. It is not uncommon to see a tweet with photos attached while the caption is simply ”OMG”, for instance. In contrast, photos taken from the same event usually share common visual patterns. We believe that by including visual information, a more useful event summarization can be achieved.

As mobile devices become more popular and the cameras on them get more powerful, people are taking far more pictures to share their personal experience or to share new information. However, how to utilize these huge amounts of visual data remains an ongoing research topic in the data mining and computer vision communities. Without a sound summarizing technique, it is difficult for users to find the information that is relevant to them. For instance, if one wants to see what are some interesting photos about Oscar 2014, he or she can search on twitter.com via the hashtag ”#Oscars2014”. However, what the user gets will be a list of photos ordered without a clear cue about relevance to the user, as shown in Figure 1-1.

Therefore, it is essential to develop effective ways to summarize a photo collection. The task is fundamentally challenging, however, because visual data is by nature harder to summarize than textual data. Hence, in the computer vision community, many previous works on how to summarize a collection of photos [9, 30] have been proposed. However, most of them are using photographers’ community site such as Flickr. In comparison, Twitter photo collection is far more challenging and less explored.

Given the extremely casual nature of Twitter, users can share a wide variety of pictures. Some of them are mundane daily life sharing (e.g., selfies), which usually are less valuable unless the reader is socially close to the writer. Some of them are of better value in terms of news, such as political demonstrations or festivals or conferences, which might be inter-
Figure 1-1: The image search result on Twitter via the query term "#Oscars2014" as of April, 2014. Only a few of them are directly related to the event.
esting to a wider range of audience. Given this highly noisy nature, the standard clustering techniques on visual features are unlikely to perform well.

To address this issue, we developed a novel algorithm that can deal with noisy datasets such as Twitter photos, based on spectral geometry. Specifically, in contrast to most of the previous works (e.g., [30]) that used only the global distance as the similarity metrics between images, we use diffusion distance [8] instead, which puts emphasis on the local connectivity. The method increases the accuracy of similarity between images, as we will detail in a later section.

In addition, we made a key observation that users tweet about a photo because they think the photo is worth sharing in some way. Consequently, a tweeted photo can be regarded as a "vote" toward an event. For instance, people might want to tweet photos about their selfie or lunch, but presumably within the collection we will not find too many ones similar to that particular photo. In contrast, for an interesting parade or conference, there might be a fair number of similar photos. Hence, we can use this property for picking representative photos and removing the outliers. Heat Kernel Signature (HKS) can be used to quantitatively measure the density of a point and its local neighborhood. Finally, we applied Farthest Point Sampling (FPS), which maximizes the distance between sampled points.

To the best of our knowledge, this work is so far the first to incorporate spectral geometry techniques into the context of image collection summarization. This is also the first work towards an effective visual information summarization on a popular micro-blogging platform. With this novel framework, we can effectively summarize a potentially large photo collection, even for a noisy data set such as Twitter photos. This technique can be generalized to other micro-blogging platforms or photo-sharing websites. We argue that this work provides a key feature that is missing in the current summarization techniques and thus makes one critical step toward a crowd-sourced news platform.

The paper is organized as follows. In the rest of this section, we will briefly review the related works in the literature. In Section 2, we will formalize our problem and explain our metrics for a good summarization. In Section 3, we will review the mathematical background for spectral geometry, as our work is built on top of these concepts. Section 4 introduces the details of the image collection summarization algorithm. Section 5 presents
results of our algorithm and its evaluation. Finally, we conclude the paper with discussion and future directions.

Our work overlaps with two areas. In the one hand, the web mining community dealing with Twitter event detection tries to extract useful information from a huge number of tweets. In the other hand, the computer vision community dealing with image collection summarization aims at effectively convey the essense of a scene. In the rest of this section, we discuss these areas respectively and their relation to our work.

1.1 Brief review of image collection summarization techniques

As the applications are compelling, there are quite a few previous works in the field of image collection summarization. Simon et al. [30] pioneered the topic of computationally summarizing a large online photo collection. Their approach is to build a term-document matrix in a visual bag-of-words framework via SIFT descriptor. Then, a quality function is defined to maximize the similarity between each photo and its closest ”canonical view” as Simon et al. termed it. There are two penalty terms to discourage the system for (a) including too many canonical views and (b) including canonical views that are too similar to each other. The quality function is then optimized by using a greedy algorithm. In each iteration, the algorithm includes the one that produces the maximum marginal increase.

There are several differences between our method and this method. First, by considering the global distance, the quality function is maximizing something irrelevant while points are too far away. While this might be acceptable when the photo collection has intrinsic cluster, we argue that our method is suitable for more general cases. Second, by picking the canonical views that maximize the similarity between it self and the represented individual view, it is possible to be biased by outliers. In contrast, our method considers the density of data points, thus is more robust to noise and outliers.

Kennedy and Naaman et al. [22] extends the work of Simon et al. with a similar goal, to pick representative photos for a single location. The work claims its novelty by (1) the
ability to discover potential tags and locations for landmarks and (2) a novel clustering-based algorithm to pick representative photos given the discovered tags and locations. For the first part, TF-IDF was used to discover location-dependent but time-invariant tags. For the second part, clusters was first formed according to visual features applied with K-Means algorithm. Then, a set of heuristics was used to rank the the clusters and images. Finally, top images within top clusters will be included as representative images.

Crandall et al. [9] propose a system that can do the following task on a large online photo collection. First, the geographical location of photos was used to identify ”interesting” area that people are more likely to take photo with. This is done by applying mean shift algorithm on the latitude and longitude of photos. Second, for highly-photographed, the system is able to pick representative photos by first doing spectral clustering and then use the photos with highest similarity as representative photo, where similarity is deinfed by number of SIFT descriptor matched. The second task overlap with our topic to some extent but there are also fundamental differences. The paper used spectral clustering technique but did not take full advantage of the property. For instance, the dense part of the feature space is defined in a straight forward way which can be biased by highly similar outliers.

There are also several works aim at providing a computational framework for predicting aesthetics or ”interestingness” in a photo, which can in turn be used in ranking photo and summarize a collection. Datta et al. [11] use a machine learning approach to predict the aesthetics of photographs. Features such as colorfulness, low DoF indicator was extracted.

Dhar et al. [12] built a model to predict interestingness. The model used high-level describable features, such as presence of salient object, rule of thrids, low depth of field, in combination with low-level features. The authors argues that these high-level features add useful information to the prediction of aesthetics and interestingness. The high-level features are trained by manually labelled training set and then use relevent visual cues to train the classifier.

Chen and Abihishek [5] also detect event on visual data, namely Flickr. However, they used only metadata such as tags, because of the difficulty applying data analysis techniques on visual data.
Several works tackle the problem of how to organize photos with geographical information. Jaffe et al. [20] propose a system for generating summaries and visualization for a large collection of geo-referenced photographs. The system conduct a modified version Hungarian clustering on locations to produce a hierarchy of clusters. Then clusters are ranked according to factors such as time, location, quality and relevance. However, the essential factor, relevance is externally given and no visual similarity measures are used.

Epshtein et al. [14] built a system for hierarchical photo organization by reconstructing viewing frustum and choose the representative photo as the one that maximize viewed relevance.

While these works present convincing results, none of them deal with Twitter photo collection, which we believe a special mechanism is needed for its noisy and casual nature.

1.2 Brief review of Twitter event summarization techniques

Sakaki et al. [28] built an emergency event detection system by regarding individual Twitter user as ”social censors”. A given event topic is used to train a classifier so that each tweet will in turn provide a binary output. Then, techniques such as Kalman filtering or particle filtering can be applied to the output to extract the location and time of the event.

Ritter et al. [26] built TWICAL – a system that perform automatic event extraction and categorization. To determine whether an event is significant or not, TWICAL looks at the co-occurrence of tweets for a specific phrase. To categorize the event type, the system adopt an unsupervised approach based on latent variable models.

Mathioudakis and Koudas [24] demonstrate TwitterMonitor, which use bursty keywords to detect the trend on Twitter. Where the bursty keyword is defined by unnormal high rate of occurrence of a specific keyword.

Becker et al. [2] use incremental clustering technique to do online real event detection on Twitter. The features they used mainly consists of four categories: temporal features, social features, topical features and Twitter-centric features. Becker et al. has also built another system [3] that selects representative tweets for events.

Lee and Sumiya [23] develop a system that will automatically detect abnormal pattern,
namely events in their definition, mainly via number of tweets and number of crowds in a given region. The regions of interest (RoIs) are automatically constructed via clustering techniques on latitude and longitude.

Fruin et al. [16] also tried to visualize news photos. However, instead of getting the source in a crowd-sourced way, they scrape the photos from manually-picked news media sources. While the system is useful, it is suspectable that whether the system scale well since it is difficult for one to define a list of all representative news media.

While aforementioned works succeed at detecting events to a certain level using textual data or metadata, to the best of our knowledge, our work is the first one to incorporate visual information in the framework.
Chapter 2

Problem statement

The problem we are trying to solve can be formalized in the following way. Given a potentially large photo collection $X$, we would like to find a small subset $X_s \subset X$ such that $X_s$ summarizes $X$ well. While the definition of a good summarization might not be unique, in this project we are trying to find summarizations that are both representative and diversified.

By representativeness we mean that it should be possible to find many photos that are similar to $X_s$. The intuition behind this principle is that each photo can be regarded as a vote on a scene. People take a photo and share it on the social media because they think something has the value to be shared. Hence, while forming summarization, it is desirable to have photos that have the highest number of ”votes”.

By diversity we mean that the photos in $X_s$ should be dissimilar from each other so that we can minimize the redundancy. The major motivation for doing summarization is to use few entries to represent the whole set. Repeating the same information is thus undesirable.

At first look, these two criteria might sounds somewhat contradictory to each other. However, we frame the problem so that they are in fact complementary to each other. We achieve representativeness by using only the denser part in the feature space, and then within the denser part, we maximize the diversity of the points used to do the sampling.

In this thesis, we will mainly focus on the photo summarization problem although we also note that problems such as layout or the integration of the text part also have practical value. We will briefly discuss such issues in Section 5.
Chapter 3

Spectral Geometry

As the paper is closely related to the concept of spectral geometry, we will briefly guide the reader through the methametical background in the context of the problem we are trying to solve in this section. Specifically, we will discuss Diffusion Maps/Distance, Heat Kernel Signature (HKS) and Farthest Point Sampling (FPS).

3.1 Diffusion maps

Given a dataset $X$ with $n$ data points, the structure can be revealed by the similarities of the data point to each other within the set. For instance, K-Means is a clustering method that iteratively partitioning the data points into subsets, according to the distances to the centroids. However, methods like this does not work well for more complex dataset. As an example, for dataset shown in the Figure 3-1a, clusters are assigned according to global distance, as opposed to the distance along the manifold.

To address more complex dataset, manifold learning methods are proposed. Diffusion Maps is one of the technique that model the data as a Markov process and the similarity between two points can be defined as probability that one point travel to the other in a given time frame. The main advantage of doing so is to focus on the local connectivity instead of the global distance. The intuition is that, distances are usually only meaningful within a local neighborhood. For instance, two images that are very similar might be photos of the same object taken from a slightly different setting. In contrast, while comparing a
image with two very different images, there is no general rule for determining which one is "closer". Indeed, by focusing at connectivity instead of the global distance, we are able to achieve better clustering result such as Figure 3-1b.

Diffusion Maps transfer the original data points into an embedding where the Euclidean distance equal the diffusion distance, a measure that how well the points are connected to each other, as elaborate later. It mainly compose of the following steps and we will discuss them in order in the following sections.
1) Construct the affinity matrix
2) Model the Markov process
3) Non-linearly maps the data to the new embedding

### 3.1.1 Similarity graph

Given dataset $X$ with $n$ points, a standard way to model the data in the spectral geometry literature is the similarity graph $G = (V, E)$. In the graph, $V$ is the set of vertices, where each vertex represents a data point and the edges $E$ represent how well the vertices are connected to each other. In order to measure the similarity between any two given pair of points, a similarity measure kernel $k(x_i, x_j)$ should be defined. A kernel $k(x_i, x_j)$ should meet the following properties:

(a) Clustered by K-Means clustering  
(b) Clustered by spectral clustering

Figure 3-1: The difference between linear and non-linear clustering. Demonstrated by a swiss roll dataset. [Photo credit: scikits-learn.org]
1) Symmetry: \( k(x_i, x_j) = k(x_j, x_i) \)

2) Non-negativity: \( k(x_i, x_j) \geq 0 \)

3) Locality: \( k(x_i, x_j) \approx 0 \) for \( ||x_i - x_j|| > \epsilon \), where \( \epsilon \) is a given distance threshold.

Notice that the locality constraint implies that any pair of points whose distance to each other that is more than \( \epsilon \) will be considered negligibly weakly connected, as \( k(x_i, x_j) \) goes to zero quickly outside the \( \epsilon \) neighborhood. This relates to the notion of local connectivity: only distances within a certain neighborhood are considered meaningful, while the distances outside the neighborhood give us almost no information. This property also differentiate Diffusion Maps and other related methods to methods that consider the global distance, such as Principal Component Analysis (PCA).

The choice of kernel function is application-dependent, as the similarity should be measured in the context of specific applications. One common choice, and also what we used in this paper is the Gaussian kernel:

\[
k(x_i, x_j) = \exp\left(-\frac{||x_i - x_j||^2}{2\sigma^2}\right)
\]

The choice of scale parameter \( \sigma \) is not trivial and will influence the overall connectivity of the graph and thus the final result. For a smaller \( \sigma \), the locality constraint is intensified and can result in poorly-connected graph. In the other hand, a greater \( \sigma \) tends to make the graph better connected. However, noise outside the local neighborhood can get in and influence the final result. Selecting an appropriate \( \sigma \) can be an analytical process, such as Hein and Audibert [18], but to determine in an emperical way is also quite common.

### 3.1.2 Markov process

Then the weight matrix \( W \) can be constructed as the adjacency matrix form of the similarity graph:

\[
W_{ij} = k(x_i, x_j)
\]

Each entry \( W_{ij} \) represents how well a point \( x_i \) is connect to another point \( x_j \).

If we normalize the \( k(x_i, x_j) \) by the total outgoing degree \( d(x_i) = \sum_j W_{ij} \), then
\[ p(x_i, x_j) = k(x_i, x_j)/d(x_i) \] can be seen as the transition probability in a Markov process from node \( x_i \) to node \( x_j \). A degree matrix \( D \) can be formed with diagonal entries as \( d(x_i) \):

\[
D = \begin{pmatrix}
d(x_1) \\
d(x_2) \\
\vdots \\
d(x_n)
\end{pmatrix}
\]

Then we can form the transition probability matrix by normalizing \( W \) with \( D \):

\[
P = D^{-1}W
\]

With \( P \), advancing the Markov process in \( t \) steps corresponds to raising the power of \( P \) to \( P^t \). We denote each entry in \( P^t \) to be \( p_t(x_i, x_j) \)

As we are using density as an important criteria of representative images or outliers, one issue with equation 3.2 is that it does not take the sampling density into account. It is well studied in [8] that a family of anisotropic diffusion process can be obtained by specifying how much the influence of the density is desirable. The family of the diffusion process can be parameterized by a single variable \( \alpha \). Specifically, we change laplacian calculation to:

\[
L = D^{-\alpha}WD^{-\alpha}
\]

While \( \alpha = 1 \), the diffusion process reduce to an approximation of the Laplace-Beltrami operator. In this case, the distribution of the points is not considered. While \( \alpha = 1/2 \), the Markov chain is an approximation of the diffusion of a Fokker-Planck equation, and the influence of the density is considered. The latter case would be ideal for our usage, result in:

\[
L = D^{-1/2}WD^{-1/2}
\]

For more detailed explantion, we refer the readers to Coifman and Lafon [8].
3.1.3 Sparsified weight matrix

By constructing $W$ according to equation 3.1, a fully-connected matrix will be generated. However, as most of the points are out of the $\epsilon$-neighborhood, the matrix will contain mainly values nearly zero. In terms of the similarity graph, this means most of the edges are very weakly connected. Hence, we can sparsify the matrix by either:
1) Set a threshold and set everything under that to be zero
2) Keep only the $k$ nearest neighbor for each point

The sparsification serves for two purposes. First, it can speed up the computation. Second, the points that are negligibly weakly connected should have no influence. By ruling out these edges, it further increase the accuracy of the algorithm.

However, the sparsification result in an asymmetric $W$ so that equation 3.2 generates an asymmetric $L$ as well. Fortunately, by using equation 3.3, $L$ is symmetrized.

3.1.4 Diffusion maps and diffusion distance

Applying Singular Value Decomposition (SVD) to $P$ will generate the singular values $\lambda_i$, left and right singular vectors $\phi_i, \psi_i$. Then, singular values and right singular vectors can be used to construct a mapping termed diffusion maps [8]:

$$\Psi_t(x_i) = [\lambda_1^t \psi_1^t(i), ..., \lambda_l^t \psi_l^t(i)]^T$$

The new embedding $Y$ has a desirable property that the Euclidean distances between points are the diffusion distance, while all eigenvectors are used:

$$D_t(x_i, x_j) = \|\Psi_t(x_i) - \Psi_t(x_j)\|^2$$

Diffusion Maps is a very useful technique in doing dimensionality reduction. If we visualize the image in 2D using the first two coordinates from the new embedding, we can actually see that similar images are close to each other while outliers tend to be in a sparse part of the graph, as shown in Figure 3-2. This property is useful while removing the outliers, as detailed in the later section.
Figure 3-2: TED image scatter plot according to the first two dimension in $Y$
3.2 Heat Kernel Signature and Auto Diffusion Function

Heat Kernel Signature (HKS) [32] is a way to compactly encode the geometric information of a shape. Gebal et al. [17] is another closely related technique that shares similar properties, referred to as Auto Diffusion Function (ADF). Both of the works are time-dependent as it encode the geometric information in a multi-scale way. In this paper, we follow their work but consider one specific time which is empirically chosen. In the following of this section, we first introduce the definition of HKS and ADF first, and then illustrate the application of these techniques on our study.

HKS can be represented by the following equation:

\[ HKS(x; t) = \sum_{i=0}^{\infty} e^{-\lambda_i t} \phi_i(x)^2 \]  

(3.5)

Where \( \lambda_i \) and \( \phi_i \) are the \( i \)th eigenvalue and eigenfunction of the Laplace-Beltrami operator.

Similarly, ADF can be expressed by the following equation:

\[ ADF_t(x) = \sum_{i=0}^{\infty} e^{-t \lambda_i / \lambda_1} \phi_i(x)^2 \]  

(3.6)

As we can see from the definitions, the two methods are very similar. In the following of the paper, we will focus the discussion on HKS while the readers should note that the same discussion also applies to ADF.

The property we are seeking from HKS is that, for a fixed \( t \), \( HKS(x, t) \) captures the density around the local neighborhood of point \( x \). This property is highly applicable for filtering out outliers and to focus on the popular views only, as detailed in a later section. With a slightly abuse of notation, we will denote \( HKS(x, t) \) with fixed \( t \) as HKS value in the following of the paper.

An example of HKS value is shown in Figure 3-3. As we can see from the figure, denser part has lower value while sparse part has higher values.
Figure 3-3: An example of HKS value. Blue colors indicate lower values while red colors indicate higher values.
3.3 Farthest Point Sampling

Farthest Point Sampling (FPS), originally introduced by Eldar et al. [13], is a technique to sample a set of data points $X$ so that the sampled points are dissimilar to each other in terms of a given distance metric $d$. In this way, given any point $x_i$ in the data set, the distance between the point and the closest sampled point $x_n$ is minimized. Thus, the distortion of not presenting in the sampled set can be minimized.

Put it more formally, assuming that pairwise distance matrix $D$ is given, where each entry $D_{ij}$ represents the distance between point $x_i$ and point $x_j$. i.e., $D_{ij} = d(x_i, x_j)$. The FPS is a progressive process and the first point sampled can either be given by user or randomly generated. For each succeeding step, the point sampled will be determined by the distance of currently sampled points. The next sampled point $x_n$ will be the point that maximize the distance with the aggregated currently sampled points $S$:

$$x_n = \arg \max_j d_t(j)$$

Where $d_t$ is the minimum distance we have seen so far:

$$d_t(j) = \min(D_{ij}), i \in S$$

Then we add $x_n$ into $S$ and continue the next iteration until the number of point sampled reached the desired number.
Chapter 4

Image Collection Summarization

Algorithm

In this section, we will go through our algorithm for summarizing photo collection in detail. In a nutshell, the key steps are: generating affinity matrix according to selected feature(s), computing diffusion distance, remove outliers according to Heat Kernel Signature and then apply Farthest Point sampling. The overall workflow can be summarized as Figure 4-1. Please note that two features was used in the figure, but it can be extend to any number of $n$ features where $n$ is a small positive integers. We will discuss each component in detail in the following sections.

4.1 Data cleaning

As mentioned earlier, Twitter data by its nature is very noisy. One of the problem is that it contains a fair number of duplication. This can be a problem as we are looking at how many similar images in the dataset to determine the ”voting” for a specific scene. For example, Figure 4-2 shows an example of duplicate images finally got selected as part of the summarization.

To eliminate the duplicated image is conceptually straight-forward. However, we need to consider the computational complexity. A brute-force solution is to compare every image to all the other images in the dataset. This result in a $O(n^2 \times w \times h)$ complexity algorithm
Figure 4-1: System overview

Figure 4-2: Duplicated Images in the input photo collection collected from Twitter API
where \( n, w, h \) is the number of data points, width and height of the image, respectively. This can be impractical while \( n \) is non-trivial.

We used hash mechanism to eliminate this problem. For each image, we used MD5 algorithm to hash it into a table. While traversing the data set, we checked the hashed key and see whether it is already in the table or not. If so, duplication is found and we remove the image from the set. This result in an algorithm of complexity \( O(n) \).

### 4.2 Feature generation and affinity matrix

While comparing the similarity of a given pair of images, apparently pixel-by-pixel comparison is too naive to use, it is too sensitive to the setting and alignment of the images. The right features to describe an image has long been an active direction in the computer vision research. In this project, we referenced the SUN Database by Xiao et al. [38] for a list of state-of-the-art visual features. The list includes densely sampled SIFT, sparse SIFT (bag-of-words approach [31]), GIST [35], HOG [10], Tiny Images [34] and so on.

In addition to that, we have also implemented Gaussian blurred vectorized image and color histogram by our own. For the blurred vectorized image, we subsample the image to a fixed dimension and apply Gaussian blur to each of them. Then we vectorize the image as the feature vector.

Given a dataset \( X \) with \( n \) data points, we first generate a feature matrix \( F \) where each row represents a data point encoded by the chosen feature. We normalize each column into a unit vector so that each dimension has equal contribution to the distance.

For the color histogram, each of the R, G, B channel, a histogram with 32 bins are constructed. Each histogram is then normalized by the number of pixels. Finally, the three histogram is concatenated to form the feature vector for a data point.

We then generate the distance matrix \( R \) using the pairwise distance of \( F \). Where \( R_{ij} \) shows the affinity of two points \( x_i \) and \( x_j \) and is defined by the distance of row \( i \) and row \( j \) in \( F \). We use \( L^2 \) distance to calculate the distance except the case of histogram. In the case of histogram, Earth Mover’s Distance [27] was used to calculate the distance.

As mentioned in the previous section, to construct a weight matrix that only local dis-
tance matters, Gaussian kernel is used. The scale parameter $\sigma$ is determined by taking the median of all the closest neighbor in $R$, and scaled by a hand-picked parameter $c$. This variable is usually around 10 in our system.

Then, the weighting matrix is constructed by applying Gaussian function to $R_{ij}$:

$$W_{ij} = \exp\left(-\frac{R_{ij}^2}{2\sigma^2}\right)$$

### 4.3 Diffusion maps and diffusion distance

With the weighting matrix, we then calculate the diffusion distance matrix $D$, mostly follow the steps mentioned in the previous section. First of all, we construct the diagonal matrix where the diagonal entries are as the following:

$$D_{ij} = \sum_j W_{ij}$$

Then the normalized graph Laplacians is constructed. As mentioned in the previous section, the matrix is normalized in a way we mentioned in section 3.1.2:

$$L = D^{-1/2}WD^{-1/2}$$

Singular Value Decomposition (SVD) is then applied to the graph Laplacians $L$ to generate eigenvalues $\lambda_i$ and left and right eigenvectors $\phi_i, \psi_i$.

With eigenvalues and eigenvectors of $L$, we then are able to construct the diffusion maps:

$$\Psi_t(x_i) = [\lambda^T_i \phi^T_i(i), ..., \lambda^T_i \psi^T_i(i)]^T$$

By definition, the diffusion distance can then be calculated as the Euclidean distance between data points.

$$D_t(x_i, x_j) = \|\Psi_t(x_i) - \Psi_t(x_j)\|^2$$
4.4 Outliers removal

As mentioned in the previous section, given a set of data points, Farthest Point Sampling (FPS) will try to maximize the diversity by maximizing the distance between sampled points. This works well if the dataset does not contain outliers. However, with outliers, the process will tend to select the outliers since it typically has larger distance to other non-outliers. In real world data, outliers are very common. For instance, Figure 4-3 shows some example of outliers in our data. Hence, a method for removing outliers before FPS applied is necessary.

As discussed in the previous section, Heat Kernel Signature [32] has a nice property that the value will be lower in the denser part and vice versa. We calculated HKS value for each of the data point according to equation 3.5. The data points are then sorted according to the value. FPS then to be applied on the data points that have lowest HKS value. In practice, for highly noisy datasets such as Twitter images, we apply FPS to only the lowest 20% data points.
To further remove the outliers, one can repeat the process for different features and apply FPS to the intersection of densest part of each feature. The principle is similar to RANSAC [15] in the sense that by taking union of the outliers, we increase the robustness of the process.

### 4.5 Farthest point sampling

After the outliers removal process as described in section 4.4, the subset of the dataset $X_s$ will have none or only a few outliers. On that FPS can be applied. Specifically, from the new embedding $Y$ generated by Diffusion Maps, we keep only a subset of rows $Y_s$ where the points are in $X_s$. From there we calculated Diffusion Distance $D_s$ between each point in $Y_s$. Then FPS can be applied on $D_s$ as described in 3.3.

### 4.6 Feature selection and parameter tuning

Which features to use depends on the nature of the dataset. For a clean and continuous dataset, blurred vectorized image performs the best. In this case, for each image, the difference between itself and its local neighbors is only a small offset. The blurred image can thus connect the images by averaging the neighboring pixels.

In the other hand, for noisy dataset we have found that features considering the overall statistics, such as normalized color histogram, sparse SIFT and GIST, perform better.

There are a few parameters in our systems. The important ones include the scale parameter $\sigma$ of Gaussian kernel, the time $t$ for Diffusion Maps and the filtering rate $k$ for HKS.

The choice of $\sigma$ will affect the locality of the graph. For choosing smaller $\sigma$, the system becomes stricter about whether two images are connected or not. However, this might result in a poorly connected graph. We choose the parameter emperically around 10 times of the median of closest neighbors. We increase the $\sigma$ as the dataset gets noisier.

From our experience, $t = 1$ usually give us the best result. We have also tried commute time distance [39] but it did not improve the result.
The choice of $k$ also depends on the noisiness of the data. We can filter out nothing if we are certain about the dataset contains no outliers. For a noisier dataset such as Twitter images, we applied FPS on the densest part (around 20%) only.

### 4.7 Implementation details

We implemented most of the system, including diffusion maps, Heat Kernel Signature and Farthest Point Sampling, in Python with the usage of libraries such as NumPy, SciPy and Scikit-Learn. For part of the feature generation, the Matlab code from the SUN Database [38] was used. OpenCV was used for histogram calculation and Earth Mover’s Distance.

We run the code on a late 2012 Macbook Pro with 2.9 GHz Intel Core i7 CPU and 8 GB 1600 MHz DDR3 RAM. The calculation time is dominated by pairwise distance calculation. For around 1K data points and reasonable dimensionality, the elapsed time ranges from a few second to tens of seconds. For instance, the Oscar dataset #2 with 2250 images using color histogram (96 columns) as features took 23 seconds to compute the diffusion distance and HKS.
Chapter 5

Results and Evaluation

In this section, we will present our results from applying the proposed image collection summarization technique.

We first demonstrate the effectiveness of diffusion distance and farthest point sampling by applying the technique to photos taken from many points surrounding an object in a full circle. Part of the photos are shown in Figure 5-1. Since we circle the object, the first few frames have a similar view to the last few frames. The fact can be captured by the diffusion distance. If we plot the diffusion distance to the first data point, as shown in Figure 5-2, the distances are in a reversed U shape. As expected, the first few frames are similar to the first frame. In addition to that, the last few frames can also be ”connected” through the last frame.

One desirable property of diffusion maps is that the technique can recover the intrinsic low dimensional variables. Vectorized blurred images as feature vectors were used in this example, hence the observed dimension is $w \times h$ where $w$ and $h$ are the width and height of the blurred image, respectively. However, the intrinsic variable is simply the degree of the angle from which the camera surrounds the car. Hence, the use of diffusion maps is able to recover the intrinsic parameters, as shown in Figure 5-3, where we reduce the dimensions to only 3. We noted that this example is more non-trivial than the toy example shown in Talmon et al. [33] since our bare feet recording produced some shakiness.

After diffusion distance matrix is constructed, FPS was applied with $m = 10$ and the first image as the seed. The result is shown in Figure 5-4. As we can see from the figure,
Figure 5-1: Part of the photos from the Media Lab City Car dataset
Figure 5-2: The diffusion distance to the first data point in the Media Lab City Car dataset

Figure 5-3: Data points in the Media Lab City Car dataset in diffusion maps embedding projected to 3D. The color indicates the frame numbers.
Figure 5-4: Media Lab City Car data set summarized in ten images

the angle of taking the picture is roughly evenly sampled. By our problem statement, this completes a summarization with diversity because the redundancy is minimized while presenting an event/object.

We also collected Twitter images and verified our technique on real Twitter events. Twitter Streaming API was used for collecting the data. Tweets from the United States that have Geo information and photos attached were collected. As the time of writing, five months of the tweets and images were collected, result in a dataset around two Terabyte with millions of images. The collected tweets was stored in a MongoDB-powered NoSQL database. The potential candidate events are systematically investigated by sorting the hashtags by its frequency of occurrence. Then, events are discovered by high-frequency hashtags that are time-dependent. Here, we present two events: TED 2014 and Oscars 2014.

We got the TED dataset by querying the database with #TED2014 hashtags and got 331 images. Part of the images are shown in Figure 5-5. As we can see from the figure, many images actually have little to do with the event. If we simply apply the techniques used in the Media Lab City Car dataset, the result will be unsatisfiable because of the outliers.

Hence, an outlier removal mechanism needs to be developed in order to have a summarization that is more representative. To remove the outliers, we apply the heuristics discussed in the introduction section, that the non-outliers should appear in the denser part of
Figure 5-5: Part of the photos from TED event dataset
(a) Images from TED dataset that have the lowest HKS value

(b) Images from TED dataset that have the highest HKS value

Figure 5-6: Images from TED dataset with lowest and highest HKS value
the feature space because there are more similar data points nearby. In contrast, outliers usually appear in the sparse part of the feature space since it distinguishes itself from other points and thus there are little data point nearby. As described in the previous section, HKS is an highly applicable technique for the task. We applied HKS with a fixed small $t$, and then sort the images according to the value. In our setting, photos with lowest values corresponds to the data points in the densest part of the feature space, and vice versa. While applied the technique on feature space generated by the normalized color histogram feature, the result is shown in Figure 5-6, where images with lowest and highest HKS value are listed. As we can see from the figure, images with lowest HKS value are indeed correlates better with the TED event, while images with highest HKS value tend to be outliers.

To further eliminate the outliers, the process can be repeated for multiple features. Figure 5-7a shows the images with lowest 5% to 10% HKS value applies to affinity matrix generated by normalized color histogram. As we can see from the figure, there are still a few outliers. However, if we intersect the lowest 20% of HKS value applied on both normalized color histogram and sparse SIFT feature, the result is shown in Figure 5-7b. While there are still one or two outliers, if we compare this to the version that filtered only by a single feature, it is much more applicable to apply FPS on.

After filtering out the outliers, we apply FPS to diversely sample from the filtered subset, and the result is shown in Figure 5-7. Our technique focus on the most common scene (e.g., speech by Edward Snowden) but also has different settings from the same scene to attain the diversity.

Another dataset we experimented with is the Oscar dataset. By querying the database with the hashtag "#Oscars2014", and filtered out the duplicated images we got 1165 images. Part of the images are shown in Figure 5-8. As we can see from the figure, the dataset is even noisier than the TED dataset. HKS can again be used to separate the denser part of the data points from the sparse part.

Normalized color histogram and GIST descriptor [35] was used to generate the affinity matrix in this dataset. Figure 5-9a and 5-9c show the photos with lowest HKS value using normalized color histogram and GIST as the feature, respectively. The intersection of the top 15% of the two sets is shown in Figure 5-9. For comparison, photos with highest HKS
(a) Images from TED dataset with lowest 5% to 10% HKS value, affinity matrix generated by normalized color histogram

(b) Intersection of the lowest 20% HKS value applied on both normalized color and histogram and sparse SIFT feature
Figure 5-7: Summarizing TED 2014 in five photos

Figure 5-8: Part of the photos from the Oscar data set
(a) Photos from Oscar dataset with low HKS value using normalized color histogram as feature

(b) Photos from Oscar dataset with high HKS value using normalized color histogram as feature
(c) Photos from Oscar dataset with low HKS value using normalized color GIST as feature

(d) Photos from Oscar dataset with high HKS value using normalized color GIST as feature
Figure 5-9: Intersection of lowest 15% HKS value applied on both normalized color and histogram and GIST
value generated by normalized color histogram and GIST are shown in Figure 5-9b and 5-9d, respectively.

By using normalized color histogram, we have better result for filtering out the outliers. However, result from GIST is also useful in the sense that a photo has to be ”non-outliers” for both of the feature to be included in the filtered subset. Although there are still a few outliers in the intersection, it can be addressed by further tuning the parameter or features and it was not included in the final summarization. We also noted that we omitted a few images in Figure 5-9a for reducing the redundancy. There are several images similar in the form to the famous selfie by Ellen DeGeneres with some trivial modification or image compressing offset.

Finally, we applied FPS on the intersection and the resulting summarization is shown in Figure 5-10. As we can see from the figure, all images are directly relevant to the event. Our algorithm did not use clusters implicitly, however, we can see the summarization is
formed by different part of the feature space. In particular, two images coming from the famous selfie, two images coming from the red carpet photos, and one image coming from red carpet photo in the television (which is visually different from the original red carpet photos). Our system is able to capture that many photos are in fact a modification of the famous selfie and include it in the summarization. In this particular example, we can see our system achieves both representativeness and also diversity.

5.1 Evaluation

In this section, we present the evaluation by comparing our result to ones that generated by metadata. When measuring the popularity of a tweet, people usually look at the number of retweet. Hence, one way to summarize a Twitter image collections can be looking at the tweets that are retweeted the most. To some extent, this method overlap with our method in the sense that using popularity as ”voting” toward the ”interestingness” of a scene.

Figure 5-11 shows the top photos that are retweeted the most with hashtag ”#TED2014”.

Figure 5-11: Photos most retweeted with hashtag ”#TED2014”
As we can see from the figure, both our result and the most retweeted photos share the scene that Mr. Edward Snowden is speaking. This validates our result by proving that our method is able to capture the important moment. In addition, our result demonstrates a better diversity by having different speakers.

Figure 5-12 shows the top photos that are retweeted the most with hashtag "#Oscars2014". Compare to our result, the representativeness of the photos that are most retweeted seems to be low because almost all of them are not actually related to the event. This also validates one of our hypotheses: the photo most retweeted might be biased by other factors. For instance, an user with more followers tends to get more retweets (e.g., Figure 5-13a). Or tweets about a controversial topics (e.g. Figure 5-13b), ongoing social movement (e.g., Figure 5-13c) might get more retweets because of the supporters. Hence, the summarization by picking photos most retweeted can be biased by one or more of the above factors.

One limitation of this comparison is that not all the photos related to the event will be included but only the photos that has a specific tag we used. In this particular example,
(a) Number of retweets biased by number of followers

(b) Number of retweets biased by a controversial topic

(c) Number of retweets biased by an ongoing social movement

(d) The famous selfie by Ellen DeGeneres
<table>
<thead>
<tr>
<th>Event/Method</th>
<th>Proposed method</th>
<th>Most retweeted</th>
</tr>
</thead>
<tbody>
<tr>
<td>Event 1: TED 2014</td>
<td>60%</td>
<td>40%</td>
</tr>
<tr>
<td>Event 2: Oscar 2014</td>
<td>95%</td>
<td>5%</td>
</tr>
</tbody>
</table>

Table 5.1: The result of comparing proposed method with baseline on two events using Amazon Mechanical Turk

<table>
<thead>
<tr>
<th>Event/Method</th>
<th>Proposed method</th>
<th>Most retweeted</th>
</tr>
</thead>
<tbody>
<tr>
<td>Oscar 2014 (#Oscars &amp; #Oscars2014)</td>
<td>52%</td>
<td>48%</td>
</tr>
</tbody>
</table>

Table 5.2: The result of comparing proposed method with baselines on a larger data set of Oscar photos using Amazon Mechanical Turk

...the famous Oscar selfie taken by Ellen DeGeneres has the hashtag "#Oscars" instead of "#Oscars2014" (Figure 5-13d) so it was not included in the summarization by counting the retweet numbers. This result also demonstrates the advantage of our approach since the users do not always know which hashtags will be the most appropriate one. However, via looking at visual features with meaningful amount of data, the pattern can be identified and thus reasonable summarization can be generated.

We have also verified our result by Amazon Mechanical Turk (MTurk) by comparing our result with most retweeted tweets for the above mentioned two events. For each event, we upload our input photo collections and two summarizations on the MTurk. We asked the users to choose which summarization they think is better without knowing which result was generated by which method. We provided clear criteria about what is our definition of good summarization, namely representativeness and diversity. We noted that representativeness is generally weight higher than diversity because a diverse set of outliers provide very little information.

We conducted the MTurk experiment and have collected 20 responses. The result is shown in the Table 5.1. The value shows the percentage of users that prefer a specific method. In the evaluation of the result of TED data set, we notice that our result is slightly better than that of most retweeted, or at least of similar quality. Consider that our method is purely automatic, we think this illustrates the usefulness of our method. As for the result of Oscar data set, it might not be a surprise that most of the users agree on the fact that our method produces better result.
Surely, one can argue that it is more appropriate to use hashtags is more favorable for the most retweeted method. Although we see this as the advantage of our method, in the sense that users do not need to have the knowledge of the most appropriate hashtags a priori. However, we are still interested in the result of comparing our method with baseline methods in the data set that generated by the most appropriate hashtag. We generate another data set in which photos has either "#Oscars" or "#Oscars2014" as hashtag. In addition to comparing to most retweeted photos, we would also like to compare with most favorited photos. Again, we upload the input photo collection and the results of our result (as in Figure 5-13a), and most retweeted photos (Figure 5-13b) to Amazon Mechanical Turk. We ask turkers which summarization they prefer the best, similar to the above experiment, and the reulst is in Table 5.2 based on 50 responses. As we can see from the result, users still prefer our method the most in this setting. We think the reason might be the fact that there are still a few outliers in the baseline methods that are biased by the factors we mentioned before.
5.2 Discussion

From the results above, we found several interesting topics to discuss. Firstly, we would like to discuss how our work can be potentially extend to a crowd-sourcing approach. Secondly, we explain how to integrate the textual part of the event summarization. Finally, we discuss the insight of how similar photos are influencing the performance of the system.

Figure 5-14: A prototype for Rep or Not. A crowd-sourcing approach for image collection summarization problem
5.2.1 A crowd-sourcing approach

Recently, crowd-sourcing and human-in-the-loop computation have been extensively studied. For example, VisionBlocks [4] [6] provides users with a user interface for visual programming. CrowdCam [1] help users navigate crowd images. For a survey paper on this topic, we refer the readers to [36]. Among them, "hot-or-not" style crowd-sourcing produced some interesting results such as Place Pulse [29] or StreetScore [25], where users are asked which one in a set of two images looks safer or welthier, etc. Then the pattern can be analyzed to derive the possible visual features that make a place look "safer".

It is tempting to speculate whether the same approach is applicable to image collection summarization or not. We have built a Human Intelligence Task (HIT) task on Amazon Mechanical Turk. Figure 5-14 shows the interface of it. We name the task "Rep or Not" as it collects the ground truth about whether users think a specific image is representative or not. Compared to projects such as Place Pulse, the problem we are solving is more challenging in the sense that whether an image is representative or not depends on not only the image perse, but the relation of the image and the whole collection. Hence, although it is an interesting topic, how the ground truth provided by users can be incorporated is not straightforward. However, we do note that the ground truth combined with ranking algorithms can be used as another form of evaluation.

Specifically, let us assume the data was collected from the pairwise comparison for the whole image collection. We can rank the representativeness of each image by the TrueSkill algorithm [19] or methods that are similar. In the case of TrueSkill algorithm, the representativeness of each image can be modeled by a normal distribution $N(\mu, \sigma^2)$. All the representativeness are initialized as the same normal distribution with chosen parameters $\mu$ and $\sigma$. After that, we update distribution with each round of "match". Let us assume two images $x_i$ and $x_j$ had a match, in which $x_i$ was chosen as more representative than $x_j$. The update will be as follows:
\[
\begin{align*}
\mu_i &\leftarrow \mu_i + \frac{\sigma_i^2}{c} \cdot f \left( \frac{(\mu_i - \mu_j)}{c}, \frac{\epsilon}{c} \right) \\
\mu_j &\leftarrow \mu_j - \frac{\sigma_j^2}{c} \cdot f \left( \frac{(\mu_i - \mu_j)}{c}, \frac{\epsilon}{c} \right) \\
\sigma_i^2 &\leftarrow \sigma_i^2 \cdot \left[ 1 - \frac{\sigma_i^2}{c} \cdot g \left( \frac{(\mu_i - \mu_j)}{c}, \frac{\epsilon}{c} \right) \right] \\
\sigma_j^2 &\leftarrow \sigma_j^2 \cdot \left[ 1 - \frac{\sigma_j^2}{c} \cdot g \left( \frac{(\mu_i - \mu_j)}{c}, \frac{\epsilon}{c} \right) \right]
\end{align*}
\]

(5.1)

Where \( \epsilon \) is an empirically chosen parameter, which represents the probability that two players will tie and \( \beta \) indicates a per-game variance. \( \mathcal{N}(\theta), \Phi(\theta) \) represent Normal probability density function and Normal cumulative density function, respectively.

Assuming enough comparisons are conducted, the TrueSkill algorithm will converged to a stable score, by which we can assign representativeness score \( r_i \) to images. For instance, one simple choice is to assign \( r_i \) linearly according to the rank:

\[
r_i = n - \text{rank}(i)
\]

Where \( n \) is the number of images.

With this, two tasks can be done. First, we can use this for evaluating whether our system proposed in Section 4. Second, we can train a regression system that predicts the representativeness of images in the input collection, which can be another approach for solving the summarization problem.

For evaluation, we can evaluate a summarization, generated by our proposed method or not, by calculating the following quality function, which measures both representativeness and diversity:
\[ Q(S) = \sum_{i=0}^{n} r_i - W \cdot Sim(S) \]

\[ Sim(S) = \sum_{i,j} S(i,j), \forall i \in S \land i \neq j \]

The basic idea here is to maximize the representativeness but to minimize the similarity. \( W \) is a parameter to control the weighting between maximizing representativeness and minimizing similarity. \( S(i, j) \) is a measurement of similarity between point \( x_i \) and \( x_j \) and is thus application-dependent.

Following this framework, we can also derive another automatic approach that solve the image collection summarization problem. We can extract features as described in Section 4.2 and the crowd-sourced ranking data becomes the target variable that we can train a regression system to predict. A common choice for a regression system will be Support Vector Regression, which finds a regression function \( f(x) \) that approximates \( y \):

\[ f(x) = (w \cdot x) + b, \]

\[ w, x \in \mathbb{R}^N, b \in \mathbb{R} \]

The ideal regression function \( f(x) \) will minimize the error between predicted value and the target value and also consider the complexity by minimizing the following function:

\[ \frac{1}{2} ||w||^2 + \frac{C}{K} \sum_{i=1}^{K} |y_i - f(x_i)| \]

Once we predicted the representativeness using the regression system, we can again find summarization by maximizing the quality equation 5.2.

One issue with the automatic system using crowd-sourcing data is the scalability. The number of comparisons grows at least linearly with the number of images in the collection. If we consider the noise produced by Turkers, the complexity can be even steeper. However, we do see an opportunity here to integrate our system with the crowd-sourcing approach. As we have noted in section 4.4, our system is good at automatically removing most of the
outliers. However, in practice we still see a few outliers in the denser part of the feature space. These outliers are probably visually similar to non-outliers so that it is hard to separate them in a fully-automatic approach. Hence, we can derive the following algorithm. First, in the outliers removal process, we focus on the top $k\%$ images with lowest HKS value. Second, we apply the crowd-sourced comparison described in this subsection and rank them accordingly. Finally, regression model can be used or FPS can be applied to the top images. In other words, crowd-sourcing approach can be a potentially useful extension to our proposed method.

5.2.2 Textual data and metadata integration

As we mentioned in the Section 2, our goal in this study is to summarize Twitter event from a visual perspective. Hence, the textual summarization part, although practically valuable, is beyond the scope of this project. However, we do note that our work can be a valuable addition to the textual summary techniques. One straightforward way to combine the visual part and the textual part is to compute the summary separately and integrate it in a well-designed layout. Another way is to select tweets that have photo selected by our algorithm as the representative tweets. In that case, both photos and texts will be included in the summary. Moreover, our algorithm can be used in image search. Given a keyword, images can be sorted in the order of HKS value as it indicates how many similar photos are in the data set and how many people are paying attention to the scene.

One interesting future application is to summarize photos across the globe using the location data. We can summarize what people are paying attention in different cities within different time, which can be served as an crowd-sourced local news. This can be done by dividing the world into several metropolitan area. For each of them, we apply the summarization framework we described in the paper. Finally, the highlighted photo can be visualized at a world scale map. This is one of our original motivation for attacking the image collection summarization problem, and we think this can be an interesting future direction of this project.
5.2.3 Near-duplicate photos and its influence

As we mentioned in Section 4.1, we removed the duplicated data so that the system will not be biased by it. However, we do note that there are almost identical photos with slight variation (due to cropping, different hue, image compression offset, etc.) that are not filtered out by our system. As the photos are almost identical, it can be arguably regarded as another form of retweet. We referred to this kind of photos as "visual retweets", as it is different from the general form of retweets, which based on a combination of factors such as text. For example, Figure 5-15 shows the photos from Oscar data set with lowest HKS value applied on color histogram feature matrix. As we can see from the figure, many of them are the same image with slightly different variation. Hence, to a certain degree, one can argue that our system use the retweet signal. It is also because of this signal, we are able to include the famous Ellen’s selfie in a group of highly similar photos in our summarization (Figure 5-10).
It naturally leads us to several questions. First, does the fact that retweet signal was used undermines the value of our system? Second, to what degree do we have the signal in our system? Third, can we remove the signal? Our answers to these questions are as follows.

We argue that the value of our system is not undermined by the fact that retweet signal, to some degree, is in our system. Apparently, our system is not purely operating on only the retweet signal. If that is the case, we will see our result identical or very similar to the most retweeted photos. However, our system generates very different result from the most retweeted photos and proven to be preferred by users. For instance, Figure 5-13 shows fairly different result from our method and the most retweeted photos. Moreover, our system is able to include Ellen’s selfie even if different hashtag was used (#Oscars2014). The basic assumption of our algorithm is that by looking at what photos people are taking, we can find representative photos within the similar photos. Whether the users physically took the photo by themselves should not affect the fact that photos are similar and worth sharing. Therefore, the contribution of our system lies in the ability to extract pattern from similar photos, no matter they are visual retweeted or not.

We further examined how significant is the phenomenon of visual retweet exists in our data set. In the Oscar data set, there are about 20 to 30 images that are Ellen’s selfie with slight variation within 1165 images. In the TED data set, there is little or no such phenomenon exist, as we manually examined the data set. In addition, the summarization photos we found generally have low retweet numbers. For the TED data set summarization generated by our method, the retweet counts from left to right in Figure 5-7 are 18, 6, 3, 4, 1 respectively. For the Oscar data set summarization generated by our method, the retweet counts from 2nd left to rightmost in Figure 5-10 are 0, 15, 0, 0 respectively. The first one in Figure 5-10 appears to be deleted so we can not get the retweet count but the first three visual retweets in the dataset returned 0, 4, 0 as retweet counts. Hence, within the two summarization photos we generated so far, we speculate that only Ellen’s selfie is influenced by the visual retweet signal, while others are generated by the regular similar photos. We think Ellen’s selfie is special in the sense that it is a) highly worth sharing and b) only very few people can physically take the picture from that time and space. Hence,
it helps to explain why visual retweet works and is needed in this case. We also want to note that the retweet numbers suggest that the photos we found have only little overlap with the photos that are highly retweeted. This implies the signal we found is orthogonal to the general retweet signal and thus the combination of two signals can be further investigated.

Finally, one might want to ask whether it is viable to filter out the visual retweet or not. In the current system, we filter out the images that are exactly the same. Following the same logic, we can introduce a threshold $\epsilon$ so that any pair of images that has a distance smaller than $\epsilon$ will be filtered out. The tricky part is the value of $\epsilon$ might not be easy to tune. For an $\epsilon$ that is too small, the system will fail to filter out the almost identical photos. For an $\epsilon$ that is too large, it will filter out the regular similar photos and the signal our system relies on will be removed. In addition, feature selection also plays a factor and that near-duplicate image detection is not trivial. Several papers already published in the field, for instance, [7], [21] and [40].

In summary, we do not considered the value of our system is undermined because of the fact that our system partly using the visual retweet signal. Also, although further examination might be needed, we speculate the influence of the visual retweet is limited. Finally, it is possible to partly remove the visual retweet by setting a carefully chosen threshold as the minimum distance within the image collection.
Chapter 6

Conclusion

In this thesis, we presented a novel method to summarize a potentially large image collection with the motivating scenario of summarizing Twitter photos over events. The method mainly composed of three parts. First, diffusion distance is used to model the connectivity between data points. Second, outliers are filtered out according to the Heat Kernel Signature value. Third, Farthest Point Sampling is applied to sample the denser part in the feature space. By using this method, we achieved both representativeness and also diversity. Representativeness is achieved by focusing on the denser part in the feature space, where the data points have more similar photos in the dataset. Intuitively, it can be think of focusing on scenes that have more “votings” compare to others. In the other hand, diversity is achieved by applying FPS, which maximize the distance between sampled points.

We presented results of our method. In a clean and continuous dataset generated by a video surrounding an object, the system is able to sample evenly from the dataset, providing a roughly evenly spaced view for an object. We also presented datasets from real-world Twitter data. We presented both the summarization of TED and Oscar in the year of 2014. Both summarization include images that are directly relevant to the event despite of the high ratio of the outliers. Different part of the dataset are also sampled. We compared the result with the most retweeted photo and claim that our result is at least have similar quality despite of being an automatic method. In some case, our result achieves even better result because the retweet number can be biased by a number of factors.

There are several limitations of our method. First, the selection of feature and the tuning
of parameters is not trivial. Different dataset usually requires different features. Diffusion Maps is relatively sensitive to parameters. For instance, the $\sigma$ used in the Gaussian Kernel.

Second, the system is ignorant of high level semantic knowledge. For instance, red carpet photos can be detected, however, the system is indifferent to who is actually on the red carpet, as long as they are visually similar. However, this is a fundamental question in computer vision which probably beyond the scope of this paper.

We foresee several future directions. First, we want to explore the combination of visual and textual data as features. The textual data can be noisy and casual in social media environment and the metadata data can be biased by a number of factors. However, they can be potentially useful if we can find a clever way to combine both textual and visual data. Second, we want to extend the event summarization techniques to event discovery method. Finally, it would be interesting to extend the method to online clustering, so that the summarization can happen in the realtime and update itself as the new information arrives.
Bibliography


