Image Set Summarization For Album Creation
Project Report: CIS 519

Mihir Pattani MIHIRSA@SEAS.UPENN.EDU
Nidhi Angle VNIDHI@SEAS.UPENN.EDU
Sarvesh Surana SSURANA@SEAS.UPENN.EDU

Abstract

The goal of our project is to select a small a set of images which best represent a large corpus of images. The input set of images may be of an an event, movie, place or object. What we eventually wish to do is to create a summary album from a large set of images. Our solution consists of using the SIFT algorithm for feature extraction. We use clustering methods for grouping similar views (images) and output a small set of images which are representative of each cluster. This small set of images forms the summary album.

1. Introduction

With the advent of social media, one of the most common and convenient ways to store/share a memory of any event is in the form of pictures. Any event, can have hundreds or thousands of pictures from a multitude of sources which can essentially define the event. People also like to click large number of pictures of places and objects such as historic monuments and hill stations. Summarising these images into a small subset which gives a good description of that event, object or place is a tedious task if done manually.

Our goal is to shorten this laborious task and instead present the user with a set of images which present the best summary. This paper demonstrates the power of image processing(SIFT) and machine learning(K-Means, Spectral Clustering etc.) algorithms and how they can be used for the above mentioned purpose. We implement this with three steps. First we take the dataset of images and identify and extract the relevant features which will be used as a comparison metric with other images. This will help to create a view to which these features will belong to or which best represent it. Second we will use a clustering algorithm to create a feature incidence matrix. This matrix will then be passed through a clustering and selection algorithm to obtain groups of views. One image will be outputted as a representative of each view. Further in this paper, we will describe how the final images are chosen. The final output is a small set of images which summarize the large corpus of images given as input. There can be multiple sources of images hence it can be expected that every image will not portray best quality. We will attempt to provide a good summary irrespective of the drawbacks generally contained in real world data.

University of Pennsylvania, CIS 419/519 Course Project.

2. Work Done

We tried multiple techniques to achieve the goal of this project. The method of processing images in the project consists of three parts:

- Feature Extraction
- Feature Clustering
- Image Summarization

Each technique tried by us performs above three steps using different algorithms.The main two techniques tried by us have been described in detail below. Based on the accuracy of the results, we chose the second technique to finally achieve the goal of our project.

2.1. Dataset

For the purpose of testing our tool we are using mainly 2 datasets-

- "The Oxford Flowers Dataset" created by the Visual Geometry group at the University of Oxford using various websites. This dataset contains 1360 images.
- The Oxford Buildings Dataset created by the Visual Geometry group at the University of Oxford using Flickr. This dataset consists of 5062 images.

3. Technique I

In Technique I, we extracted and matched features between each image pair to form an affinity matrix using ASIFT and RANSAC algorithms. We applied Affinity Propagation Clustering on this affinity matrix to obtain clusters, centroids of which were outputted as the final output (small set of images).

Figure 1. Performance comparison between ASIFT and SIFT using image from the dataset

3.1. Feature Extraction and Matching

The dataset of images was processed using the ASIFT(Guoshen Yu, 2011) feature extraction and matching algorithm to give us an output of matching features between pairs of images in the dataset. We also tested other algorithms such as SIFT, SURF and MSER. The SIFT algorithm can also be used for this purpose (Simon Ian, 2007) but it has poor performance in feature extraction when compared to ASIFT. ASIFT though just an extension of the SIFT algorithm, can identify the deformations in an object and use them. This effectively makes use of affine transformations where the SIFT lags as it uses only 4 of the 6 parameters of an affine transform. SIFT is rotation, scale and translation invariant. ASIFT deals with the remaining two- the camera angles determining the axis orientation. The tests too determined that ASIFT provides better performance in terms of accuracy of feature matching.

The FLANN based matcher, which is optimized for fast nearest neighbour search, was used to match the keypoint descriptors. We set k=2, to apply the ratio test (Lowe, 2004). The resulting matches were filtered using RANSAC. The feature extraction and matching algorithm for given dataset of images is provided in Algorithm 1.

Algorithm 1 ASIFT - RANSAC feature extraction and matching (Simon Ian, 2007)

Input: dataset of images x_i, size n.
for every pair a, b **do**
  Choose image pair a, b from x_i
  $keypoints[a]$, $desc[a]$ = ASIFT(a)
  $keypoints[b]$, $desc[b]$ = ASIFT(b)
  $matchedKeypoints[a, b]$ = FLANNMatcher(a, b)
  filter $matchedKeypoints[a, b]$ using RANSAC
end for

Using the number of matches between a pair of images as the weight of the edge between the two, we created an affinity matrix

3.2. Image Summarization using Affinity Propagation

We used affinity propagation (Frey* & Dueck), a type of clustering based on the concept of message passing between datapoints, to obtain the summary of images. Unlike k-means, affinity propagation tries to determine the number of clusters itself, using the affinity matrix of the dataset. It finds ”exemplars”, members of the input set that are representative of clusters. These ”exemplars” are given as output.

4. Technique II

In this technique we will use SIFT to extract image descriptors for each image in the dataset and store them. We will then cluster these images to form a boolean feature incidence matrix which represent the ”closeness” between the images as a pair. This image feature vector will then be fed into the greedy clustering and selection algorithm which will filter out the best representation(subset) of these images.

4.1. Feature Extraction and Matching

Feature descriptors of images were extracted using SIFT. K-means clustering was run on the entire collection of SIFT descriptors, with the aim of segregating descriptors based on the image feature they represented. This method has been used previously by (Sivic & Zisserman, 2003) to find

centroids of these clusters, which then became the visual words representing the chosen vocabulary. We deemed each of the detected K clusters to be a separate feature. The image-feature incidence matrix was populated based on the number of descriptors of the image belonging to that particular feature. The resulting matrix was input to the Greedy Clusterng and Selection algorithm in the next step.

Algorithm 2 KMeans clustering on image descriptors to generate image-feature matrix

```
Randomly generate K points as centroids
Input: collection of descriptors x, number of clusters k.
while not converged or max iterations not reached do
  for each descriptor in x do
    Assign descriptor to nearest centroid
  end for
  Update cluster centroids
end while
for Every image do
  for every descriptor in image do
    increment feature count in image-feature matrix
  end for
end for
```

4.2. Greedy Clustering

We use the below cost function in our Greedy algorithm (Simon Ian, 2007).

$$Q(C) = \sum_{V_i \in V} (V_i \cdot C_{c(i)}) - \alpha|C| - \beta \sum_{C_i \in C} \sum_{C_{j>i} \in C} (C_i \cdot C_j)$$

The cost function implicitly ensures following properties -

- Restrict number of images in the output set Subtracting alpha times number of images in C puts a limit on the number of images selected in the final output
- Ensure uniqueness of chosen images Subtracting beta times sum of similarities between all images ensures that every image in the output set is unique

The Greedy Clustering and Selection Algorithm takes in the boolean feature incidence matrix as the input and starts with an empty set C which will finally store the output images. The algorithm iterates based on the Cost of the final set C. We calculate the cost value of the entire set by taking one image at a time and finding the image which maximizes the cost value. This image is then appended to the set C and the loop is run again. At any time of the max cost evaluates to below zero we stop the procedure and return the unique images which we obtained in the set. This algorithm effectively tries to find images which do not share the same set of features between them and are different from each other.

Algorithm 3 Greedy Clustering and Selection Algorithm

```
Input: Boolean Feature Incidence Matrix calculated from the input image dataset.
C = 0
while max iterations not reached do
  for each view V in V, compute do
    Q_V = Q(C ∪ {V}) - Q(C)
  end for
  Compute max Q_V
  if max Q_V > 0 then
    Append V to C
  else
    break out of while loop
  end if
end while
Return the unique images from C as the Output
```

5. Observations

We make the following observations w.r.t the parameters α, β *and* k.

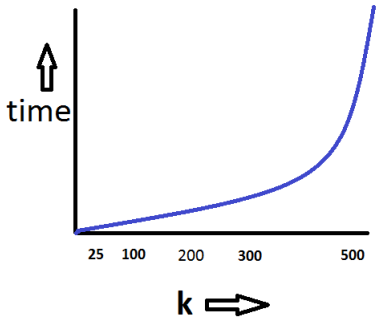

Figure 2. Analysis of K vs Time taken

a. k vs Time (Fig. 2) :
By varying the value of 'k' in our k-Means algorithm we could observe a huge fluctuation in the processing time. Based on our observations we decided to keep the value at 200 which would mean that the algorithm would execute in an acceptable time-frame.We observed the same for both Technique I and II.

b. k vs Accuracy (Fig. 3):
Accuracy in this context is defined as the - uniqueness of the output images divided by number of unique images expected. For example given an input of 100 images of 10 different types of flowers, if the output is 8 images each displaying a unique type of flower then the accuracy becomes 8 / 10 == 0.8. We observed that by keeping k small, the algorithm outputted only a small set of images

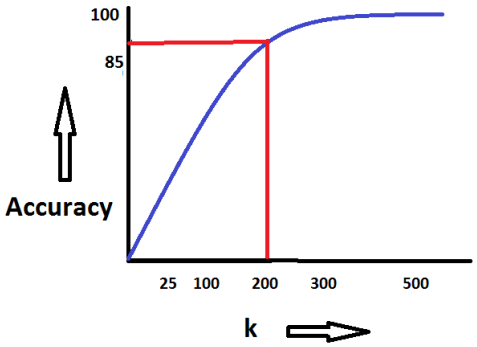

Figure 3. Analysis of K vs Accuracy

whereas by increasing the it to a very large value (say 350-500) made the accuracy asymptote. we found the ideal value of 'k' to be around 200 yielding an accuracy of 71.4% in case of Technique I and 85.7% in the case of Technique II.

c. α and β :
α and β play a role in calculating the cost value of the image set C. As we increase the value of either α or β we see that the cost value moves quickly towards such that at a point no images are selected for the output. Hence it is important that we tune these values to efficiently calculate the cost function. In our experiment we found these values to be $\alpha = 5.5$ and $\beta = 0.01$.

6. Results

We gave the images as shown in Figure 4 as input (not all input images included in figure 4). The total number of images was 220 which basically consisted of 7 different kinds of flowers. Technique 1 gave output corresponding to Figure 5 such that it outputted 8 images consisting of 5 unique types of flowers. As the number of unique flower types was 7, the accuracy of this method is 5 / 7 == 71.4%. On the other hand Technique 2, corresponding to Figure 6, gave an output of 6 images each of which represented a unique flower type. Therefore accuracy of this technique is 6 / 7 == 85.7%.

The first technique, though able to predict the approximate number of clusters correctly, did not do a very good job in clustering similar images together. The ASIFT matches were not able to effectively differentiate between similar and dissimilar images, probably since most features of all types of flowers are similar. Hence the generated affinity matrix did not prove to be a very helpful comparison metric in this case.

On the other hand, though the Greedy Clustering algorithm was able to detect most, if not all the clusters, the clustering was more accurate. The proposed 'centroids' of each cluster effectively summarized the dataset as well. Also Technique 1 was computationally more expensive as compared to Technique 2 and required almost 4 times the time to produce results. Hence we concluded that Technique 2 was the better one to create a summary of images.

Thus, we successfully implemented an efficient technique to summarize a huge corpus of images into a small set of unique ones.

Figure 4. Input - Multiple Images of 7 Unique types of flowers

Figure 5. Technique 1 Output - 8 images which represent only 5 unique flower types

Figure 6. Technique 2 Output - 6 images which represent 6 unique flower types

Acknowledgments

None

References

Frey*, Brendan J. and Dueck, Delbert. Clustering by passing messages between data points. *Science 315, 972976*.

Guoshen Yu, Jean-Michel Morel. Asift: An algorithm for fully affine invariant comparison. 2011.

Lowe, D. Distinctive image features from scale-invariant keypoints. *Int. J. of Computer Vision*, 2004.

Simon Ian, Noah Snavely, Steven M. Seitz. Scene summarization for online image collections. 2007.

Sivic, J. and Zisserman, A. Video google: A text retrieval approach to object matching in videos. 2003.

IMAGE SET SUMMARIZATION FOR ALBUM CREATION

PROBLEM

- To select a small a set of images which best represent a large corpus of images
- This could be used to create an album / collage representing an event, place, person(s) or any object in general

PROCEDURE

- Feature Extraction - Features of each image extracted using SIFT
- Feature Clustering - K-Means clustering done on the image descriptors to form a Boolean feature incidence matrix
- Image Summarization – A small set of images generated by applying a greedy clustering and selection algorithm on the feature incidence matrix. This algorithm automatically chooses number of images returned such that
  - Each returned image is unique
  - Each returned image is a representative of a group of images from the image corpus

Results

- In the example given above 220 images (not all shown above) of 7 types of flowers was passed as input.
- An output of 6 unique images (out of 7 desired) was generated.
- Accuracy = 85.7%