International Journal of Science and Research (IJSR)
ISSN (Online): 2319-7064
Index Copernicus Value (2013): 6.14 | Impact Factor (2013): 4.438

GUI Based Large Scale Image Search with SIFT Features

Jerrin Varghese[1]

[1]Computer Science and Engineering, ICET, Muvattupuzha, India

Abstract: ***Any image consists of descriptive features that uniquely describes the characteristics of the image. The scale invariant feature transform descriptors(SIFT descriptors) are one such descriptors that will not change on resizing of image. The patches generated from scale invariant feature transform descriptors are all converted to unit length. The main concern in searching of image is matching. The visual vocabulary consists of collection of features and this can be clustered using the k means algorithm. The two main k means are approximate k means and hierarchical k means. The clustering helps in indexing of image features. More over the words represented as code word which is in binary form helps in minimizing the memory cost to a great extend.***

Keywords: Image Search, GUI, SIFT, K means, Indexing

1. Introduction

In computer vision, the bag of words model can be applied to image classification by treating image features as words. In document classification, a bag of words is a sparse vector of occurrence counts of words, which is a sparse histogram over the vocabulary. In computer vision, a bag of visual words is a vector of occurrence counts of a vocabulary of local image features.

To represent an image using bag of words model, an image can be treated as a document. Words in image usually includes following three steps : feature detection, feature description and code book generation. After feature detection, each image is abstracted by several local patches. Feature representation methods deal with how to represent the patches as numerical vectors. These vectors are called feature descriptors. A good descriptor should have the ability to handle intensity, rotation, scale and affine variations to some extent. One of the most famous descriptors is scale invariant feature transform. Scale invariant feature transform converts each patch to 128 dimensional vectors. After this step, each image is a collection of vectors of the same dimension. (128 for scale invariant feature transform) where the order of different vector is of no importance. The next step for bag of word model is to convert vector represented patches to code words, analogy to words in text document, which also produces a code book which is a analogy to word dictionary. A code word can be considered as a representative of several similar patches. One simple method is performing k-means clustering over all the vectors. Code words are then defined as centers of learned clusters. The number of cluster is the code book size. Thus each patch in an image is mapped to a certain code word through the clustering process and the image can be represented by the histogram of the code words.

1.1 Scale Invariant Feature Transform

Scale-invariant feature transform (or SIFT) is an algorithm in computer vision to detect and describe local features in images. The algorithm was published by David Lowe in 1999. Applications include object recognition, robotic mapping and navigation, image stitching, 3D modeling, gesture recognition, video tracking, individual identification of wildlife and match moving. However, in practice SIFT detects and uses a much larger number of features from the images, which reduces the contribution of the errors caused by these local variations in the average error of all feature matching errors. SIFT can robustly identify objects even among clutter and under partial occlusion, because the SIFT feature descriptor is invariant to uniform scaling, orientation, and partially invariant to affine distortion and illumination changes.

SIFT keypoints of objects are first extracted from a set of reference images and stored in a database. An object is recognized in a new image by individually comparing each feature from the new image to this database and finding candidate matching features based on Euclidean distance of their feature vectors. From the full set of matches, subsets of keypoints that agree on the object and its location, scale, and orientation in the new image are identified to filter out good matches.

Each cluster of 3 or more features that agree on an object and its pose is then subject to further detailed model verification and subsequently outliers are discarded. Finally the probability that a particular set of features indicates the presence of an object is computed, given the accuracy of fit and number of probable false matches.

1.2 Histogram

A histogram is a graphical representation of the distribution of numerical data. It is an estimate of the probability distribution of continuous (quantitative variable) and was first introduced by Karl Pearson. To construct a histogram, the first step is to bin the range of values—that is, divide the entire range of values into a series of intervals—and then count how many values fall into each interval. The bins are usually specified as consecutive, non-overlapping intervals of a variable. The bins (intervals) must be adjacent, and are usually equal size. If the bins are of equal size, a rectangle is erected over the bin with height proportional to

the frequency, the number of cases in each bin. In general, however, bins need not be of equal width; in that case, the erected rectangle has area proportional to the frequency of cases in the bin. The vertical axis is not frequency but density: the number of cases per unit of the variable on the horizontal axis. A histogram may also be normalized displaying relative frequencies. It then shows the proportion of cases that fall into each of several categories, with the sum of the heights equaling 1. As the adjacent bins leave no gaps, the rectangles of a histogram touch each other to indicate that the original variable is continuous.

Histograms give a rough sense of the density of the underlying distribution of the data, and often for density estimation: estimating the probability density function of the underlying variable. The total area of a histogram used for probability density is always normalized to 1. If the lengths of the intervals on the x-axis are all 1, then a histogram is identical to a relative frequency plot. Another alternative is the average shifted histogram, which is fast to compute and gives a smooth curve estimate of the density without using kernels.

1.3 Feature Detection

In computer vision and image processing the concept of feature detection refers to methods that aim at computing abstractions of image information and making local decisions at every image point whether there is an image feature of a given type at that point or not. The resulting features will be subsets of the image domain, often in the form of isolated points, continuous curves or connected regions. There is no universal or exact definition of what constitutes a feature, and the exact definition often depends on the problem or the type of application. Given that, a feature is defined as an "interesting" part of an image, and features are used as a starting point for many computer vision algorithms. Since features are used as the starting point and main primitives for subsequent algorithms, the overall algorithm will often only be as good as its feature detector. Consequently, the desirable property for a feature detector is repeatability: whether or not the same feature will be detected in two or more different images of the same scene. Feature detection is a low-level image processing operation. That is, it is usually performed as the first operation on an image, and examines every pixel to see if there is a feature present at that pixel. If this is part of a larger algorithm, then the algorithm will typically only examine the image in the region of the features.

2. Related Work

2.1 Review Stage

In content based large scale image search retrieval by bag of visual words model, retrieval is based on steps such as feature representation, feature quantization, image indexing, image scoring and post processing. The scale invariant feature transform can be used to find invariant local features. Interest point detection and feature description are the two steps involving in description. The points with high repeatability over various changes can be described as interest points. The visual appearance corresponding to the local region can be considered as the descriptors.

Visual vocabulary involves efficient indexing for local image features. An image consists of a set of local feature descriptors. The local image descriptors are high dimensional, real valued feature points. A quantization on the feature space of local image descriptors are proposed. [2]. This way any novel descriptor vector can be coded in terms of the region of the feature space to which it belongs. The standard pipeline to form a so called visual vocabulary consists of collecting a large sample of features from a representative corpus of images and quantizing the feature space according to the statistics. Often simple k means clustering is used for the quantization, the size of the vocabulary k is a user supplied parameter . in that case, the visual words are the k cluster centers. Once the vocabulary is established, the corpus of sampled features can be discarded. Then a novel image feature can be translated into words by determining which visual word they are nearest to in the feature space that is based on the Euclidean distance between the cluster centers and input descriptors. [2].

The quantizing of local image descriptors for the sake of rapidly indexing video frames with an inverted file is proposed by Sivic and Zisserman. It show that local descriptors extracted at interest points could be mapped to visual words by computing prototypical descriptors with k means clustering and having these tokens enabled faster retrieval of frames containing the same words. Csurka and colleagues first proposed using quantized local descriptors for the purpose of object categorization, the descriptors of the image are mapped to a bag of words histogram counting the frequency of each word, and categories are learned using the vector representation. An important concern in creating the visual vocabulary is the choice of data used to construct it. Generally researchers report that the most accurate results are obtained when using the same data source to create the vocabulary as is used for the classification or the retrieval task. [2]. The choice of feature detector or interest operator will also have notable impact on the types of words generated, and the similarity measured between the resulting word distributions in images. Factors to consider are the invariance properties required, the types of images to be described and the computational cost allowable.

Visual vocabulary offer a simple but effective way to index images efficiently with an inverted file. An inverted file index is just like a book where keywords are mapped to page numbers, where those words are used. In the visual word case, a table that points from the word number to the indices of database images in which that word occur. [2]. For indexing and retrieval, the visual vocabulary is built and quantized to visual word.

Clustering algorithms are generally used for generation of visual vocabulary. This involves hierarchical k means [3] and approximate k means [3]. After the visual vocabulary is

defined, the local features are quantized to visual words. To speed up the quantization process, some approaches used are k-d tree [7], vocabulary tree [4]. The approach for scalar quantization can also be used. The visual word after quantization can be represented as visual word vector.

Approximate k means use randomized k-d tree code, optimized for matching SIFT descriptors, supplied by Lowe [6]. Usually in a k-d tree, each node splits the dataset using the dimension with the highest variance for all the data points falling into that node and the value to split on is found by taking the median value along that dimension (although the mean can also be used). In the randomized version, the splitting dimension is chosen at random from among a set of the dimensions with highest variance and the split value is randomly chosen using a point close to the median. The conjunction of these trees creates an overlapping partition of the feature space and helps to mitigate quantization effects, where features which fall close to a partition boundary are assigned to an incorrect nearest neighbour. This robustness is especially important in high-dimensions, where due to the curse of dimensionality [8], points will be more likely to lie close to a boundary. A new data point is assigned to the (approximately) closest cluster center as follows. Initially, each tree is descended to a leaf and the distances to the discriminating boundaries are recorded in a single priority queue for all trees [9]. Then, iteratively choose the most promising branch from all trees and keep adding unseen nodes into the priority queue. This way, more trees can be used without significantly increasing the search time.[3].

Generation of a vocabulary tree using a hierarchical k-means clustering scheme is proposed in [4]. On the first level of the tree, all data points are clustered to a small number (K = 10) of cluster centers. On the next level, k means (with K = 10 again) is applied within each of the partitions independently. The result is K^n clusters at the nth level. For example, using a branching factor of 10 with 6 levels results in 1M leaf nodes. A new data point is assigned by descending the tree. Instead of assigning each data point to the single leaf node at the bottom of the tree, the points can additionally be assigned to some internal nodes which their path from root to leaf passes through. This can help mitigate the effects of quantization error. It is important to note that traditional flat k-means minimizes the total distortion between the data points and their assigned, closest cluster centers, whereas the hierarchical tree minimizes this distortion only locally at each node and this does not in general result in a minimization of the total distortion.

2.2 SIFT Binarization

In a given image, the interest points are represented by $\{f_i\}$ with i=0 to N-1. [1] in which N represents the total number of interest points. The true matches with the binary code is to be identified. The similarity between images are measured by the found matches. As compared with the binary space, those local features with less variations have less hamming distance in the hamming space. The sift descriptor of local feature is denoted by $d=(d_0,d_1,\ldots.d_{D-1})$. For a descriptor d, a comparison array is built to represent the relationship between its each dimension. [1]. This can be denoted by C(i,j) = 1 1 if $d_j-d_i > \alpha$; 1 0 if $\alpha \geq d_j-d_i \geq -\alpha$; 0 0 otherwise. where C represents the three dimensional comparison array with size D×D×2. And C(i,j) means the comparison result between the magnitudes in the i^{th} and j^{th} dimension of descriptor d. The comparison array is redundant since C(i,j) is highly related with C(j,i) for $i \neq j$. Therefore only those entries C(i,j) satisfying i<j and concatenate them into a string s with β=2D(D-1). S is denoted as $s=\{s_0,s_{1\ldots.}s_{\beta-1}\}$. To obtain an l bit binary code $B=\{b_0,b_1,\ldots.b_{l-1}\}$, the comparison string is encoded into l bits. [1]. Then it is partitioned into l groups. The 1 bit is assigned for each group to build l bit binary code.

2.3 Find Matching Strategy

For each feature, an integer is assigned and identity is followed by entry list of image features. Each indexed feature in the list records its image id and some other clues. It is to be verified that the target and query image regions were generated by the same object or scene region. [3]. The first thirty two bits of binary code are represented as index key and is called code word[1]. The visual word are formed by clustering sift descriptor. For that the nearest neighbor approach or approximate nearest approach is used. The feature space is generally consisted of code word and visual word respectively. The code word cells are smaller than the visual word cells. In high dimensional space, given a query feature, its true matches may be in different cells. To get improved result, it usually needs to check multiple cells that is either code word cells or visual word cells and is represented as code word expansion and visual word expansion respectively. The find matching strategy is used to perform the code word expansion and visual word expansion iteratively [1]. For code word expansion, those code words which are not more than r bits difference with the query features code word are usually checked. For visual word expansion, it is not efficient to perform visual word expansion for all query features evenly. In order to find the visual words that should be checked, small code word expansion are used. The true matches found by checking the visual word cells are used to perform code word expansion again. [1].

2.4 Image Background

The images usually consists of an object along with its background which is unique for each image feature. The descriptor of the background is also used for matching of images in testing. This provides enhanced search result as compared to the previous versions where only the shape of the images are used in image matching. The RGB color model is mainly used in computer graphics. In this model, each color appears as a combination of red green and blue. This model is called additive and colors are known as primary colors. The importance of this is that it relates very closely to the human eye perceives color.

3. Conclusion

This paper consists of image search based on the scale invariant feature transform descriptors. A visual vocabulary is created which consists of image set. The clustering is done based on k means. The code word consists of the binary features which reduces the cost of memory to a great extend. The result can be enhanced by using same data source to create the vocabulary. The code word and the visual word are used for matching.

4. Acknowledgment

The Author would like to thank Femithamol.A.M. Assistant Professor, Department of Information Technology, Ilahia College of Engineering and Technology, Muvattupuzha for her moral and technical support.

References

[1] Z.Liu, H.Li, L.Zhang, W.Zhou, Q.Tian, –Cross Indexing of Binary SIFT Codes for Large Scale Image Search," IEEE Trans. Image Processing, pp. 2047- 2057, May 2014.
[2] K.Grauman, B.Leibe, –Indexing and Visual Vocabularies," Excerpt Chapter from Synthesis Lecture Draft :Visual Recognition, pp.62-69.
[3] J. Philbin, O. Chum, M. Isard, J. Sivic, and A. Zisserman, –Object Retrieval with Large Vocabularies and Fast Spatial Matching," in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., Jun. 2007, pp. 1–8.
[4] D. Nister and H. Stewenius, –Scalable Recognition with a Vocabulary Tree," in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., Sep. 2006, pp. 2161–2168.
[5] W. Zhou, Y. Lu, H. Li, and Q. Tian, –Scalar Quantization for Large Scale Image Search," in Proc. ACM Int. Conf. Multimedia, 2012, pp. 169–178.
[6] D. G. Lowe, –Distinctive Image Features from Scale-invariant Keypoints," Int. J. Comput. Vis., vol. 60, no. 2, pp. 91–110, 2004.
[7] J. Bentley, –K-d Trees for Semidynamic Point Sets," in Proc. Symp. Comput. Geometry, 1990, pp. 187–197.
[8] U. Shaft, J. Goldstein, and K. Beyer, –Nearest Neighbours Query Performance," Technical report, 1998.
[9] S. Arya, D. Mount, N. Netanyahu, R. Silverman, and A.Wu, –An Optimal Algorithm for Approximate Nearest Neighbor Searching Fixed Dimensions," Journal of the ACM, 45(6):891–923, 1998.

Author Profile

Jerrin Varghese received the Bachelor of Technology degree in Information Technology from Mahatma Gandhi University, Kerala. He is currently doing Master of Technology degree in Computer Science and Engineering with Specialization in Information Systems from Mahatma Gandhi University, Kerala.