PROJECTE FINAL DE CARRERA

EXTENSION OF INSTANCE SEARCH TECHNIQUE BY GEOMETRIC CODING AND QUANTIZATION ERROR COMPENSATION

Studies: Enginyeria de Telecomunicació

Author: Ana García del Molino

Supervisor at NII: Shin'ichi Satoh

Co-advisors at UPC: Xavier Giró-i-Nieto; Carles Ventura.

Year: September 2013

Contents

List of Figures

Introduction

After working at the Image Processing Group at UPC with a fellowship, I confirmed I wanted my thesis to be related to image processing. Then, I found out of the **NII International Internship Program** (at the National Institute of Informatics of Japan, in Tokyo) and applied for a funded position at Prof. Shin'ichi Satoh's group.

I was aiming to work in a video retrieval evaluation campaign with Cai-Zhi Zhu, a project researcher of the group who had got the best performance in 2011 [3, 4] and one of the best in 2012 [5] for this same campaign. This was one of the reasons why I got so interested in working with them, as I was sure I could learn a lot from the experience. Luckily, I got accepted and started analysing the state of the art on the topic from Barcelona with Prof. Xavier Giró-i-Nieto and Carles Ventura as co-advisors of my thesis. A month and a half later I moved to Tokyo bringing with me an initial thesis proposal we had been elaborating during those weeks.

Unfortunately a postdoc from the lab, Sebastien Poullot, had previously worked in a similar proposal with bad results, so this first research line was discarded, as well as other possible solutions to the problematic (described in 5, *Additional Research*). Finally, two research lines were proposed: Pairwise of Interest Points, which had retrieved good results for image search, and Hamming Embedding, which had been proved to boost performance in similar applications.

During the last four months, I studied, modified, implemented and tested the two approaches, which had to be embedded in Cai-Zhi's implementation without representing any change to the original code. In this document I have tried to resume and make clear the state of the art on video and image retrieval, as well as all my work done to achieve my PFC. I want to specially present my personal contributions to the state of the art, which have been:

- Pairwise of Interest Points:
  - use of the scale information for pairing.
  - feature quantization taking into account the different concatenation possibilities.
- Hamming Embedding (HE):
  - implementation of the inverted file indexing using HE
  - consideration of the asymmetry between query and video when comparing the HE signature.

Work plan for the PFC

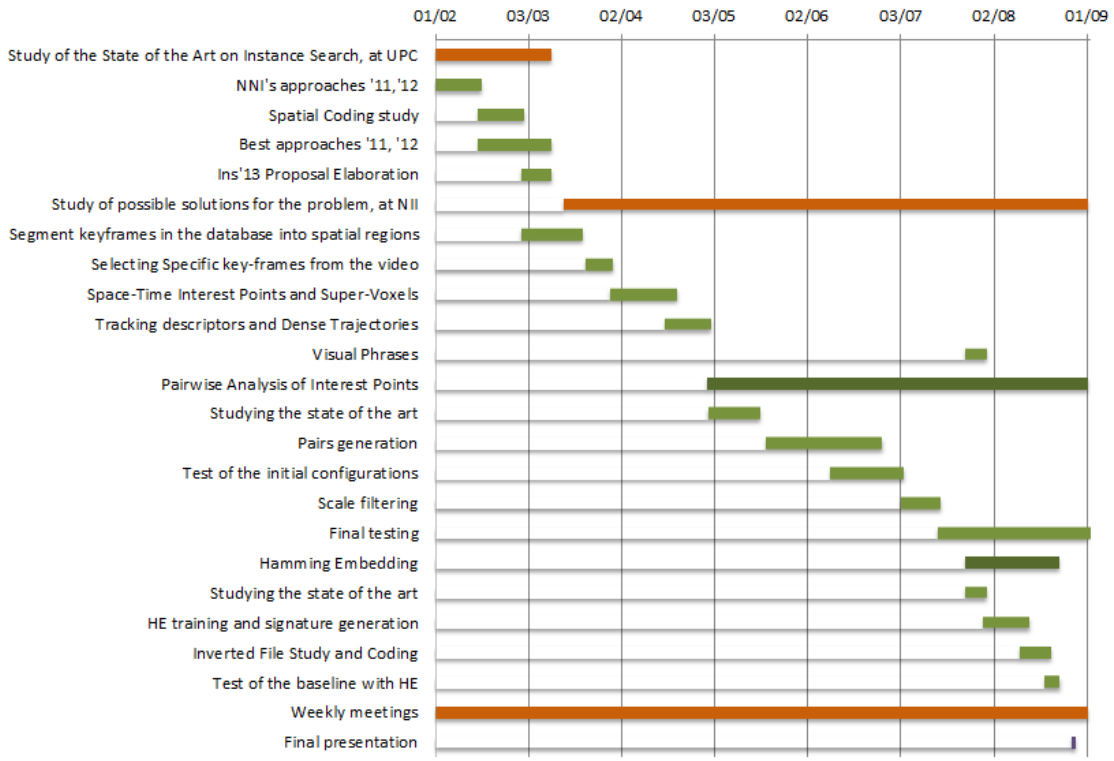


Figure 1.1: Gantt chart
The chart shows the calendar and organization of the stay both at UPC and NII. The bars in red show the whole extension of the project, split in between both institutions. The two main research topics are displayed in dark green, with their sub-tasks below.

Motivation

2.1 The Instance Search problem: description and applications

Search for visual information in big databases is a great challenge nowadays. An important need in many situations involving video collections (archive video search, personal video search/organization, surveillance, law enforcement, protection of logo use...) is to find related videos or images in an easy and effective way, and when textual tags are not accurate enough, computer vision algorithms are needed for this task.

For my PFC, I wanted to improve the performance of Cai-Zhi's algorithm for TRECVID Instance Search (best run in 2011) and submit it this year if I achieved the objective. TRECVID (TREC Video Retrieval Evaluation)[6] is an evaluation campaign whose main goal is, quoted from their website, "to promote progress in content-based analysis and retrieval from digital video via open, metrics-based evaluation. TRECVID is a laboratory-style evaluation that attempts to model real world situations or significant component tasks involved in such situations". It is very interesting for video-retrieval researchers, as they can test their approaches in real databases and compare their runs with all the other participants' in order to take new ideas from them.

Specifically, the TRECVID Instance Search (TRECVID ins) task requires finding query topics in a collection of reference videos. The query topics consist of a set of 2-6 query source images with the mask of the query topic to search, which are categorized among PERSON, LOCATION or OBJECT. As for the reference

collection, it contains a variable number of reference videos (20.982 the year 2011[4], 76.751 on 2012[5], around 300.000 this year...). The goal is to generate a ranked list of 1,000 video candidates for each topic, with the videos most likely to contain an instance of the query topic at the upper parts of the list.

2.2 State of the art for image and video retrieval

Currently, the most used technique for large image database search is Bag of Words or Bag of Features, using signatures to represent each image or video. Those signatures are then compared using different kinds of distances in order to retrieve the similar files (to a given query) in the database. Taking the definition by Stephen O'Hara and Bruce A. Draper at Introduction to the Bag of Features Paradigm for Image Classification and Retrieval:

BoF [Bag of Features] methods are based on orderless collections of quantized local image descriptors; they discard spatial information and are therefore conceptually and computationally simpler than many alternative methods.

In image analysis the great challenge is to link human semantic interpretation of images with the perceptual information a computer can handle. This is the so called "semantic gap" described by Smeulders et al in 2000[7]. In order to cover this gap, image representations easily correlated with the semantic representation are needed, and for that bag-of-words model has proved to be very useful. Descriptors are quantized into visual words with the k-means algorithm. An image is then represented by the frequency histogram of visual words obtained by assigning each descriptor of the image to the closest visual word. The general scheme for the model is the following:

1. frames are extracted from the video;
2. key-points are detected for each frame;
3. SIFT is computed in each key-point;
4. A visual vocabulary is trained out of some of those descriptors;

5. SIFT is quantized into visual words with the already trained vocabulary;

6. The signature is generated as the histogram of visual words.

7. If we want to compare images, or look for something special inside the picture, those signatures are used to check the similarity between them, through the inverted file index.

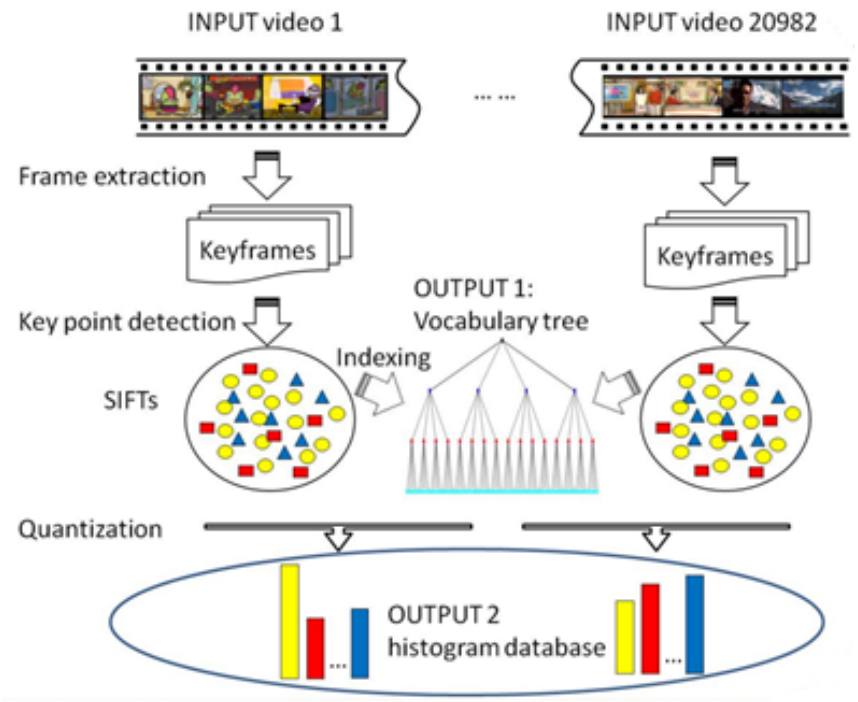


Figure 2.2: Schema for the signature generation in a whole video dataset.
1. frames are extracted from the video; 2. key-points are detected for each frame; 3.SIFT is computed in each key-point; 4.A visual vocabulary is trained out of some of this descriptors; 5.SIFT is quantized into visual words with the already trained vocabulary; 6.The signature is generated as the histogram of visual words.

In the following lines I will explain the details for each step.

Key-points detection and descriptor computation

Signatures are generated from image descriptors: vectors containing precise information on different image features. Those descriptors are not computed for the whole image, but for selected points ((x,y) position) on the image. Those points are called key-points, and a detector is needed to decide for which points the descriptor will

be computed and which area should the descriptor cover. This means the descriptor is obtained from the data surrounding the key-point, and this area is determined by some parameters stored with the (x,y) information. Each key-point i is represented as follows (1), where x, y correspond to the position and d the descriptor:

$$f(i) = (x_i, y_i, \mathbf{d_i}) \tag{1}$$

There are different ways of key-points detection:

- Dense sampling: patches of fixed size and shape are placed on a regular grid. It gives a better coverage of the image, a constant amount of points per image area, and simple spatial relations between features [8].
- Sparse sampling: focuses on key locations in the image. Depending on what you want to look for in an image, different detectors can be used, such as corner, edges or blob detectors,.
- Spatio-temporal detectors: For video, we can either apply those same key-point detectors presented above on the extracted frames, or use spatio-temporal detectors to generate a model ('signature') of that whole video [9, 10, 11].

During my project, I worked with color Scale Invariant Feature Transform descriptor, a 192 dimension vector invariant to location, scale and rotation. SIFT descriptors were computed on a series of sparse sampled key-points obtained from Hessian Affine sparse detector over the sampled frames [12].

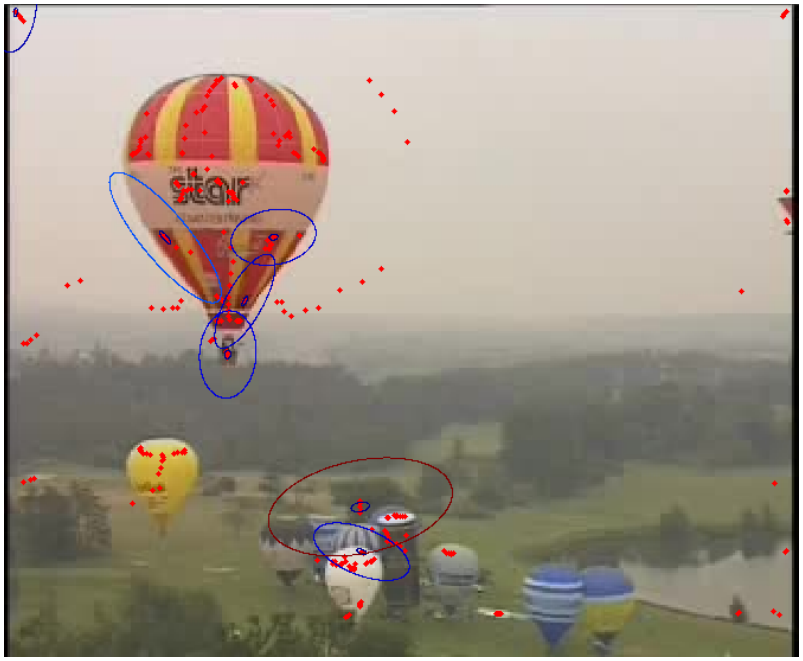


Figure 2.3: Sparse detection of key-points in one frame
The relevant area for the key-points has been represented for some of them with ellipses. The ellipses mark the boundaries for the descriptor computation for that key-point.

Quantizing with the created vocabulary

These descriptors are used to create the signature of the image or video, which will represent it. The signature is nothing else but a histogram of quantized descriptors. As descriptors are vectors and it is expensive to compare all vectors in a query to all vectors in the target database, its quantization is needed in order to create the histogram, being just one vector per image to compare. This way, similar descriptors are assigned to the same visual word, and considered as equals.

The vocabulary is created with clustering methods such as k-means, regression trees, etc. Taking a very small part of the dataset for clustering, a vocabulary is generated by giving each cluster (visual word) an id. Then, each descriptor is assigned to a visual word from the vocabulary, according to the closest cluster's center if following the hard assignment method. Soft assignment, in the other hand, assigns several visual words to the descriptor, with different weights according to the distance to the cluster centers or any other criteria.

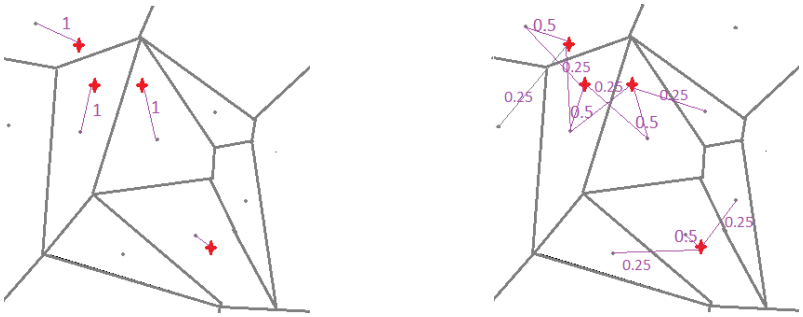


Figure 2.4: Hard and soft assignment

At the left, hard assignment gives the whole weight to the closest center. At right, this soft assignment with 3 nearest neighbors assigns different weights for the 1st or next neighbors, independently of their distance. Some other soft assignments give weights proportional to the distance of the point to the cluster center.

Signature generation

Once quantized the descriptors, for each frame a histogram of its visual words is generated, being this the signature of the frame. When obtaining this signature, though, the spatial information is lost, as the only used data is a descriptor array invariant to location, scale and rotation. In some applications, this can make the search algorithm not as good as required.

As the aim for Instance Search task is to retrieve the video, and the specific information inside each frame is not relevant, in Cai-Zhi's approach for ins2011 a video signature is then generated merging all frame histograms. This way the query signature has to be compared only to one signature per video, considering as if each video was an only image. This is done for speeding up and simplify the search.

2.3 Tf-idf weighting

Once having the signature for each video in the dataset, it is relevant to know the relative importance of each visual word in the whole dataset and in the videos independently. Some visual words are common to the whole dataset, so they have not much importance in the video signature, and some others only appear in certain videos, so should be considered more significant when searching for similarities between query and video.

The term frequency - inverse document frequency (tf-idf) is a numerical statistic used as a weighting factor which reflects how important a visual word is to a codebook and a video. The video's signature must then be multiplied by the weighting in order to be really representative of the video inside the database.

2.4 Inverted File Indexing

When retrieving videos from the database, similarities are computed one by one using histogram intersection. For huge datasets, though, this can take a lot of time, so it is interesting to search only for the relevant videos (and thus not all of them) in order to save time. This is why inverted-file index structure is employed: for each query, only the videos containing the query words are checked.

As illustrated in figure 2.5, for each visual word in the vocabulary, this structure contains the list of videos in which the visual word appears and its term frequency, that is the value of the weighted BoW for that video and visual word.

Given a query image q, the search can be interpreted as a voting scheme:

1. The similarity scores of all videos in the database are initialized to 0.

2. For each word j in the query, we retrieve the list of videos that contain this word through the inverted file. For each video i in the list, we increment its score by its term frequency already weighted. After processing all words in the query, the final score of video i provides the dot product of the BoW of video i and the query.

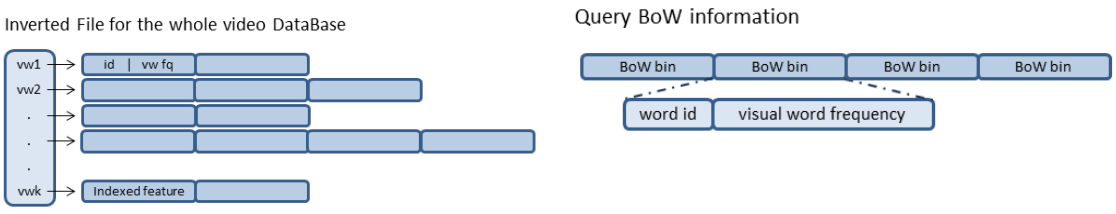


Figure 2.5: Inverted File Index structure.
Inverted file index for the complete dataset on the left and the query one at right.

Pairwise analysis of interest points

The first approach widely studied was pairing key-points in order to preserve some of the spatial information lost when creating the signature, as the new descriptors would contain information of two spatially close points.

This technique, introduced by Nobu et al. [13] with image datasets, proved to boost the performance for dense SIFT. We plan to publish the approach for sparse SIFT as a joint publication NII-UPC at the International Conference on Multimedia Retrieval, ICMR 2014 (or similar venue), so the detailed technical report is in the attached file "Pairing Interest Points for a better Signature using Sparse Detector's Spatial Information", which is the paper sketch. (refer to Appendix B)

The main idea is to pair spatially close points and store as the new descriptor the concatenation of both SIFT:

$$d_{ij} = \left(\frac{x_i + x_j}{2}, \frac{y_i + y_j}{2}, [\mathbf{d}_i \mathbf{d}_j] \right) \tag{2}$$

In order to find pairs, nearest neighbor algorithm is used. For Nobu's, using sparse SIFT, the more neighbors, the better performance is achieved. Of course, that means multiplying the number of features by the number of neighbors and, with that, the needed time for quantization increases the same way.

For such a big database as TRECVID ins', quantization had to be optimized. As all pairs came from the same small amount of points, I decided to compute distances to the clusters for only this small group, and then use those distances to get all paired features' distance, in both possible concatenation ways ij, ji. This is widely presented in the annexed document.

Initially, the vocabulary was generated out of 100 million pairs, being selected from a dataset of K=100 pairs per original point. Both the natural quantization (random order ij-ji) and the flipped one (selecting the minimum distance ij, ji as the cluster) were tested for different number of neighbors, giving worse results that the current state of the art (Cai-Zhi's algorithm):

# neigh. per point	5	10	20	35	50	Cai-Zhi's
no flip	0.3305	0.3250	0.3149	0.3086	0.3081	0.4026
flip	0.3202	0.3008	0.2705	0.2521	not tested	0.4026

Table 2.1: Results for the pairing method.
Different configurations were tested, pairing from 5 to 50 nearest neighbors for each key-point without scale filtering. The vocabulary used to quantize those paired descriptor was trained with features obtained from K=100 neighbors.

Surprisingly, the results for the flipped configuration turned out to be worse than the *natural* one.

I then realized I was pairing not only close descriptors, in many cases the paired features were very far one to the other because my key-point detector was sparse and not dense, so I decided to pair only points within a certain neighborhood. This neighborhood is established using the detectors spatial information, which provides the surroundings of the point for which the SIFT descriptor has to be computed. A new vocabulary was then trained with the new paired descriptors.

Figure 2.6: Paired key-points in a frame.
In the image at left, original pairs are shown. At right pairs with scale filtering can be seen.

Figure 2.6 shows the differences in pairing using the scale information (right) or not (left) for k=20 neighbors. The color ellipsoids in the right show the neighborhood for some of the key-points (represented with blue circles). The red crosses are the center position between the paired key-points. The wrong pairs in red can be appreciated on the left in the middle of the sky. On the right, those pairs are excluded, as the pairing points are not inside the neighborhood. This is also notorious inside the green neighborhood on the right: for the central point, only three key-points below can be paired, all pairs above the center disappear.

The computation requirements for this method are very heavy so the experimentation on large datasets can last for days. Even though the results could be good, as the Instance Search database was so big this year, we could not afford to submit this approach because of the needed time. We then decided to focus on the Hamming Embedding approach since the results were more promising. We did post-pose this research line in order to continue testing and fixing the code after TRECVID ins submission on August 23th.

Right now, the available results are not as good as expected and further modifications and tests need to be done in order for the paper to be accepted. I will keep working on that from UPC in collaboration with NII.

During my stay at NII, it was also agreed that my co-advisor Carles Ventura will keep exploring this approach in his coming visit to NII. He is currently pursuing his Ph.D. at the Image Processing Group and his thesis topic is closely related to this part of my work. He will be taking my place in the group for 6 months, exploring the opportunities of bundling interest point features based on image segmentations, instead of the distance-based pairings I have explored.

Hamming Embedding

When we realized we could not handle the pairwise technique because of the coming deadline and the time needed for computation, I was proposed to implement the existing Hamming Embedding (HE) for our baseline for Instance Search 2013. As HE provides binary signatures that refine the matching based on visual words, we believed it would boost the performance. Experimentations backed this assumption.

That was a big challenge for me because of the short time available (hardly 3 weeks) and the huge amount of work and study to do: I first had to study what Hamming Embedding was and how it had to be implemented, then train the system with Matlab and generate the needed data. I also had to modify the inverted file codes (I had never seen before inverted file implementations, nor coded in C++) in order to adapt it to Hamming Embedding preserving the original structure. Finally, it had to be tested on 2011 TRECVID ins dataset at least one week before the deadline.

The reference bibliography [14] does not provide details on how to search in the inverted files using Hamming Embedding, so I got to some conclusions which lead me to code it in a way which had never been explored before. This is why, not only will this implementation will be reported to this year's TRECVID Workshop, but also be submitted to ICASSP'14 (IEEE International Conference on Acoustics, Speech and Signal Processing) call for papers, due to October 27th. The sketch is attached in "Study of the Hamming Embedding Signature Symmetry in Video Retrieval". (refer to Appendix A)

The approach is presented and explained in the paper sketch. As for the developed files, they do as follows:

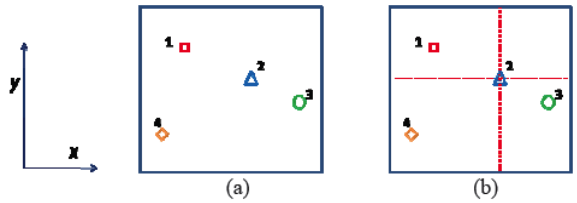
- `HE_projection.m`: trains the HE boundaries for the specified database and feature. Loads quantized descriptors from all videos in the dataset in order to generate the boundaries for each cluster.
  - Inputs: the directory of the SIFT descriptors and the directory of its quantization.
  - Output: matrix P and T (projection and boundaries) stored in one file.
- `HE_signature.m`: with the trained system, generates the signature for every descriptor in the database.
  - Inputs: the directory of the SIFT descriptors; the directory of its quantization; and the previously stored file with P and T.
  - Output: the HE signature for each key-point; the HE signatures for the whole video sorted by its quantized value, with this other info.
- `ranking_HE.m` is the video retrieval file. Generates the inverted file using the previous quantized features and its HE signatures; quantizes and generates the HE signature for the query features; computes similarities between query and videos; sorts the retrieved video list; and finally computes the mAp (mean average precission) for the approach.
- **Inverted Files**: Function `void ivBuildInvFile_HE` stores the signatures info for every word and video. Function `mxArray* ivSearchInvFile_HE_l1` compares all signatures from the word in the query with all signatures for that word in the dataset.

Additional research

In addition to the intensive work on pairing features and Hamming Embedding, my thesis has also involved exploring other research directions aimed at solving the Instance Search task from novel perspectives. This section provides an overview of these efforts, with an introductory description as well as a discussion about why they were discarded. They have helped me understand the challenges and uncertainties related to a research activity, which require flexibility and broad scopes to pose the right questions and choose a promising research path.

5.1 Spatial coding

In image-based object retrieval, the image variations due to 3D view-point change are a great challenge. When having huge databases, visual words may not be discriminative enough. The approach in [15] adds Spatial Coding after quantization, in order to improve the BoW performance. Spatial coding encodes the relative positions between each pair of features in the image, so that when finding a match in between images, this is checked in order to discard, or not, the match. This technique can be very efficiently performed, but the whole spatial maps of all features in an image require a large amount of memory resources.



$$Xmap(i,j) = \begin{cases} 0 & if\ x_i < x_j \\ 1 & if\ x_i \geq x_j \end{cases} \quad Ymap(i,j) = \begin{cases} 0 & if\ y_i < y_j \\ 1 & if\ y_i \geq y_j \end{cases}$$

$$Xmap = \begin{bmatrix} 1 & 0 & 0 & 1 \\ 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 \end{bmatrix}; \quad Ymap = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Figure 5.7: An illustration of spatial coding for images features
(a) shows an image with four features; (b) shows the image plane division with feature 2 as the origin point. [15]

5.2 Selecting key-frames from the sampled video

Image-based analysis techniques applied to video require previous extraction of keyframes to be processed. Keyframes could be consciously selected between uniformly spaced frames by detecting gradual and abrupt cuts, as in [16]. So that, less keyframes would be extracted and the computational cost afterwards would be smaller. This option was soon discarded, as the change was not as relevant as we thought.

5.3 Segmenting videos to search sequentially

This search approach proposes a multi-scale solution for video search, adopting search criteria that especially focus on efficiency. Our proposal aimed to discard at first many videos, and then look for the query topic inside the remaining ones. For that, video frames are segmented into spatial regions, and then selective search based on hierarchical partitions and features is done: the query signature is compared first

to the global video signature, and then to the signatures of the segmented parts, as in [17] and [18]. Finally, color would be checked with HSV.

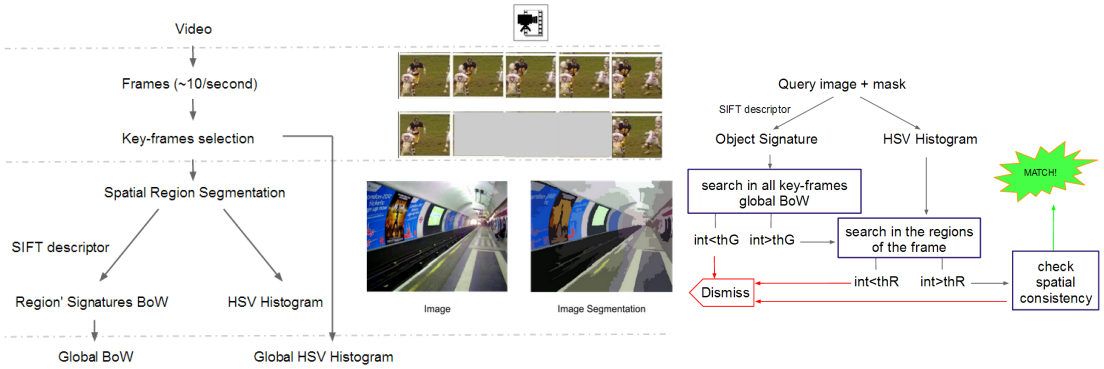


Figure 5.8: Scheme of the procedure for the segmented video frames.
At the left the off-line procedure is shown. The on-line search is at right. [17, 18]

This approach had already been tested by Sebastien Poullot for instance search in 2010, providing bad results.

5.4 Spatio-Temporal Interest Points (STIPs)

The approach presented in [11, 10] selects key-points according to space properties and changes between frames. For steady backgrounds, no key-points were detected, so it is only useful if combined with the points sparsely detected from one sampled frame of the video. If not, only moving objects would be represented in the Bag of Words.

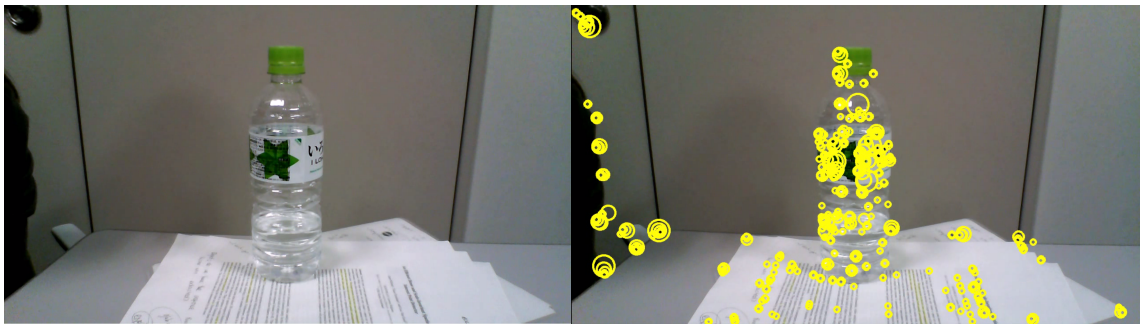


Figure 5.9: STIP detector performance
The steady image at the left has no points detected, while the moving image in the right has edges marked.

5.5 Tracking descriptors

Spatio-temporal interest points encode video information at a given location in space and time. In contrast, trajectories track any given spatial point over time and, thus, giving one visual word to the whole trajectory can reduce the quantization error. Sun et al. [19] compute trajectories by matching SIFT descriptors between two consecutive frames, considering a matching descriptor that one with a similarity higher than a threshold. They impose a unique-match constraint among the descriptors, and discard matches that are too far away in term of spatial coordinates: the geometric distance between the points is computed, and all matches whose distance is higher than 64 pixels are discarded. This process is applied recursively on up to 25 frames (the length of the trajectories will be 25 or less) and for each key-point separately. Matches that extend over several frames are used to built a motion trajectory of the SIFT salient point.

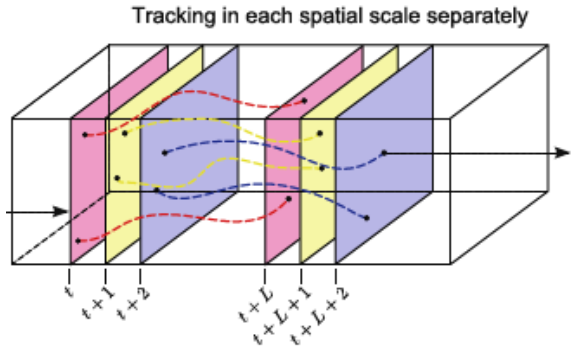


Figure 5.10: Trajectories generation [1]
Each key-point is tracked from one frame to the following, with a maximum length of L=25.

Once the trajectory is estimated, it is characterized by a descriptor by computing the average of the SIFT inside the trajectory. Finally, this descriptor is quantized into one only visual word. This approach was reviewed by Wang et al. in 2013. [1].

Our proposal included obtaining not the average, but the median so we would discard outliers. The problem we found for using this approach is that obtaining the trajectories in such a large dataset as TRECVID was very expensive computationally, as it had to be done one descriptor by one, and could not be done for the whole frame at a time using matricial operations.

5.6 Geometry-Preserving Visual Phrases

Visual Phrases aim at improving Spatial Coding approaches. As described in 2013[2], visual phrases (group of visual words) allow to encode spatial information of the objects in the image. The image is divided into bins, and the object is encoded according to the (x,y) distance (in bins) between its key-points. This information can then be used as a combination with BoW. Even though I started exploring the approach, another fellow at NII with more background in this specific topic decided to test its performance for instance search, so I was encouraged to study other research directions.

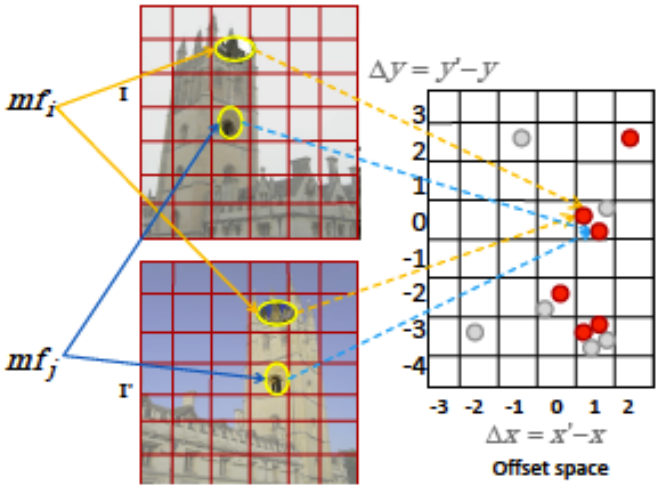


Figure 5.11: Illustration of the min-hash method with GVP. [2]
Visual Phrases encoding and representation. For this two images, the relative distance between exemplified visual words is the same, so a matching will be set.

Conclusions

When I started the PFC at Barcelona in late January, I didn't expect the research to go as it eventually went. Even though I had already studied Cai-Zhi's approach in Barcelona, the reality was completely different when facing the codes. We had also been elaborating a possible solution for the problem, and I was expecting to get to Japan and start working on it right away. The proposal was then discarded and I had to come back to earth, realizing it was not going to be that evident. Nevertheless, the work done in Barcelona studying the state of the art was very useful for thinking of other possibly interesting approaches.

The stay at NII has widen my view on research field and the researcher's need of creativity and autonomy. I soon realized that even though the group had weekly meetings for discussing the individual current research, approach and results, I would have to research by myself without much guidance the rest of the week, which was not easy at first. Hopefully I had my UPC co-advisors Xavi and Carles always there suggesting me what to read or what to try.

I learned after a couple of weeks to search for codes or algorithms on libraries provided on the internet, or find data in the servers from information on Cai-Zhi's codes, as he would always ask me to understand the code and find everything by myself. I also realized it would be faster to implement what was written in the paper after understanding it, rather than trying to find any code to guide me. This is one of the most important things I have learnt from this research experience.

Having group meetings every Wednesday allowed me to hear of many other image analysis approaches and applications I had never heard about before. This lead me to read many literature I thought could help our problem, and discard it afterwards, learning a lot from that reading, testing and failure. This being said, I enjoyed

researching, even though it took me a while to realize how research had to be faced.

Also, as the National Institute of Informatics was an international multidisciplinary center, I had the opportunity of meeting people from all over the world, each of them researching in something completely different to my research in concept, objectives or ways, and I found a lot of people (students and postdocs) willing to help or teach even though it was not their research topic or field. For all of it, I must say the experience was great, and I would definitely take the opportunity again if I had to, knowing I would have to face most of the challenges on my own without receiving much feedback from the supervisors.

During these months, I have studied and tested many interesting technologies such as STIP, which I believe can be very useful for retrieving from video queries; or sift trajectories, very used in action retrieval. They were not useful, though, for the problem we wanted to solve, either because of the extracted data, or for the needed time and cost of computation. As for the studied techniques, pairing neighboring points turned out not to be as good for our problem as we thought, and Hamming Embedding can get hard to run for such a big database.

In one hand, pairing spatially close descriptors technique results expensive, as it takes huge computation time. Even though this can be solved training way in advance, the difficulty of quantization is still there. There are too many points to quantize, and a tree is not the best option for that, this is why the distance matrix is used. But this quantization method has a limitation: the vocabulary can not be as large as 1.000.000 words if we still want to have some memory usable in the server. This is why 10.000 words were used in our tests instead. Of course, more powerful servers can be used, but there is always a time and memory restrictions.

In the other, Hamming Embedding performs promising, but our results still are not so much good. Also, saving all this binary masks information in the inverted file uses many memory: for the whole dataset of 2011, up to 160Gb of ram memory were needed only for building the inverted file. Too much for a normal server (I had to ask for permits on a 500Gb server in order to run it).

Having the appropriate resources, and for this kind of problem, mixing both approaches could provide very good results, as the limitation of the small vocabulary

for paired descriptors can be palliated with Hamming Embedding's adaptability, boosting the performance. This is a research line I want to explore in a near future, as I find it very interesting and motivating. Even though I would have liked to do so at NII sharing results and conclusions with all the researchers in the group, unfortunately the internship was only 6 months long and there was not enough time for all the topics I would like to explore. Even if I had more time, I am sure I would always find some other promising and interesting line to research in.

Acknowledgements

First of all, I would like to acknowledge the ETSETB, for forming me during those hard five years, and the National Institute of Informatics of Japan for hosting and mentoring me during those last 6 months. Also, to the IEEE BCN Student Branch, DAT, Distorsió and l'Oasi, and so many other student associations in my school for showing me the other side of university.

I want to express my very great appreciation to my co-advisors Xavier Giró-i-Nieto and Carles Ventura, without whom this experience would have been much difficult, for being always so much willing to solve my doubts and guide me. Also to Professor Shin'ichi Satoh, for offering me this great opportunity and hosting me in his research group; to Cai-Zhi Zhu for showing me all his work and teaching me so many things on how to research; and to all the group members for making me feel so comfortable working with them. Finally, to Ferran Marqués, whose advice has always been very valuable, and who encouraged me not to be afraid of anything and go for it.

Last but not least, I want to deeply thank my family for being there exposing and discussing all pros and cons on every important choice I have ever had to make, helping me out on picking the right one, and supporting me with the final resolution; to Quim, for bringing some sanity to my decisions and actions, sometimes so impulsive. And to my friends: Miriam, Genís, thank you for being always there cheering me up when needed, or showing yourself happy and friendly jealous for my achievements; to my great cinema companions on thursday evenings, thank you all very much for helping me out disconnecting from all the rest; to my fellows *upesé*, for sharing so many labs, lessons, exams and study hours, it would not have been the same without this companionship. Thank you all very much.

Bibliography

[1] H. Wang, A. Kläser, C. Schmid, and C.-L. Liu, "Dense trajectories and motion boundary descriptors for action recognition," *International Journal of Computer Vision*, pp. 1–20, 2013. (document), 5.10, 5.5

[2] Y. Zhang, Z. Jia, and T. Chen, "Image retrieval with geometry-preserving visual phrases," in *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*. IEEE, 2011, pp. 809–816. (document), 5.6, 5.11

[3] D. Le, C. Zhu, S. Poullot, and S. Satoh, "National institute of informatics, japan at trecvid 2011," in *TRECVID Notebook Papers/Workshop*, 2011. (document)

[4] P. Over, G. Awad, M. Michel, J. Fiscus, B. Antonishek, W. Kraaij, A. F. Smeaton, and G. Quéenot, "Trecvid 2011 – an overview of the goals, tasks, data, evaluation mechanisms and metrics," in *Proceedings of TRECVID 2011*. NIST, USA, 2011. (document), 2.1

[5] P. Over, G. Awad, M. Michel, J. Fiscus, G. Sanders, B. Shaw, W. Kraaij, A. F. Smeaton, and G. Quéenot, "Trecvid 2012 – an overview of the goals, tasks, data, evaluation mechanisms and metrics," in *Proceedings of TRECVID 2012*. NIST, USA, 2012. (document), 2.1

[6] A. F. Smeaton, P. Over, and W. Kraaij, "Evaluation campaigns and trecvid," in *MIR '06: Proceedings of the 8th ACM International Workshop on Multimedia Information Retrieval*. New York, NY, USA: ACM Press, 2006, pp. 321–330. 2.1

[7] A. W. Smeulders, M. Worring, S. Santini, A. Gupta, and R. Jain, "Content-based image retrieval at the end of the early years," *Pattern Analysis and*

Machine Intelligence, IEEE Transactions on, vol. 22, no. 12, pp. 1349–1380, 2000. 2.2

[8] T. Tuytelaars, "Dense interest points," in *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*. IEEE, 2010, pp. 2281–2288. 2.2

[9] C. Xu and J. J. Corso, "Evaluation of super-voxel methods for early video processing," in *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*. IEEE, 2012, pp. 1202–1209. 2.2

[10] G. Willems, T. Tuytelaars, and L. Van Gool, "An efficient dense and scale-invariant spatio-temporal interest point detector," in *Computer Vision–ECCV 2008*. Springer, 2008, pp. 650–663. 2.2, 5.4

[11] I. Laptev, "On space-time interest points," *International Journal of Computer Vision*, vol. 64, no. 2-3, pp. 107–123, 2005. 2.2, 5.4

[12] K. Mikolajczyk and C. Schmid, "An affine invariant interest point detector," in *Computer Vision—ECCV 2002*. Springer, 2002, pp. 128–142. 2.2

[13] N. Morioka and S. Satoh, "Building compact local pairwise codebook with joint feature space clustering," in *Computer Vision–ECCV 2010*. Springer, 2010, pp. 692–705. 2.4

[14] H. Jegou, M. Douze, and C. Schmid, "Hamming embedding and weak geometric consistency for large scale image search," in *Computer Vision–ECCV 2008*. Springer, 2008, pp. 304–317. 2.4

[15] W. Zhou, Y. Lu, H. Li, Y. Song, and Q. Tian, "Spatial coding for large scale partial-duplicate web image search," in *Proceedings of the international conference on Multimedia*. ACM, 2010, pp. 511–520. 5.1, 5.7

[16] P. Kelm, S. Schmiedeke, and T. Sikora, "Feature-based video key frame extraction for low quality video sequences," in *Image Analysis for Multimedia Interactive Services, 2009. WIAMIS'09. 10th Workshop on*. IEEE, 2009, pp. 25–28. 5.2

[17] V. Vilaplana, F. Marques, and P. Salembier, "Binary partition trees for object detection," *Image Processing, IEEE Transactions on*, vol. 17, no. 11, pp. 2201–2216, 2008. 5.3, 5.8

[18] K. E. van de Sande, J. R. Uijlings, T. Gevers, and A. W. Smeulders, "Segmentation as selective search for object recognition," in *Computer Vision (ICCV), 2011 IEEE International Conference on.* IEEE, 2011, pp. 1879–1886. 5.3, 5.8

[19] J. Sun, X. Wu, S. Yan, L.-F. Cheong, T.-S. Chua, and J. Li, "Hierarchical spatio-temporal context modeling for action recognition," in *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on.* IEEE, 2009, pp. 2004–2011. 5.5

Appendix

Contents

Appendix A:

Study of the Hamming Embedding Signature Symmetry in Video Retrieval

e-mail correspondence between Shin'ichi Satoh, Cai-Zhi Zhu and me:

Dear Sensei, dear Ana

I think we can probably start the asymmetric idea on Hamming embedding, and target a ICCASP paper. I have asked Herve in INRIA about that, but get no answer, and only learned that he is very much busy recently, so he probably won't take that direction. @Sensei, how do you think?

FYI, the attached document is our ICCV paper, please consider how to combine these two idea.

Finally, I would like to say that your hard work impress me, and also you achieved quite good performance during your internship. Thank you very much for your contribution, and I'm sorry I cannot help you much.

Best,

Cai-Zhi Zhu

Hi all,

Yes, I think we can go ahead. Thanks!

Shin'ichi Satoh

Thanks, Sensei.

So, Ana, do you think you can manage to submit a paper to ICCASP, the deadline is Oct. 27, 2013. Time should be enough for you if you can fully work on that. In this paper, you will need to answer following questions:

1. (Must) Provide experimental evidences to show that HE is better that the

baseline without HE.

2. (Must) Read our ICCV paper and implement the asymmetrical HE, and get evidence that aysm-HE is even better than HE.

3. (Optional) generalize HE to handle soft-assignment.

All these should be verified on at least two datasets, ideally INS2011 and INS2012. If these two datasets are too difficult and we cannot see any improvement, we can turn to Oxford datasets. FYI, I usually generalize my code to handle both TrecVid INS datasets and Oxford datasets, so you can easily tailor my code, in the folder ~caizhizhu/per610a/vgg/code, to fit your need.

I would recommend you to start from Oxford building datasets first, since they are cleaner and smaller compared with INS datasets, and we should observe improvement on these datasets, otherwise something may go wrong and we have to give up this direction, of course it's quite unlikely according to your preliminary results.

Finally, I have to say that, if you would like to try this direction, I can only help you in giving my advice on algorithm design and paper composition, since I will be fully committed until Nov. 1 on other paper work, you will take all related work including coding, experimenting, and drafting the paper. Do you think you can manage it?

Cai-Zhi Zhu

STUDY OF THE HAMMING EMBEDDING SIGNATURE SYMMETRY IN VIDEO RETRIEVAL

Ana García del Molino

UPC, Barcelona – NII, Tokyo

ABSTRACT

To be completed before submission.

***Index Terms*—** Hamming Embedding, Bag-of-Words, video retrieval, asymetric.

1. INTRODUCTION

Search for visual information in big databases is a great challenge nowadays: an important need in many situations involving video collections is to find related videos or images in an easy and effective way. Currently, the most used technique for large image database search is Bag of Words or Bag of Features [1], using *signatures* to represent each image or video.

Bag-of-Words model include quantization, which is highly useful for sparse coding (saving memory) and efficient indexing afterwards. Descriptors are quantized into visual words, being those defined by a vocabulary usually trained with k-means algorithm. This is decisive in the performance of the algorithm.

There are two possible ways of assigning visual words to features: hard or soft assignment. Hard assignments give each descriptor the id of the closest cluster's center, while with soft *assignment*s the descriptor is related to multiple visual words, with different weights according to the distance to the cluster centers or any other criteria (Fig. 1).

Two main types of errors occur when matching after quantization: either two descriptors are matched as the same visual word when they actually have little in common; or two very similar descriptors are quantized to different words and thus not matched.

The first error can be reduced with larger vocabularies (smaller clusters), but this forces more errors of the second type. While hard assignments are computationally cheap, soft assignments can reduce the effect of the second type of error. Many approaches show that big vocabularies using soft assignment perform better than small ones with either assignment [2, 3], as is seen in TRECVID workshop every year [4].

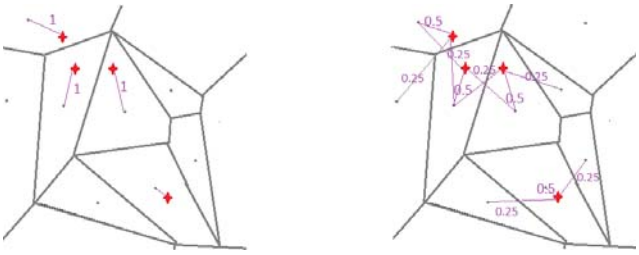


Fig. 1: At the left, hard assignment gives the whole weight to the closest center. At right, this soft assignment with 3 nearest neighbors assigns different weights for the 1st or next neighbors, independently of their distance. Some other soft assignments give weights proportional to the distance of the point to the cluster center.

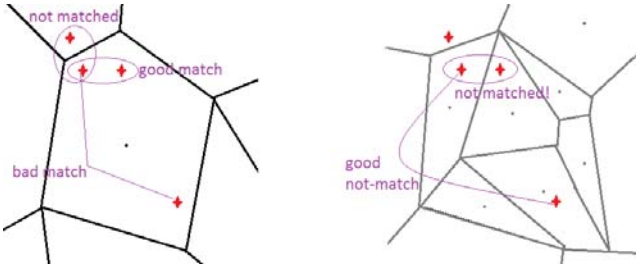


Fig. 2: Bad matches or missing matches due to a small vocabulary (left) or big (right)

Hamming Embedding [5] aims to solve this compromise by providing binary signatures that refine the match based on visual words. It encodes the location of the descriptor inside the cluster, and then keeps or dismisses the match according to the Hamming distance between the descriptors. The location is encoded in a way that the encoded vectors' Euclidian distance is equivalent to the Hamming distance in the descriptors space.

This approach then takes into account the distance between matched descriptors, and discards the match if that distance is greater than a threshold, creating relative cluster boundaries. This way, both errors can be mitigated when using a smaller vocabulary, or only the first if the vocabulary size is not reduced.

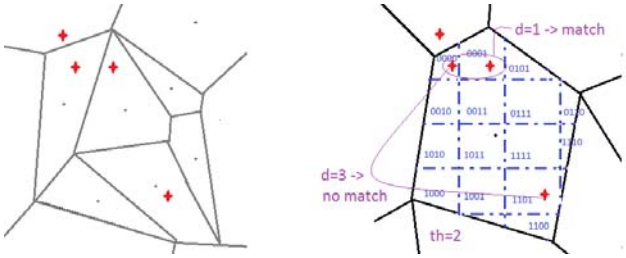


Fig. 3: Comparison of a big vocabulary to a smaller one using HE. With Hamming Embedding, once reduced the vocabulary size, many good matches are preserved because of the smaller amount of fixed boundaries (still some are missing), and bad matches are discarded because of their distance, even being in the same cluster.

In this paper a new search method is introduced: when searching for the similarity between query and target video, both directions are explored. First the number of query features matching features in the video, and vice versa. The minimum between the two values is taken as the number of matched features.

The results are then compared to the alternative search: only considering the matches for the query features. Both results are intended to beat the baseline performance. Our baseline is NII's 1st run in TRECVID ins 2011 [4]: BoW model with 1.000.000 visual words, using SIFT features sparsely extracted using Harris-Laplace and MSER.

2. GENERATION OF THE HAMMING EMBEDDING SIGNATURE

In order to generate these binary signatures, the system needs to be trained. The boundaries for each sub-cluster are different, and are defined once generated the vocabulary:

1. Right after training the vocabulary a d_b x D orthogonal projection matrix P is generated out of a random Gaussian matrix filtered with QR (taking only the first d_b rows of Q). It is proved that higher values of d_b get better performance but need more memory. In our approach d_b=64 (8 bytes) will be considered.

2. All descriptors used for the k-means are projected with P, and a matrix T (d_b x k) is generated with the median of the projections for each word. This matrix T sets the boundaries for each sub-cluster. Being $q(x)$ the visual word assigned to the descriptor x, z the projection Px and h each dimension of the HE signature:

$$T_{h,l}=median\{\ z_h \mid q(x)=l\ \} \qquad (1)$$

After quantizing each descriptor of the whole dataset, both T and P are used the following way:

1. The descriptor x is projected generating the d_b dimension vector $z=Px$.
2. z is compared to $T_{q(x)}$. Dimensions where $z_h > T_{h,q(x)}$ take value 1 or 0 otherwise:

$$b_h(x) = \begin{cases} 1 & if\ z_h > T_h, q(x) \\ 0 & otherwise \end{cases} \qquad (2)$$

This mask b_x is then used for assessing the matching. When two elements x, y, assigned to the same visual word, are compared, their masks $b(x)$, $b(y)$ are first computed. If $d(b(x),b(y))>th$ the match is dismissed, where th has a value in between 0 and d_b. As those signatures are binaries, the distance is computed with a fast *XOR* operator.

3. INVERTED FILE IMPLEMENTATION

In order to optimize the search in such large datasets, inverted-file index structures are used. This indexing allows searching only in the relevant videos for that query, saving computational cost and time. For each query, only the videos containing visual words in the query are checked.

HE signature is included in the inverted file in order to compute distances between the binary signatures of the matching words. For each visual word in the vocabulary, the new structure contains the list of videos in which the visual word appears with its term frequency and binary signatures:

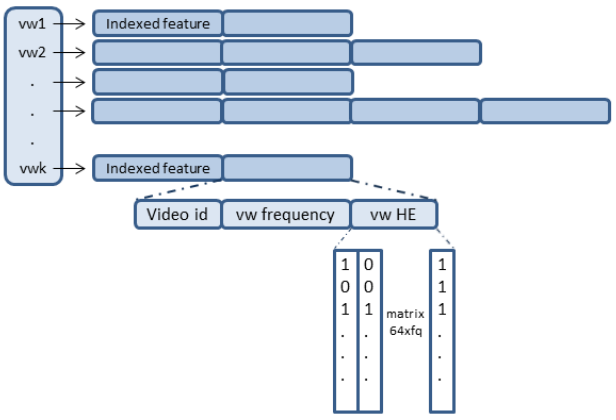


Fig. 4: Inverted File structure for the whole database using Hamming Embedding. A matrix containing all features' binary signature for that visual word is added to the file.

Analogously, the query file is created as shown in fig. 5.

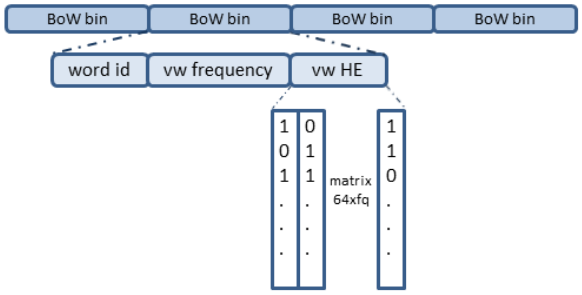


Fig. 5: Query signature for HE.

When searching, every HE signature from the query is compared with the ones for the matching words in the target video. Their distance is computed and, if it is smaller than the threshold *th*, the match is considered. The final score is set as the minimum between the number of query signatures matched to the video, and the number of video signatures matching to the query, multiplied by the tf-idf weighting value. The *term frequency – inverse document frequency* parameter is a numerical statistic used as a weighting factor which reflects how important a visual word is to a codebook and a video.

Being *x* a descriptor in the query *q*, and *y* a descriptor in a target video *v*,

$$match(x) = \begin{cases} 1 & if\ \exists\, d(b_x, \forall\, \{b_y \mid q(y) = q(x)\} < th, \\ 0 & otherwise \end{cases} \quad (3)$$

$$match_{q,v}(l) = \sum_{\forall x} (match(x) \mid q(x) = l) \quad (4)$$

Analogously,

$$match(y) = \begin{cases} 1 & if\ \exists\, d(b_y, \forall\, \{b_x \mid q(x) = q(y)\} < th, \\ 0 & otherwise \end{cases} \quad (5)$$

$$match_{v,q}(l) = \sum_{\forall y} (match(y) \mid q(y) = l) \quad (6)$$

The similarity between query *q* and video *v* is given by

$$sim_{q,v} = \sum_{\forall l} tf\_idf(l) \cdot min(match_{q,v}(l), match_{v,q}(l)) \quad (7)$$

An alternative implementation considers only the matches for the query, being asymmetric:

$$sim_{q,v} = \sum_{\forall l} tf\_idf(l) \cdot match_{q,v}(l) \quad (8)$$

Or a weighting compensation using the normalization parameter for the video and query bows:

$$sim_{q,v} = \sum_{\forall l} tf\_idf(l) \cdot min(match_{q,v}(l)/norm_q, match_{v,q}(l)/norm_v) \quad (9)$$

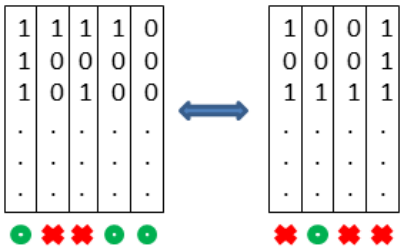


Fig. 6: Similarity search using HE. All binary signatures for one word in the query (left) are compared with the HE signatures stored for that same word in the inverted file for the target video (right), and assumed a correspondence if any distance is bigger than the threshold. Each possible match is represented by a red cross if it is discarded (0), or a green circle if kept (1). In the example, $match_{q,v}$=3 and $match_{v,q}$=1. For this proposal, the similarity would be of 1, while in the first alternative implementation it would be of 3. As for the compensated one, it depends on the full db.

4. EXPERIMENTS AND RESULTS

This method was tested in instance search 2011 dataset with a vocabulary size of 1.000.000 words, as in the baseline. As in the reference paper [5], *th* was given values between 25 and 35. After viewing the results, it was tested again for values up to 51. The retrieval results were better than the baseline, having that one *map*= 0.5260, and the approach for *th*=45 a *map*= 0.6030.

It was also tested for the alternative implementations (asymmetric HE, (8), and compensated HE, (9)), retrieving the following results:

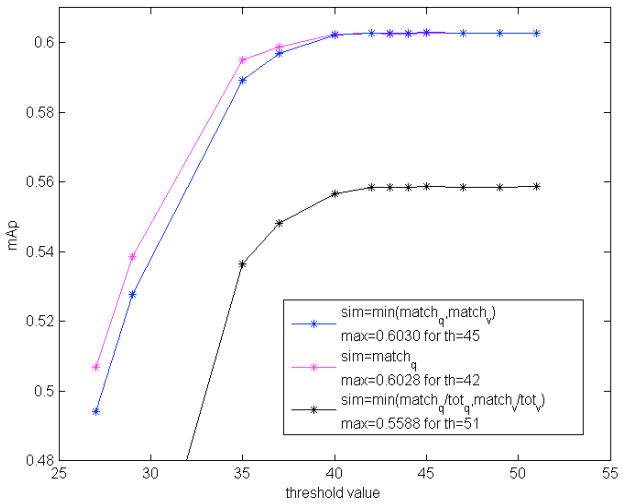


Fig. 7: Plotted *map* results for the tested configurations. The original idea is plotted in blue, magenta for the asymmetric and black for the compensated one.

Further experiments with other databases need to be done in order to ratify the results.

5. CONCLUSIONS

As shown in figure 7, the best result is found for the original implementation (blue) with *th*=45, while for lower values of *th* the asymmetric implementation (magenta) seems better. It is important to note that there is not much difference between both configurations' results: as in general the number of HE matches in the query will be inferior to the HE matches in the whole video, the original implementation will normally end up being equivalent to the asymmetric one.

Finally, the compensated similitude (black) is not that good, even though still gets better results than the baseline. It is still not clear why this happens; more tests need to be performed on the matter.

It is certain that using Hamming Embedding improves the performance, but for our application not as much as in [5]. That might be because of the vocabulary size. In our experiments, the HE procedure has been applied to points quantized with a vocabulary the same size as the baseline we are comparing the performance to. If using a smaller vocabulary, it would probably boost the performance even more. This is being tested and still in development.

6. REFERENCES

[1] J. Sivic and A. Zisserman. "Video Google: A text retrieval approach to object matching in videos", In ICCV, 2003.

[2] Nist´er, D., Stew´enius, H.: "Scalable recognition with a vocabulary tree", In: CVPR. (2006) 2161–2168

[3] J. Philbin, O. Chum, M. Isard, J. Sivic, and A. Zisserman. "Object retrieval with large vocabularies and fast spatial matching", In CVPR, June 2007.

[4] D Le, C Zhu, Sebastien Poullot, and S Satoh. National Institute of Informatics, Japan at TRECVID 2011. In TRECVID Notebook Papers/Workshop, 2011. (document)

[5] H. J´egou, M. Douze, and C. Schmid. "Hamming embedding and weak geometric consistency for large scale image search", In *ECCV*, October 2008.

Appendix B:

Pairing Interest Points for a better Signature using Sparse Detector's Spatial Information

PAIRING INTEREST POINTS FOR A BETTER SIGNATURE USING SPARSE DETECTOR'S SPATIAL INFORMATION

Ana García del Molino

UPC, Barcelona – NII, Tokyo

ABSTRACT

To be completed before submission.

Index Terms—*To be completed before submission.*

1. INTRODUCTION

Very large video databases require effective search methods, able to retrieve correct matches in short time. Image signatures are a common way of representing an image, giving us general information of the picture. But even though the Bag of Words (BoW) approach makes it fast to evaluate the similarity between images, the spatial distribution of the interest points is lost after the quantization process when building the signature.

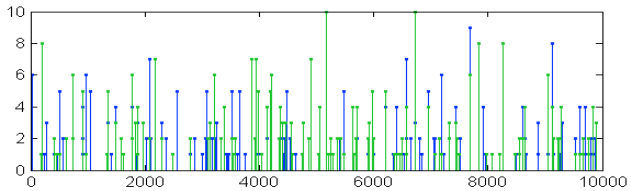
In order to solve this limitation, many spatial coding approaches have been explored [1, 2], but most of them are computationally expensive and require large memory resources. For this reason they are not useful for very large datasets. Our approach pairs spatially close interest points, as has been explored in [3,4,5]. A vocabulary is trained with previously paired descriptors, so that the centers have double dimension. The new features are the result of a concatenation of the features associated to each point in the pair, and are then quantized with the new vocabulary.

This solution was already introduced by Nobuyuki et al. in [6] using a dense sampling. The novelty of our work is the use of a sparse sampling combined with the pairing approach.

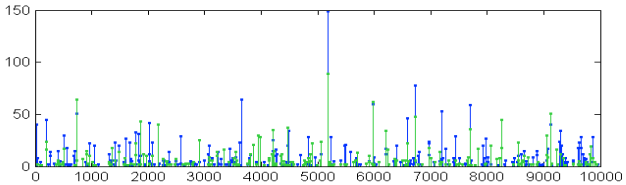
2. RELATED WORK

There has been an evolution in video retrieval when given a query. Raw search (without quantization) for matching features according to the nearest neighbor between descriptors was tested with good results for TRECVID instance search 2011 [2, 7]. But even though matching raw features saves training cost and time, the computation results very expensive as each descriptor is treated and matched independently from the rest. In order to speed up the search, bag-of-words implemented with inverted file indexing retrieved the best results and accuracy on 2011.

(a) raw matching of local SIFT features.



(b) bow matching with quantized local SIFT features.



(c) bow matching with quantized paired SIFT features.

(d) raw matching of paired SIFT features.

(a) raw feature matching	(d) raw pairs matching
(b) bow matching	(c) pairs' bow matching

Fig. 1: Chart of the different matching approaches. (b) and (c) show the Bag-of-Words histogram of the image on the right in blue, the one on the left in green, for local features and paired features, respectively.

As can be seen in figure 1, raw matching of points misses many semantic matches which could be quantized to the same word in bow matching, but it is more accurate and the spatial information of each match is available for further analysis. Of course, the match for raw features is much more expensive than searching for words in an indexed dataset.

BoW has proved to be effective and cheaper than matching raw features, as quantization allows assessing several points at a time, but loses the spatial information when creating the histogram. In the other hand, pairing points' descriptors according to spatial distance *gave* good results for dense point detection [8]. Classic visual vocabularies are generated from single local descriptors. However, these vocabularies are not able to capture the rich spatial contextual information among the local features. In fact, several works have verified that modeling these visual contexts combining two or more local features could greatly improve the performance of many visual matching and recognition algorithms [9, 10].

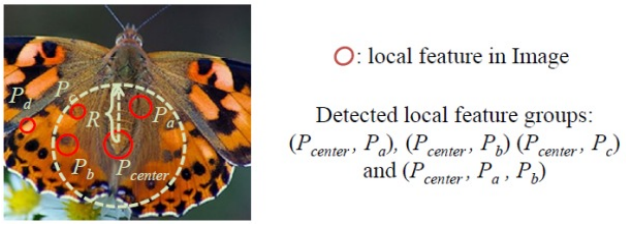


Fig. 2: At the centered local feature P_{center}, three groups containing two local features are detected. Furthermore, one group of three local features containing the two closest ones is also detected.

In [10], two or three local features are considered in each local feature group as shown in figure 2. Since if too many local features are combined, the repeatability of the combination will decrease. In addition, if more local features are contained in each group, there would be more possible feature-to-feature matches between two groups, making the repeatability even more difficult. Because of that, in our approach only pairs are combined. In figure 3 it can be noted how the descriptor's scale is important [10].

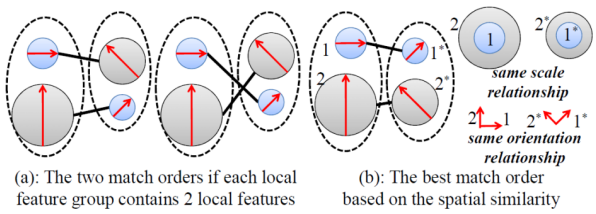


Fig. 3: When comparing spatially two groups of two local features, there are two possible matches (showed on the left). When 1st matching is considered, local matched features have the same orientation and scale relationships.

3. PROPOSAL

Due to the huge size of Instance Search databases, dense sampling makes memory requirements difficult to be achieved on this dataset. For that, and in order to slightly preserve the spatial information in BoWs, our proposal relies in combining the pairing procedure and the BoW techniques using sparse sampling:

- Key-points are selected using a sparse detector and color SIFT is then computed for each.
- For each key-point i, K nearest neighbors (so that it is scale invariant) are selected within a limited neighborhood.
- For each neighbor j, a new descriptor is generated, being the concatenation of the previous color SIFTs, that is a 384D descriptor.
- A 10.000 words vocabulary is created out of the new descriptors with k-means. The centers for the quantization clusters are therefore 384D.
- Bag-of-words model is applied with the quantized 384D descriptors.

4. LOCAL PAIRWISE CODEBOOK

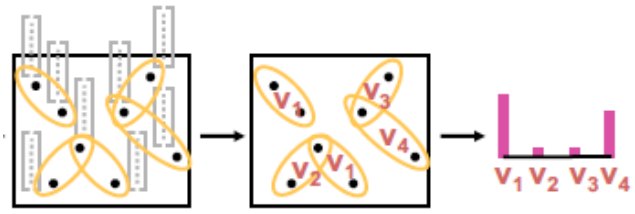


Fig. 4: An overview of our approach. (a) Features are paired locally and descriptors are concatenated. (c) A visual word is assigned to each local pair of features. (d) A histogram of pairwise visual words is established per image.

4.1. Pairing Spatially Close Feature Descriptors

Nobuyuki et al. [6] pair all descriptors densely detected within a distance γ to the point, and then concatenate them according to the first non-equal dimension of the descriptor. In our approach, each sparse key-point is paired to k-neighbor descriptors, this way the pairing is scale invariant.

As pairing sparse key-points may not guarantee its spatial proximity, a filtering is applied. If the key-point to pair is not inside a given neighborhood the pair is discarded. Each key-point's neighborhood is given by the descriptor's scale (ellipse defined by a,b,c), as it is the area for which the descriptor was computed. This is a first contribution of our work to the original proposal of Nobuyuki et al.

Figure 5 shows the differences in pairing using the scale information (right) or not (left) for k=20 neighbors. The color ellipses in the right show the neighborhood for some of the key-points (represented with blue circles). The red

crosses are the center position between the paired key-points. The wrong pairs in red can be appreciated on the left in the middle of the sky. On the right, those pairs are excluded, as the pairing points are not inside the neighborhood. This is also notorious inside the green neighborhood on the right: for the central point, only three key-points below can be paired, all pairs above the center disappear.

Fig. 5: In the image at left, original paired is shown. At right, pairs with scale filtering can be seen.

The new descriptor is a concatenation of both previous descriptors:

$$f_{(i,j)} = \left(\frac{(x_i + x_j)}{2}, \frac{(y_i + y_j)}{2}, [\mathbf{d}_i \mathbf{d}_j] \right) \quad (1)$$

Once paired the descriptors, a new codebook of 10000 visual words is generated with k-means using 1000000 paired descriptors.

4.2. Efficient Quantization of Pairwise Features

As concatenation can be done in two ways (ji and ij), quantization is also done by taking into account both different configurations: for each pair ij, both distances $dist_{ij}$ and $dist_{ji}$ are computed. $dist_{ij}$ is the distance of d_{ij} to the closest cluster center (descriptors d_i and d_j are concatenated in this order), whereas $dist_{ji}$ is computed concatenating the other way round, that is the distance of d_{ji} to the closest cluster. The smaller one resolves the quantization value.

Having a 2*k times bigger dataset, quantizing all pairs of points separately would be very expensive. This is why quantization is efficiently computed by employing the original local descriptors, as the new ones are nothing but a concatenation of those: since the vocabulary has been trained with concatenated descriptors, any visual word (cluster center) c_n can also be represented as a concatenation of two vectors $[c_{n1}\ c_{n2}]$. Given a descriptor $d_{ij}=[d_i d_j]$, its Euclidean distance to the cluster will be the sum of both distances $dist(d_i,c_{n1})$ and $dist(d_j,c_{n2})$. d_i and d_j will create several other pairs, so their distance to each cluster doesn't need to be computed again if it is stored in the matrix Q_1,Q_2, which will be of dimensions n x p, being p the number of local key-points and n the number of clusters:

$$\begin{aligned} d(d_{ij}, c_n) &= \| c_n - d_{ij} \|^2 = \| [c_{n1} c_{n2}] - [d_i d_j] \|^2 \\ &= \| c_{n1} - d_i \|^2 + \| c_{n2} - d_j \|^2 \\ &= Q^1_{n,i} + Q^2_{n.j} \end{aligned} \quad (2)$$

In order to assure the same quantized visual word for $d_{ij}=[d_i d_j]$ and $d'_{ji}=[d'_j d'_i]$ (being those very similar descriptors to the initial d_i, d_j), quantization will be defined by:

$$vw_{ij} = vw_{ji} = \min_n(min(d(d_{ij}, c_n)), min(d(d_{ji}, c_n))) \quad (3)$$

5. IMPLEMENTED SOLUTIONS

The current results are not satisfactory as expected. They are inconsistent, so further tests need to be done in order to find the bug.

6. REFERENCES

[1] Yimeng Zhang, Zhaoyin Jia, and Tsuhan Chen. Image retrieval with geometry-preserving visual phrases. In Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on, pages 809–816. IEEE, 2011.

[2] Wengang Zhou, Yijuan Lu, Houqiang Li, Yibing Song, and Qi Tian. Spatial coding for large scale partial-duplicate web image search. In Proceedings of the international conference on Multimedia, pages 511–520. ACM, 2010.

[3]. Lazebnik, S., Schmid, C., Ponce, J.: "A Maximum Entropy Framework for Part- Based Texture and Object Recognition". In: ICCV (2005)

[4]. Ling, H., Soatto, S.: "Proximity Distribution Kernels for Geometric Context in Category Recognition". In: ICCV (2007)

[5]. Liu, D., Hua, G., Viola, P., Chen, T.: "Integrated feature selection and higher-order spatial feature extraction for object categorization". In: CVPR (2008)

[6] Nobuyuki Morioka and Shin'ichi Satoh. Building compact local pairwise codebook with joint feature space clustering. In Computer Vision–ECCV 2010, pages 692–705. Springer, 2010. (document)

[7]. Boiman, O., Shechtman, E., Irani, M.: "In defence of Nearest-Neighbor based image classification". In: CVPR (2008)

[8] Tinne Tuytelaars. Dense interest points. In Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on, pages 2281–2288. IEEE, 2010.

[9] Zhong Wu, Qifa Ke, Michael Isard, and Jian Sun, "Bundling features for large scale partial-duplicate web image search", Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on, IEEE, 2009, pp. 25-32.

[10]. Shiliang Zhang, Qingming Huang, Gang Hua, Shuqiang Jiang, Wen Gao, and Qi Tian, “Building contextual visual vocabulary for large-scale image applications”, *Proceedings of the international conference on Multimedia*, ACM, 2010, pp. 501–510.