Subspace Clustering for Information Retrieval in Urban Scene Databases

Marcelo de M. Coelho*
Teaching Division
Prep. School of Air Cadets
(EPCAR)
Barbacena, MG, Brazil
Email: mcoelho@dcc.ufmg.br

Eduardo Valle
Institute of Computing
University of Campinas
(UNICAMP)
Campinas, SP, Brazil
Email: mail@eduardovalle.com

Cássio E. dos S. Júnior
Arnaldo de Albuquerque Araújo
*Department of Computer Science
Federal Univ. of Minas Gerais (UFMG)
Belo Horizonte, MG, Brazil
Email: {cass,arnaldo}@dcc.ufmg.br

Fig. 1. Improvement of image matching by using subspace clustering: the first line shows normal matching where the query (first left image) is matched with the 209^{th} (last image) returned database image. After our method, at the second line, query image (first left image) is recognized as the 2^{nd} (third image) returned database image.

***Abstract*—We present a comprehensive study of two important subspace clustering algorithms and their contribution to enhance results for the difficult task of matching images of the same object using different devices at different conditions. Our experiments were performed on two distinct databases containing urban scenes which were tested using state-of-the-art matching algorithms.**

Our start point was the hypothesis that low discriminant local point descriptors lead to misclassification, which can be reduced employing clustering techniques as filters. A significantly amelioration of the results obtained for the two tested databases was achieved, which indicates that subspace clustering techniques have much to contribute at this kind of application.

Another point is whether the occurrence of obstacles like trees and shadows are responsible for misclassification of images.

***Keywords*-Subspace Clustering; Information Retrieval; Large Databases; Urban Databases**

I. Introduction

This work is concerned with the improvement of visual local features in large databases by using unsupervised learning to remove low-quality samples. More specifically, we use subspace clustering techniques to separate discriminating local features from non-discriminating ones in the applicative context of visual recognition in urban databases [1].

Visual recognition is an important task in computer vision and artificial intelligence, which aims at the automatic identification of objects and scenes [2]. This has been an active area of research for the last few decades, with many important breakthroughs, notably the development of discriminant local features [3] and the availability of a powerful theory of statistical learning [4] with practical algorithms to perform classification tasks [5].

However, plenty of challenges remain, many of which related to the ability of the visual feature extractors to provide relevant information to the retrieval and classification engines. Valle [6] and Picard [7] identify the presence of many low discriminating local features as one of the main difficulties for highly accurate recognition of urban scenes. They perform an experiment to match images which are stored in a urban scene database of Paris. Both database and query images are described using Scale-Invariant Feature Transform (SIFT) [8].

Then, data set points nearby query points are found using the Multicurves [9] index and the matching of images is made by: a) a database image containing the greatest number of points which are the nearest from a query image is suggested as its most similar image. The subsequent images are arranged in decreasing order of the number of correspondent points and they are suggested as options for the correct match; and b) the previous matches are filtered by their geometric consistency, which is obtained with the RANSAC algorithm [10], and similar resulting images are ranked as it was seen at the item a). The advantages of the method are its robustness regarding techniques that use global descriptors, because there are many local descriptors, and its efficiency, since individual descriptor matching is performed independently, rather than pairwise image comparison. On the other hand, the multiplicity of descriptors penalizes the performance and matches made incorrectly might cause troubles [6].

A. Related work

The use of the Bag-of-Visual-Features (BoVF) [11] is another way to match or classify images and videos where each image is represented through the occurrence of its visual words. A multilevel BoVF, named Spatial Pyramid [12], has been used in order to reach better results in classification tasks. Its idea is to represent an image by a vector which includes computed histograms of its features over hierarchical divisions of such image. Then, by using Spatial Pooling, a training set is created and submitted to a supervised learning technique and used to classify a set of test images.

In [13], boosting in recognition of human actions is measured by incorporating knowledge about actor localization in the BoVF representation. In addition, they used randomly chosen points as centroids to the codebook instead of applying a clustering algorithm. By using this strategy, the cost is smaller than employing clustering techniques. The choice of how many random points shall be used as centroids is approached experimentally.

Selection of features is proposed in [14] as an additional step for BoVF in order to save memory. Useful features are detected by comparing each database image against the full feature set and applying geometric constraints as well as the concept of image adjacency.

An image representation for natural scenes using local semantic description is proposed in [15]. The main idea behind that work is to perform a semantic modeling in order to classify local image regions into semantic concepts such as water, sky, rocks and foliage. They achieve the best classification of image regions using SVM and nine concept classes.

Contributions: The goal of our work is to show that subspace clustering algorithms can be used to improve classification and recognition tasks, mainly those which are concerned with very large databases.

As practical application, we will explore the results obtained in [6] and [7]. Their authors claim that the presence of trees and shadows on the images causes mismatches between query and database images. Therefore, subspace clustering algorithms are employed as an attempt to separate points that are responsible for mismatches from the ones that are not. Then, a different urban data set is used in order to confirm the results.

In Section 2, two subspace clustering algorithms are discussed and evaluated. The experiments concerning the image matcher and its behavior after the data clustering are presented in Section 3. Work remarks, drawbacks and future work are posed in Section 4.

II. Subspace Clustering Algorithms

Clustering algorithms aim to group data according to a similarity measure function. Depending on the similarity function and the nature of the data, group formation can vary substantially. Nevertheless, traditional clustering algorithms, such as k-means, are not suitable to deal with high dimensional data. In that case, we can adopt subspace clustering algorithms, which analyze the similarity among data points at sets of specific dimensions rather than all possible ones. Those analyzed dimension sets, in turn, conceive subspaces.

Parsons, in [16], establishes a comprehensive study of subspace clustering algorithms and the Fast and Intelligent Subspace Clustering Algorithm using Dimension Voting (FINDIT) [17] is indicated as the most appropriated for large multidimensional sets. Besides, FINDIT algorithm reaches the best performance, it is a method that uses a voting scheme and it is able to explicitly define what are the key dimensions of each cluster found. Therefore, this algorithm was the first choice to be tested in this work.

The second algorithm analyzed here was proposed in [18] and employs fuzzy techniques to separate data in clusters. It is named Mean-Shift for Subspace Clustering (MSSC) and it works selecting random data points as estimated cluster centroids, which are iteratively refined until convergence.

In the following sections, both algorithms are detailed and compared through the use of a synthetic database which was generated according to [19] and parametrized in agreement with the tests made in [17]. More details about that synthetic database generation are given in Section III.

A. FINDIT

The FINDIT algorithm [17] promises to be fast and accurate for the discovery of clusters and their dimensions.

In order to achieve this performance, there are eight steps that FINDIT performs from the initial analysis of data to the final assignment of each database point to its cluster. It receives as initial parameters $C_{minsize}$, which is the minimum size of each cluster, and $D_{mindist}$, the maximum distance between two clusters that implies in their merging. These steps are:

1) Sampling – data points are sampled in two sets for the initial discovery of clusters: a sample set (S) and a *medoid* set (M). Through the use of the Chernoff bounds, it is granted that such samples are representative. Hence, this algorithm is able to perform clustering just using a sample from original database which means memory and execution time saving.

2) Dimension Voting – here, we need to select the key dimensions for each *medoid*. Therefore, to each *medoid* in M, the V nearest neighbors are sought in S. Those V neighbors are voters to determine the key dimensions and the distance among the points in V and every *medoid* is evaluated by:

$$dod_\epsilon = |D| - |\{d||m(d) - p(d)| \leq \epsilon,\ d \in D\}|,$$

where $p(d)$ is the dth-dimensional value of a point p, $|D|$ is the number of data dimensions and ϵ is the greatest distance accepted between each *medoid* and a point for a dimension d.
3) Member Assignment – every sampled point is assigned to a *medoid* by observing its key dimensions evaluated in the previous step.
4) *Medoid* Clustering – *medoids* or a set of *medoids* are merged if they have, at most, $D_{mindist}$ dimensions where their distance is higher than ϵ.
5) *Medoid* Cluster Tuning – the new key dimensions for *medoid* clusters are found.
6) Refinement – those clusters having few members are removed.
7) Evaluation – the quality of clusters is evaluated through the soundness criteria which means the summation of the key dimensions by the number of members of each cluster product. Steps two to seven are repeated with the variation of ϵ, so the best soundness is chosen and its clusters are used to the final assignment of data. ϵ variation is calculated taking in account data value range.
8) Data Assignment – every database point is assigned to a cluster in the set with the best soundness, observing key dimensions in order to perform that association.

An important remark of this algorithm is the selective use of the data that makes it suitable for a set with many points. The FINDIT algorithm was programmed in C and compared with the MSSC. Results of the test are shown at the beginning of Section III.

B. MSSC

The MSSC algorithm is an extension of the Maximum-Entropy Clustering (MEC) algorithm which is based on the physical concept of free energy in the course of an annealing process. Therefore, MEC aims to minimize the free energy during clustering and that is assimilated into the MSSC [18], in which the free energy is given by [18]:

$$F = -\left(\frac{1}{\beta}\right)\sum_{x\in X}\ln\left(\sum_{j=1}^{k} e^{-\beta\|x-z_j\|^2}\right),$$

where β is the Lagrange multiplier and is related to the quantity of clusters that are found after clustering process. That means, if β is zero, exactly one cluster will be found. As higher is the value of β, it is more likely that the number of clusters found is near to the real division of data.

Besides the β parameter, another asked parameter is α which is the fuzzy controller of the dimensional weights. In a different way, α has as range $(1, \infty)$ and it is responsible to isolate the subspace cluster dimensions during the algorithm iterations (it is used in Steps 2 and 4 below). Also, there is the *k* parameter which defines the maximum number of clusters to be found.

The operation of this algorithm can be summarized in few steps:

1) Initially, k points from the database are chosen randomly and are stored in a $Z_{k\times dimensions}$ matrix to be considered as the initial centroids; a $W_{k\times dimensions}$ matrix is created to keep the weights for each centroid dimension. This last matrix is initialized with (1/quantity of dimensions) for all cells.
2) A fuzzy matrix $U_{k\times N}$ is calculated by using Z and W, and it stores for every data set point its proximity likelihood to each centroid.
3) The Z matrix is updated using U and stored in Z'.
4) The W matrix is updated using U and Z'.
5) Steps two to four are repeated until $|Z - Z'| \leq \epsilon$, where ϵ can be, for instance, 10^{-5}. If iteration continues, Z receives Z' values.
6) Matrix Z is searched in order to find identical centroids, if any exists, and the quantity of clusters can be encountered.
7) Every point of the database is assigned to a found centroid.

Basically, the main concern about the MSSC algorithm is its execution time cost. Cost is high because this algorithm loads all data on the primary memory and the update of the matrices is very costly. Also, this algorithm was implemented using C language.

III. EXPERIMENTS

The experiments were based on the image search algorithm provided in [6] and [7] which aims to search for an image, taken by a mobile phone, in a urban scene database. Roughly speaking, the algorithm uses feature vectors extracted by the SIFT algorithm [8]:

1) Initially, it searches for interest points of the database, that are the nearest neighbors to the query points, using the Multicurves [9] index.
2) A simple image matching is realized, just counting the quantity of a database image interest points that are related to the points of each query image.
3) Database images are sorted by descending order in accord to the previous item counting, i.e., the database image containing the greatest amount of points similar to a specific image query is placed in the first similarity rank for that query and so on. Hence, every query image has assigned to it a list containing similar database images sorted from the most similar, lower rank, to the less similar, higher rank. Such matching method is referred in [6] as Brute Vote.

TABLE I
RESULTS OBTAINED WITH THE ORIGINAL DATABASE AND USED AS REFERENCE

Query Images	Original Database	
	Brute Vote	RANSAC
1	4	10
2	32	15
3	54	16
4	50	50
5	3	21
6	163	31
7	250	138
8	173	1
9	169	0
10	172	73
Rank	106.9	39.4
Improvement to Baseline	-	-

4) Step two is repeated using the RANSAC algorithm to inspect the geometric consistency among points.
5) Again, a ranked list for each query image containing the correspondent database images in ascending order of similarity is generated. This matching method is referred in [6] as RANSAC.

Two databases were analyzed in [6] and [7], although only the database with the worst reported results is employed in this work. This database contains 300 images and an amount of 3,476,087 feature vectors. In order to test the matching algorithm, a set of ten query images described by 101,480 feature vectors is used. Matching results contain a rank for each query, which expresses where is the first correct returned image to that query, when ground-truth is observed. Note that we aim, as individual rank for each query matching, a value near from one, i.e., the first correct image is on the beginning of the query similarity list. For that database and queries, the average rank of matching was 98.8 using Brute Vote and 34.4 using RANSAC ([6],[7]). However, the values to be achieved and enhanced were defined by the best results reached in our experiments, which can be viewed in Table I. A rank zero valued indicates that there is no corresponding answer to a specific image, as it happens for Image Query 9. In that case, the average rank is evaluated discarding this input.

Following, the FINDIT and MSSC algorithms are compared before their use with image matcher and we can see that both algorithms have similar behaviors when they are evaluated using a synthetic database.

A. FINDIT versus MSSC

In order to evaluate the two previous algorithms, a synthetic database is used. It is generated employing the established criteria found in [19] and the parameters defined in [17]. We obeyed several criteria found in literature to create synthetic database, among them: a) it has 100,000 points that are 20-dimensional and split in 5 clusters; b) data is generated with no outliers; c) minimum cluster size is set as 5,000 points and the number of points assigned to each cluster follows an exponential distribution, with a mean of 1; d) the number of average correlated dimensions is 7 (which follows a Poisson distribution); and e) range for distribution of points in clusters was [2,4] (which follows a Normal distribution). The final distribution of data is shown in Table II.

TABLE II
THE COMPOSITION OF THE SYNTHETIC DATABASE USED IN THE FIRST SET OF EXPERIMENTS

Cluster	Members	Dimensions
1	7505	3,7,8,9,19
2	8029	6,7,8,9,11,16,19
3	25311	2,6,7,8,12,16,19
4	11423	1,2,9,12,16,17,19
5	47732	1,2,8,9,19

For the FINDIT algorithm, the parameters $C_{minsize} = 5000$, $D_{mindist} = 0$, $\xi = 30$ and the quantity of voters as 20 were used. Clusters found can be viewed in Figure 2a and the misclassification matrix in Figure 2c.

In Figures 2a and 2c, it is possible to perceive that FINDIT algorithm classifies some points as outliers and we can find an error in the classification of 1.852% for cluster 1, 0.005% for cluster 2 and 0.001% for cluster 4.

MSSC, in turn, does not detect outliers and its performance can be accomplished through Figure 2b, which presents the clusters encountered, and its mismatch matrix, in Figure 2d. The parameters used were $\alpha = 4.1$, $\beta = 43.9364$ and $k = 10$.

The error in the MSSC classification is negligible, i.e., it is less than 0.001% in clusters 1 and 4. However, its execution time, for a database containing millions of points is degraded rapidly. In that case, the FINDIT algorithm is indicated to be used. Another remark of both is the correct identification of cluster dimensions.

In the next subsection, subspace clustering algorithms are incorporated to the match algorithm used in [6] and [7]. After, the gain with the use of clustering is analyzed in the results.

B. Subspace Clustering before Matching Task

The basic goal of this work is to investigate the benefits of subspace clustering on recognition tasks, which is done by trying to enhance the rank obtained from the original articles, given by the first relevant image retrieved (averaged over the query set). Therefore, one wants to achieve lower ranks to the matches, as it was explained before, and subspace clustering algorithms are used on database in an attempting to clean it.

The MSSC is used with the parameters $\alpha = 3.1$, $\beta = 188.2861$ and $k = 25$. An amount of 24 clusters were found and they were submitted to the search algorithm, where the average quantity of members in clusters was 144,836 points. After each cluster found had been submitted to the algorithm, the best result was achieved for the cluster 23, which has 37,129 points, i.e., 1.06% of original points (Table III).

The original database was also clustered by the FINDIT algorithm and the quantity of clusters found for specific parameter variations was higher than 200. For $C_{minsize} = 5000$ and $D_{mindist} = 26$, which impose that each cluster has at least 5,000 points and we can ignore up to 26 dimensions

FINDIT

Cluster	Members	Dimensions
1	7366	3,7,8,9,19
2	7986	6,7,8,9,11,16,19
3	25311	2,6,7,8,12,16,19
4	11411	1,2,9,12,16,17,19
5	47864	1,2,8,9,19
Outliers	62	-

(a)

MSSC

Cluster	Members	Dimensions
1	47733	1,2,8,9,19
2	25311	2,6,7,8,12,16,19
3	11423	1,2,9,12,16,17,19
4	8029	6,7,8,9,11,16,19
5	7504	3,7,8,9,19
Outliers	-	-

(b)

FINDIT clusters

	1	2	3	4	5	Outliers
1	7366	0	0	0	132	7
2	0	7986	0	0	0	43
3	0	0	25311	0	0	0
4	0	0	0	11411	0	12
5	0	0	0	0	47732	0

(c)

MSSC clusters

	5	4	2	3	1	Outliers
1	7502	0	0	0	3	-
2	0	8029	0	0	0	-
3	0	0	25311	0	0	-
4	0	0	0	11423	0	-
5	2	0	0	0	47730	-

(d)

Fig. 2. Results of the first set of experiments on synthetic data. We can see Clusters and Misclassification matrix for FINDIT (a and c) and for MSSC (b and d). Both methods had detected correctly the intrinsic dimensionality, while FINDIT had found non-existent outliers, MSSC had achieved lower general misclassification.

TABLE III
IMAGE MATCHING IMPROVEMENT OBTAINED THROUGH THE USE OF CLUSTER 23 GENERATED BY MSSC

Query Images	MSSC $\alpha = 3.1$ and $\beta = 188.2861$ Cluster 23	
	Brute Vote	RANSAC
1	16	23
2	43	11
3	18	8
4	90	17
5	4	8
6	35	21
7	252	0
8	221	91
9	22	14
10	88	38
Rank	78.9	25.7
Improvement to Baseline	26.19%	34.93%

TABLE IV
IMAGE MATCHING IMPROVEMENT OBTAINED THROUGH THE USE OF CLUSTER 232 GENERATED BY FINDIT

Query Images	FINDIT $C_{minsize} = 1000$ and $D_{mindist} = 13$ Cluster 232	
	Brute Vote	RANSAC
1	48	21
2	29	5
3	14	5
4	53	54
5	24	11
6	1	4
7	130	63
8	24	12
9	41	16
10	24	22
Rank	38.8	21.3
Improvement to Baseline	63.70%	40.00%

when evaluating point distances, three clusters and 14.48% of outliers were found. However, the search results were not improved to any cluster, what can be explained for the $D_{mindist}$ value, i.e., if one permits 26 dimensions to be ignored for the distance evaluation, it is likely to happen the mixture of descriptors from interest regions with those regions that cause the confusion of the matching.

Another attempt using $C_{minsize} = 1000$ and $D_{mindist} = 13$ was done. At this time, 256 clusters and 52.71% of outliers were found. The matching result was analyzed for every cluster and the best result was evaluated for the cluster 232, which has 6,214 points, i.e., 0.18% of original database (Table IV).

By inspecting the results, we can see that it is possible to find at least one cluster that may ameliorate the rank, both in MSSC and FINDIT clusters. Unfortunately, it is not achieved a rank lower than 10.0, what can be explained if the points that cause the mismatches are not easily separable.

In other words, it seems that tree and shadow points absolutely confuse the matching task, but they are not the only ones. In order to investigate this hypothesis, the query vectors were clustered by the MSSC. The cluster 0 was used to match the whole database and returned good results (Table V). As in the query images there are neither trees nor complex shadows we can infer the existence of other points which cause mismatches.

In Tables III and IV, we can see significant improvement of the rank using MSSC and FINDIT on the original database.

In the course of this work, some researches that have employed dimensionality reduction as preprocessing step to perform the matching task were seen in the literature. One of them was The Locality Preserving Projections (LPP) [20] algorithm which has as advantage the neighborhood preservation. Hence, it was applied to the database, before its clustering, as

TABLE V
IMAGE MATCHING IMPROVEMENT OBTAINED THROUGH THE USE OF CLUSTER 0 GENERATED BY MSSC OVER QUERY DESCRIPTORS

Query Images	MSSC upon queries Cluster 0	
	Brute Vote	RANSAC
1	4	13
2	26	16
3	46	57
4	51	13
5	1	9
6	144	11
7	228	49
8	165	3
9	148	0
10	174	0
Rank	98.7	21.38
Improvement to Baseline	7.67%	36.79%

TABLE VI
COMPARISON BETWEEN IMAGE MATCHING FOR ORIGINAL DATABASE AND SVM CLEANED UP DATABASE USING RESIZED QUERY IMAGES

Resized Query Images	Original Database		MSSC and SVM Cluster 0	
	B. Vote	RANSAC	B. Vote	RANSAC
1	1	1	1	2
2	19	6	17	10
3	6	3	16	17
4	25	26	30	7
5	10	9	8	4
6	97	40	82	41
7	176	108	194	0
8	1	1	10	1
9	210	0	181	0
10	176	0	144	0
Rank	72.1	24.3	68.3	11.7
Improvement	-	-	5.27%	51.69%

we can see in the next subsection.

C. Dimensionality Reduction

As an additional attempt to the discovery of points which may contribute to the enhancement of the image matching, it was sampled 10% of the original database and applied the LPP algorithm to it. After that, the MSSC was employed and two clusters were found, used to train the Support Vector Machine (SVM) classifier and the resulting model was employed to classify the entire database.

However, as the results were not good, a detailed study of SIFT descriptor [8] was done and showed that scale difference among query and database images might difficult the matching task. Hence original query images were resized and the experiments were repeated for the original database and the cleaned database after SVM classification.

In Table VI, we can verify the influence of object sizes for the image matching.

Despite of the rank improvement encountered in the results after dimensionality reduction (Table VI) we can notice an undesirable effect: the presence of query images which were not classified, i.e., their rank is denoted as zero. Thus, we will compare the improvement achieved in the previous results with those obtained through the employment of a novel database, however the dimensionality reduction will not be used.

D. Another Urban Scene Database

A new database was gathered in Ouro Preto[1], a Brazilian historical city.

For the new database 618 images were employed and, in order to generate the query image set, 38 images were captured by a different device. After described by SIFT [8], those sets presented 1,839,545 and 747,250 feature vectors, respectively. The latter set was divided in three groups of queries and those four image group were matched against complete database and every cluster found. Results were compared and we have verified that clustering indeed improves image matching. It is noteworthy that the ground-truth for the complete query set makes reference to 83 database images, i.e., 13.43% of the original data set and each subset of the original query refers respectively to 74, 52 and 83 database image.

In Table VII, we can see the original classification of all the 38 query images where four of them were not classified.

After applying FINDIT subspace clustering algorithm to this database and matching each cluster found against image queries, the best improvement for 150 clusters was given by the 37^{th} cluster, as we can see in Table VIII.

Finally, we got some cues about the clusters which improve the image matching by analyzing this second set of experiments. For the clusters found by FINDIT we observed the presence of a great number of key dimensions and members around 3,000. We must remember that key dimensions are those dimensions which characterize the relationship between each point and its cluster, introduced by the FINDIT algorithm [17].

Again, we clustered the database by the MSSC algorithm in order to compare its results with those generated by FINDIT. Since we have incorporated the key dimensions concept to MSSC, we can observe that the cluster which presented the best improvement to the matches contains a lower number of key dimensions (about two or three), differently of FINDIT. However, the quantity of members continues around 3,000. Hence, an improvement of 26.72% was achieved with the use of cluster 26 (Table VIII).

After, Table VII was divided in three distinct queries in order to evaluate clusters behavior. The first one, containing 12 images, had all images ranked by the classifier. The last two query sets contain 13 images and they present three and two misclassified images, respectively. For all query subdivision, we achieved significantly improvement of matching.

For the first group, using FINDIT clusters, the cluster 88 improved in 61.52% the average rank. In respect to the other two groups we experienced an interesting effect. Other than the enhancement in the average rank for both groups, we observed that the clustering solved the problem of *non-encountered*

[1]Places in Ouro Preto: CM = Nossa Senhora do Carmo Church, SF = São Francisco de Assis Church, IN = Museum of the Inconfidência and TR = Tiradentes Square.

TABLE VII
INITIAL RANK EVALUATION FOR THE OURO PRETO DATABASE USING 38 QUERY IMAGES

Query Images	Original Database from Ouro Preto	
	Brute Vote	RANSAC
CM 001	369	102
CM 002	363	46
CM 003	318	82
CM 004	1	1
CM 005	291	88
CM 006	324	84
CM 007	145	1
SF 001	68	88
SF 002	18	0
SF 003	67	18
SF 004	82	74
SF 005	43	32
SF 006	25	11
SF 007	46	45
SF 008	33	76
SF 009	59	2
SF 010	112	0
SF 011	31	0
IN 001	131	81
IN 002	160	51
IN 003	140	43
IN 004	208	8
IN 005	130	9
IN 006	170	6
IN 007	40	30
IN 008	20	1
IN 009	152	4
IN 010	129	13
TR 001	241	4
TR 002	187	2
TR 003	263	0
TR 004	277	14
TR 005	14	6
TR 006	8	25
TR 007	403	195
TR 008	1	23
TR 009	2	18
TR 010	313	1
Rank	141.68	37.76
Improvement to Baseline	–	–

TABLE VIII
COMPARISON OF RANK IMPROVEMENT FOR THE OURO PRETO DATABASE USING CLUSTERS FOUND BY FINDIT AND MSSC

Image Queries	FINDIT $C_{minsize} = 1000$ and $D_{mindist} = 13$ Best Cluster		MSSC $\alpha = 3.1$ and $\beta = 250$ Best Cluster	
	Brute Vote	RANSAC	Brute Vote	RANSAC
CM 001	52	12	142	37
CM 002	95	24	330	36
CM 003	68	12	246	32
CM 004	44	18	41	10
CM 005	112	24	293	36
CM 006	74	13	276	37
CM 007	66	22	91	25
SF 001	23	42	86	28
SF 002	3	3	138	0
SF 003	28	28	99	26
SF 004	26	15	43	28
SF 005	22	11	20	46
SF 006	29	14	15	9
SF 007	23	15	24	18
SF 008	30	24	20	14
SF 009	23	43	59	27
SF 010	21	31	76	35
SF 011	10	8	139	41
IN 001	11	3	49	12
IN 002	6	4	44	17
IN 003	7	3	43	10
IN 004	6	7	58	10
IN 005	8	4	44	12
IN 006	9	3	39	22
IN 007	84	70	108	15
IN 008	8	3	50	13
IN 009	7	4	46	12
IN 010	8	2	47	13
TR 001	10	3	265	62
TR 002	8	4	233	43
TR 003	144	138	223	46
TR 004	21	10	259	91
TR 005	98	104	61	91
TR 006	14	23	1	1
TR 007	175	125	244	36
TR 008	41	31	1	3
TR 009	36	20	1	2
TR 010	118	66	187	28
Rank	41.26	25.95	108.97	27.68
Improvement	70.88%	31.29%	23.09%	26.72%

images, i.e., those images ranked previously as zero were correctly matched in the cluster. An improvement of 60.85% was obtained for the second subset when it was used cluster 73 to match images. For the third group, cluster 37 was responsible for 25.24% of gain in image matching.

Again, we used clusters generated by MSSC algorithm in order to compare its results with those generated by FINDIT. For the first query set which has 12 images, a gain higher than that obtained by FINDIT was observed. However, for the other two query subsets, results of MSSC were inferior, if compared with FINDIT clusters. Results can be viewed in Table IX.

As can be seen in Figure 1, the presented method improves the image matching task by classifying individual image in a more precise way and using just a small sample of the original data.

TABLE IX
RANK COMPARISON AMONG THREE SUBSETS OF THE ORIGINAL QUERY SET FOR THE RANSAC MATCHING

		FINDIT		MSSC	
Subset	Members	Rank	Improvement	Rank	Improvement
1	12	21.17	61.52%	20.83	62.12%
2	13	22.00	60.85%	30.00	46.62%
3	13	23.92	25.94%	30.58	4.44%

IV. DISCUSSION

We can observe that, in both tested databases, the employment of subspace clustering algorithms cause a significantly improvement of average rank, if compared with the results obtained without clustering. For the Paris database we experienced the amelioration on the matching task with rates up to 40.00%. Regarding the second data set, we reached about 62.00% of enhancement on query subsets and about 32.00% on the complete query. However, it is not clear which features collaborate to the correct match and which ones do not. While in the first database we have the occurrence of shadows and trees as obstacles to a good image matching, on the Ouro Preto data set the confusion may be caused by the architectonic style which is the same for the various buildings present on the images. The best results were reached through exhaustive inspection of clusters and those results should reinforce that

the task is indeed a challenging one.

Although we did not use dimensionality reduction on the second database experiments due the effect of *non-encountered* images, it is another way to separate database points and has the advantage of accelerating the clustering process. It is noteworthy that the best results obtained were those where the queries were resized in order to approximately match the database. This stresses that even when scale-invariant features, like SIFT, are employed, there are limits on how much this invariance can be stretched.

Moreover, when applying subspace clustering algorithms to databases we experienced an interesting effect inside SIFT space, since some clusters containing selected samples of data performed matching tasks better than whole database.

An interesting future step would be the identification of most discriminant clusters, i.e., which ones that enhance the image matching, without the use of exhaustive inspection of them. The employment of alternative matching algorithms such as Hough Transform, as referred in [8], is also a possibility, as well the selection of useful descriptors [14].

Besides, the MSSC algorithm can be easily enhanced in respect to time consuming performance. That can be done by incorporating sampling techniques used in FINDIT. Thus, our priority is to implement such modifications and to analyze the new MSSC actions.

V. Acknowledgements

The authors are grateful to CNPq, CAPES, FAPEMIG, and FAPESP, Brazilian research funding agencies, for the financial support to this work. Also we are thankful to Professor Alexandre Leão from Fine Arts Faculty of UFMG, who has taken the pictures used as query images, in the Ouro Preto database.

References

[1] Nokia, "Nokia challenge (2009/2010): Where was this photo taken, and how?" in http://comminfo.rutgers.edu/conferences/mmchallenge/2010/02/10/nokia-challenge/, 2009.

[2] J. Philbin, O. Chum, M. Isard, J. Sivic, and A. Zisserman, "Object retrieval with large vocabularies and fast spatial matching," in Computer Vision and Pattern Recognition, 2007. CVPR '07. IEEE Conference on, vol. 0. Los Alamitos, CA, USA: IEEE Computer Society, 2007, pp. 1–8.

[3] T. Tuytelaars and K. Mikolajczyk, "Local invariant feature detectors: a survey," Found. Trends. Comput. Graph. Vis., vol. 3, pp. 177–280, July 2008. [Online]. Available: http://portal.acm.org/citation.cfm?id=1391081.1391082

[4] V. N. Vapnik, Statistical Learning Theory. Wiley-Interscience, 1989.

[5] C. Cortes and V. Vapnik, "Support-vector networks," Mach. Learn., vol. 20, pp. 273–297, September 1995. [Online]. Available: http://portal.acm.org/citation.cfm?id=218919.218929

[6] E. Valle, D. Picard, and M. Cord, "Geometric consistency checking for local-descriptor based document retrieval," in Proceedings of the 9th ACM symposium on Document engineering, ser. DocEng '09. New York, NY, USA: ACM, 2009, pp. 135–138. [Online]. Available: http://doi.acm.org/10.1145/1600193.1600224

[7] D. Picard, M. Cord, and E. Valle, "Study of sift descriptors for image matching based localization in urban street view context," in Proccedings of City Models, Roads and Traffic ISPRS Workshop, ser. CMRT '09, 2009, pp. 193–198.

[8] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," Int. J. Comput. Vision, vol. 60, pp. 91–110, November 2004. [Online]. Available: http://portal.acm.org/citation.cfm?id=993451.996342

[9] E. Valle, M. Cord, and S. Philipp-Foliguet, "High-dimensional descriptor indexing for large multimedia databases," in Proceeding of the 17th ACM conference on Information and knowledge management, ser. CIKM '08. New York, NY, USA: ACM, 2008, pp. 739–748. [Online]. Available: http://doi.acm.org/10.1145/1458082.1458181

[10] M. A. Fischler and R. C. Bolles, "Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography," Commun. ACM, vol. 24, pp. 381–395, June 1981. [Online]. Available: http://doi.acm.org/10.1145/358669.358692

[11] J. Sivic and A. Zisserman, "Video google: A text retrieval approach to object matching in videos," in Proceedings of the Ninth IEEE International Conference on Computer Vision - Volume 2, ser. ICCV '03. Washington, DC, USA: IEEE Computer Society, 2003, pp. 1470–. [Online]. Available: http://portal.acm.org/citation.cfm?id=946247.946751

[12] S. Lazebnik, C. Schmid, and J. Ponce, "Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories," in Proceedings of the 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition - Volume 2, ser. CVPR '06. Washington, DC, USA: IEEE Computer Society, 2006, pp. 2169–2178. [Online]. Available: http://dx.doi.org/10.1109/CVPR.2006.68

[13] A. Kläser, M. Marszałek, I. Laptev, and C. Schmid, "Will person detection help bag-of-features action recognition?" INRIA Grenoble - Rhône-Alpes, 655, avenue de lEurope, 38334 Montbonnot Saint Ismier, FRANCE, Tech. Rep. RR-7373, sep 2010. [Online]. Available: http://lear.inrialpes.fr/pubs/2010/KMLS10

[14] P. Turcot and D. Lowe, "Better matching with fewer features: The selection of useful features in large database recognition problems," in Computer Vision Workshops (ICCV Workshops), 2009 IEEE 12th International Conference on, 27 2009-oct. 4 2009, pp. 2109–2116.

[15] J. Vogel and B. Schiele, "Semantic modeling of natural scenes for content-based image retrieval," International Journal of Computer Vision, vol. 72, pp. 133–157, 2007, 10.1007/s11263-006-8614-1. [Online]. Available: http://dx.doi.org/10.1007/s11263-006-8614-1

[16] L. Parsons, E. Haque, and H. Liu, "Subspace clustering for high dimensional data: a review," SIGKDD Explor. Newsl., vol. 6, pp. 90–105, June 2004. [Online]. Available: http://doi.acm.org/10.1145/1007730.1007731

[17] K.-G. Woo, J.-H. Lee, M.-H. Kim, and Y.-J. Lee, "Findit: a fast and intelligent subspace clustering algorithm using dimension voting," Information and Software Technology, vol. 46, no. 4, pp. 255–271, 2004. [Online]. Available: http://www.sciencedirect.com/science/ARTICLE/B6V0B-49HMWRS-1/2/5374bfa16fe12aeb7c31b64d7c18ba20

[18] G. Gan, C. Ma, and J. Wu, Data Clustering: Theory, Algorithms, and Applications, illustrated edition ed. Philadephia, PA: Society for Industrial and Applied Mathematics, May 2007. [Online]. Available: http://link.aip.org/link/doi/10.1137/1.9780898718348

[19] C. C. Aggarwal, J. L. Wolf, P. S. Yu, C. Procopiuc, and J. S. Park, "Fast algorithms for projected clustering," in Proceedings of the 1999 ACM SIGMOD international conference on Management of data, ser. SIGMOD '99. New York, NY, USA: ACM, 1999, pp. 61–72. [Online]. Available: http://doi.acm.org/10.1145/304182.304188

[20] X. He and P. Niyogi, "Locality preserving projections," Advances in Neural Information Processing Systems 16, vol. 16, no. December, pp. 153–160, 2003. [Online]. Available: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.4.5576&rep=rep1&type=pdf