UAIC: Participation in ImageCLEF 2009 Robot Vision Task

Emanuela Boroş, George Roşca, Adrian Iftene

UAIC: Faculty of Computer Science, "Alexandru Ioan Cuza" University, Romania
{emanuela.boros, george.rosca, adiftene}@info.uaic.ro

Abstract. This year marked our first participation at the Robot Vision task a new task from ImageCLEF competition. The paper represents a brief description of our system as the solution to the problem of topological localization of a mobile robot using visual information. We were asked to determine the topological location of a robot based on images acquired with a perspective camera mounted on a robot platform. And so, we decided that we don't need an incremental learning system and we approached a statistical method that always will work the best results. We used to apply a feature-based method (SIFT[1]) and two main systems in order to search and classify the given images written by us. At the same time, the systems preserve the recognition performance of the batch algorithm. The first system is reordering the images so we can get the most important/representative images for a room's category. This is done using SIFT. The second system is a brute one, just for testing the differences between this one and the first one, not selecting the representative images. We acquired a separation in directories for every room capturing the key points saved in files for every image (or representative) from every room. About the changes in the environment, the SIFT algorithm occupies itself. The entire recognition system consists in a server – client architecture. Server is processing one single search at a time, and after the search ends connection with the client closes. The resulting file is the asked-for file with the final results for the batched images.

1 Introduction

ImageCLEF[2] hosted in 2009 for the first time a Robot Vision task. The task addresses the problem of topological localization of a mobile robot using visual information. Specifically, we/participants were asked to determine the topological location of a robot based on images acquired with a perspective camera mounted on a robot platform.

We received training data consisting of an image sequence recorded in a five room subsection of an indoor environment under fixed illumination conditions and at a given time. We had to build a system able to answer the question "*where are you*?" (with possible answers "*I'm in the kitchen, in the corridor*", etc.) when presented with

[1] SIFT: Scale Invariant Feature Transform
[2] ImageCLEF: http://www.imageclef.org/

a test sequence containing images acquired in the previously observed part of the environment or in additional rooms that were not imaged in the training sequence. The test images were acquired 6-20 months later after the training sequence, possibly under different illumination settings. The system should assign each test image to one of the rooms that were present in the training sequence or indicate that the image comes from a room that was not included during training. Moreover, the system can refrain from making a decision (e.g. in the case of lack of confidence).

The algorithm must be able to provide information about the location of the robot separately for each test image (e.g. when only some of the images from the test sequences are available or the sequences are scrambled). This corresponds to the problem of global topological localization. However, results can also be reported for the case when the algorithm is allowed to exploit continuity of the sequences and rely on the test images acquired before the classified image.

We started with the release of annotated training and validation data. Training and validation were performed on a subset of the publicly available IDOL2 Database [1]. The database contains image sequences acquired in a five room subsection of an office environment, under three different illumination settings and over a time frame of 6 months. The test sequences were acquired in the same environment, 20 months after the training data, and contain additional rooms that were not imaged previously.

The rest of the paper is organized as follows: after a review of previous literature in the field (Section 2), we describe our system (UAIC System) (Section 3). Section 4 describes the experiments and then Section 5 presents the results. Section 6 draws conclusions regarding our participation in Robot Vision task.

2 Related Works

The research on mainly object recognition has been increasing mostly in the mobile robotics community. In this project, in the first iteration is essentially the Sift algorithm [3] by David Lowe implemented. Therefore, the papers [1], [3] (and by Lowe and referenced), and reference implementations were studied. The main objective of this paper represents, as mentioned before, a method for extracting distinctive invariant features from images that can be used to perform reliable matching between different views of an object or scene. The features have to be invariant to a lot of changes that image must suffer.

In computer vision and image processing, these insights have lead to the construction of multi-scale representations of image data, obtained by embedding any given signal into a one-parameter family of derived signals (Burt 1981; Crowley 1981; Witkin 1983; Koenderink 1984; Yuille and Poggio 1986; Florack et al. 1992; Lindeberg 1994d; Haar Romeny 1994).

3 UAIC System

Module details for our system of dynamical place recognition are presented in the sections below.

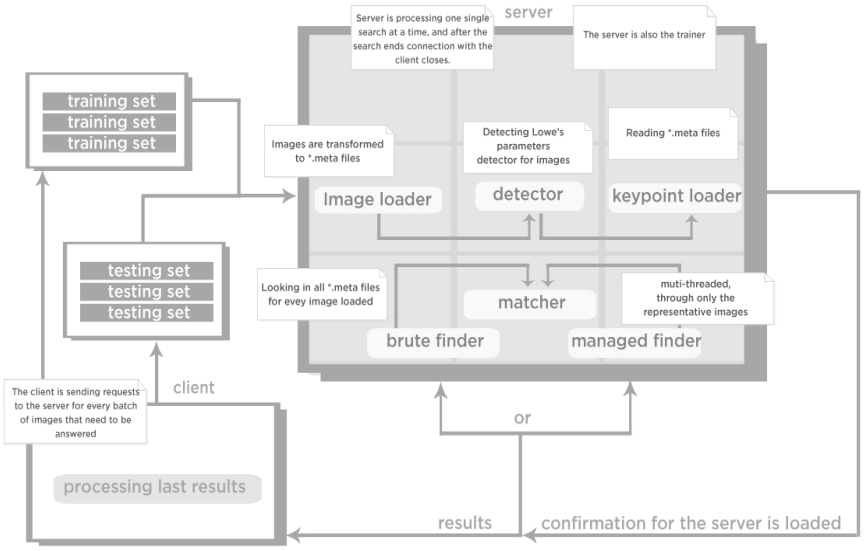

Figure 1: UAIC system used in Robot Vision task

The architecture of the system is similar to server-client architecture, and it is possible to accomplish more requests at a time. This change has brought performance improvement.

The vision of the project is a source of information all about local points of interest via the above channels to offer. The purpose of information and visualized objects is to organize and communicate valuable data to people, so they can derive increased knowledge that guides their thinking and behavior. Human learning processes mediate the building of knowledge from meaningful information and integration into one's knowledge base. Therefore, a proper understanding of human learning is important to consider while making any decision. Our need in imitating the human capability of learning has become the main purpose of science. And so, the problem proposed to solve is the problem of topological localization of a mobile robot using visual information acquired with a perspective camera mounted on a robot platform. We didn't choose a mechanism of incremental learning, we chose a statistical one as the people learn through observation, trial-and-error and experiment. As we now, learning happens during interaction. We managed the images (“interactions with objects” for human learning) so they become a mini-system of storing features of them.

This paper, as it has been said before, presents an algorithm able to recognize places on the basis of images' features, under possible variations between matching image pairs. Translations, rotations, scales and luminance changes can cause the difference of two pictures. It is virtually impossible to compare two images using

traditional methods such as a direct comparison between gray values as it could be really simple with an existing API (Java Advanced Imaging API[3]).

In addition, this paper presents a method for extracting distinctive invariant features from images that can be used to perform reliable matching between different views of an object or scene. The features have to be invariant to a lot of changes that image must suffer.

The goal of the iteration is a clear breakdown of the project objectives into individual fragments, which all run independently. We have decided the timing Features - boxed, and the key aspects in the beginning to be realized. Each iteration ends with a functional prototype.

3.1 The Server Module

This module has two parts: one part necessary for training and one part necessary for classifying. The trainer supports both single images and directories. The images must be annotated in the given format. We did not use an existing tool, but our code (Java), based mostly on finding the keypoints (points of interest) of an image. The preparation for the trainer is scaling down the images, plus the well-known SIFT algorithm applied to them.

3.1.1 Trainer component

The trainer supports both single images and directories and it is detecting and describing local features in images. The 'scale-invariant feature transform' features are local and based on the appearance of the object at particular interest points, and are invariant to image scale and rotation. In the process of training, all the chosen images will get through the key point localization processor, obtaining the keys' files. These files are included in testing the application.

In the process of training, all the chosen images will get through the key point localization processor, obtaining the keys' files. These files are included in testing the application. Additional to brute force in which all pictures are considered in training process, we have added a "get representative images" method. In this way, the trainer obtains for a single room that initially has 400 images, just 25 images. After that we have trained our application twice (that took us 2-4 days): one for all pictures and one for representative images. The brute trainer is analyzing all the images. More details are under SIFT algorithm subsection.

3.1.2 SIFT Algorithm

In the first iteration is essentially the SIFT algorithm [] implemented by Lowe David. Therefore, the papers (and by Lowe and referenced), and reference implementations were studied. Where no precise definition in the paper is available, values must be determined empirically. Finally, the stability and efficiency of the implementation

[3] Java Advanced Imaging API (JAI):
http://java.sun.com/javase/technologies/desktop/media/

will be tested. In addition, a review will take place whether the algorithm for the detection of three dimensional objects is appropriate. SIFT - Scale Invariant Feature Transform is a feature-based image matching approach, which lessens the damaging effects of image transformations to a certain extent. Features extracted by SIFT are invariant to image scaling and rotation, and partially invariant to photometric changes.

Four stages throughout the computation procedure as follows:

1. *Scale-space extrema detection*: first, use difference-of-Gaussian (DOG) to approximate Laplacian-of-Gaussian and build the image pyramid in scale space. Determine the keypoint candidates by local extrema detection. This is done by comparing each pixel in the DoG images to its eight neighbors at the same scale and nine corresponding neighboring pixels in each of the neighboring scales. If the pixel value is the maximum or minimum among all compared pixels, it is selected as a candidate keypoint. Scale-space extrema detection produces too many keypoints candidates, some of which are unstable.
2. *Strip unstable keypoints*: use the Taylor expansion of the scale-space function to reject those points that are not distinctive enough or are unsatisfactorily located near the edge.
3. *Feature description*: Local image gradients and orientations are computed around keypoints. A set of orientation, scale and location for each keypoint is used to represent it, which is significantly invariant to image transformations and luminance changes.
4. *Feature matching*: compute the feature descriptors in the target image in advance and store all the features in a shape-indexing feature database. To initiate the matching process for the new image, repeat steps 1-3 above and search for the most similar features in the database.

First of all, the images are mapped and then they are scaled up at double of their size. After we assume that the image has a blur of at least 0.5 so an initial image smoothing is passed. This is the stage where the interest points, which are called keypoints in the SIFT framework, are detected. For this, the image is convolved with Gaussian filters at different scales, and then the differences of successive Gaussian-blurred images are taken. Keypoints are then taken as maxima/minima of the Difference of Gaussians (DoG) that occur at multiple scales. For scale-space extrema detection, the image is first convolved with Gaussian-blurs at different scales. The value of octave sigma detection parameter is 2.0, how was suggested by Lowe's research paper.

The convolved images are grouped by octave (the largest possible number of octaves), each holding levels per octave = 3 scales in scale-space. Each octave is downscaled by 0.5 and the scales in each octave represent a sigma change of to 2.0 * sigma octave. Then the Difference-of-Gaussian images are taken from adjacent Gaussian-blurred images per octave. Once DoG images have been obtained, keypoints are identified as local minima/maxima of the DoG images across scales. This is done by comparing each pixel in the DoG images to its eight neighbors at the same scale and nine corresponding neighboring pixels in each of the neighboring scales. If the pixel value is the maximum or minimum (in the images maps) among all compared pixels, it is selected as a candidate keypoint. Maxima and minima of the difference-of-Gaussian images are detected by comparing a pixel to its 26 neighbors

in 3x3 regions at the current and adjacent scales (marked with circles). From Lowe's paper [1] it's not really clear whether we always need 3 neighborhoods spaces (3x3 regions) or should also search only one or two spaces.

This keypoint detection step is a variation of one of the blob detection methods developed by Lindeberg by detecting scale-space extrema of the scale normalized [2]. The difference of Gaussians operator can be seen as an approximation to the Laplacian, [5] here expressed in a pyramid setting. Scale-space extrema detection produces too many keypoint candidates, some of which are unstable. The next step in the algorithm is to perform a detailed fit to the nearby data for accurate location, scale, and ratio of principal curvatures [4]. This information allows points to be rejected that have low contrast (and are therefore sensitive to noise) or are poorly localized along an edge. After the edge filtering, each keypoint is assigned one or more orientations based on local image gradient directions. This is the key step in achieving invariance to rotation as the keypoint descriptor can be represented relative to this orientation and therefore achieves invariance to image rotation. All keypoints are written to files that represent the database for the server.

3.1.3 Classifier component

In this iteration, the database for the management of point of interests will be created. The SIFT algorithm is designed is used to browse the database. It is also on the efficiency respected. The training data is from IDOL Database [6]. Access to the database is done by the server. The server loads once with all the keypoints files and waits for requests.

The brute finder/trainer is one type of classifier as it creates and loads all the meta files into memory.

The managed one creates the representative meta files for the representative images from the batch. First of all, when it gets through all the steps that we explained at SIFT algorithm subsection, it chooses only the images that have the almost 10-16 percent similarities with images treated before. In the end, we obtain only 10 from 50 - 60 images appreciatively, also 10 meta files (with the keypoints for them), for the most representative images, in this case, every room that has been loaded as a training directory.

3.2 The Client Module

The client module is the tester and it has two phases: a naive one and a more precise one. This implies comparison at its bases. This module only sends one image to the server and receives a list of results that represents, in this case, the list with the rooms where the images for testing belong.

4 Results

In Robot Vision task participants were submitted 21 runs and we submitted 5 runs. Our runs details are presented in next table:

Table 1: UAIC runs in Robot Vision Task

Run ID	Details	Score	Ranking
155	Full search using all frames Run Duration: 2.5 days on one computer	787.0	**3**
157	Run Duration: 2.5 days for this run	787.0	4
156	Search using representative pictures from all rooms Run Duration: 30 minutes on one computer	599.5	7
158	Run Duration: 30 minutes for one run	595.5	8
159	Wise search with Unknown threaded. + trained with representative images from each room (removed similar/appropriate images) + faster search + unknown case treated	296.5	15

The results were more explicit on the brute finder even though it took a lot of time to complete the training. The 'get representative' method didn't give the expected results, but it is faster like time duration in comparison with the brute method.

4 Conclusions

This paper presents the UAIC system which took part in the Robot Vision task. We used to apply a feature-based method (SIFT) and two main systems in order to search and classify the given images.

The first system uses the most important/representative images for a room's category. The second system is a brute force one. The entire recognition system consists in a server – client architecture. Server is processing one single search at a time, and after the search ends connection with the client closes.

The results shows how the *brute method* is better like quality of results in comparison with *get representative image method*, but it is much slower like time duration. For the future we will try to do a combination between these two methods.

Acknowledgements

The authors would like to thank to the students Ştefan Orzu and Bogdan Catrinescu for their help and support at different stages of system development.

References

1. Lowe, D. G.: Object recognition from local scale-invariant features. *In proceedings of the International Conference on Computer Vision* (1999)
2. Lindeberg, T.: Feature detection with automatic scale selection. *In International Journal of Computer Vision*. (1998) http://www.nada.kth.se/cvap/abstracts/cvap198.html
3. Lowe, D. G.: Distinctive Image Features from Scale-Invariant Keypoints. In *International Journal of Computer Vision* (2004) http://citeseer.ist.psu.edu/lowe04distinctive.html
4. Lindeberg, T. and Bretzner, L.: Real-time scale selection in hybrid multi-scale representations. *In Proc. Scale-Space'03, Springer Lecture Notes in Computer Science*. (2003) http://www.nada.kth.se/cvap/abstracts/cvap279.html.
5. Lindeberg, T.: Scale-space theory in computer vision. *In Kluwer*. (1994)
6. Luo, J., Pronobis, A., Caputo, B. and Jensfelt, P.: The IDOL2 database. *In KTH, CAS/CVAP*, Tech. Rep. (2006). Available at http://cogvis.nada.kth.se/IDOL2/