Combining invariant features and localization techniques for visual place classification: successful experiences in the robotVision@ImageCLEF competition

Jesus Martinez-Gomez, Alejandro Jimenez-Picazo, Jose A. Gomez and Ismael Garcia-Varea

Abstract—In the last decade competitions proved to be a very efficient way of encouraging researchers to advance the state of the art in different research fields in artificial intelligence. In this paper we focus on the optional task of the RobotVision@ImageCLEF competition, which consists of a visual place classification problem where images are not isolated pictures but a sequence of frames captured by a camera mounted on a mobile robot. This fact leads us to deal with this problem not as stand-alone classification problem, but as a problem of self localization in which the robot's main sensor only captures visual information. Thus, we base our proposal on a clever combination of Monte-Carlo-based self-localization methods with optimized versions of scale-invariant feature transformation algorithms for image representation and matching. The goodness of our approach has been validated by being the winners of this task in the 2009 RobotVision@ImageCLEF and 2010 RobotVision@ImageCLEF@ICPR competitions.

Index Terms—computer vision, robot localization, place recognition, semantic place representation.

I. INTRODUCTION

IN the last decade competitions proved to be a very efficient way of encouraging researchers to advance the state of the art in different research fields in artificial intelligence: KDDCUP (data mining), RoboCup (robotic soccer), Image processing and retrieval (at CLEF or ICPR), Computational Intelligence in Games, DARPA grand and urban challenge (autonomous driving), time series forecasting (at IJCNN), etc.

Image classification is one of the most difficult problems in computer vision research. This problem becomes highly complex when images are captured by a robot's camera within dynamic environments with occlusions and illumination changes. One of the main applications of visual classification is robot localization, but this adds several constraints to the process. The most important one is the processing time, as images need to be processed in real-time.

Visual classification techniques with applications in indoor robot localization (visual place classification) typically use the information retrieved from sequences of training images.

Jesus Martinez-Gomez, Alejandro Jimenez-Picazo, Jose A. Gomez and Ismael Garcia-Varea are with the I3A Research Institute, Campus Universitario s/n, 02071, Albacete, Spain. E-mail: jesus_martinez@dsi.uclm.es

E-mail: jesus_martinez@dsi.uclm.es, ajimenez@dsi.uclm.es, Jose.Gamez@uclm.es, Ismael.Garcia@uclm.es

This work was supported by the Spanish "Junta de Comunidades de Castilla-La Mancha" under PCI08-0048-8577 and PBI-0210-7127 Projects and FEDER funds. The support is gratefully acknowledged.

The approaches presented here carry out classification by using techniques such as Scale-Invariant Feature Transform (SIFT) [10], RANSAC [4], and the well-known Monte Carlo localization method [3].

SIFT is used to extract invariant features from images and also to perform a preliminary matching. RANSAC provides us with a useful technique to improve the first matching obtained with SIFT by discarding invalid correspondences. The Monte Carlo localization method was used to take advantage of the similarity between consecutive frames and its relationship with the robot's location.

All experiments were carried out following the proposed procedure, using the appropriate training sequences and a final test sequence. Our proposals were evaluated for the two proposed tasks: obligatory (classification must be performed separately for each test image) and optional (the algorithm can exploit the continuity of the sequences), which is a more realistic localization task.

The rest of the article is organized as follows: a brief introduction of the task is given in Section II; Section III provides an overview of the main proposals from other participant research groups; all the techniques that have been used to develop our proposals are explained in Section IV; Section V presents a full description of the proposals for 2009 and 2010; Section VI offers official results to show the performance of the proposed solutions and conclusions and future work are discussed in Section VII.

II. ROBOT VISION TASK

The RobotVision@ImageCLEF task [18], [16] addresses the problem of visual place classification. Participants are asked to classify rooms and semantic areas on the basis of image sequences captured by a camera. This camera is mounted on a mobile robot that moves within an office environment under varying illumination conditions. Participant systems should be able to answer the question "where are you?" when presented with test sequences containing images acquired in the previously observed part of the environment (using different viewpoints and acquired under different conditions) or in additional rooms that were not imaged in the training stage (these rooms should be labelled with the new room category).

There are two subtasks: obligatory and optional. For the first subtask, the classification has to be performed without taking

into account the order and the relationship between frames. The second subtask is optional but is more related to robot localization. Within this subtask, the classification algorithms can take advantage of the continuity of the sequence of test frames.

The competition starts with the release of annotated training and validation image sequences from a database (*COLD-Stockholm* [15] in 2010 and *KHL-IDOL2* [12] in 2009). These data sets could be used to train different systems and to evaluate their goodness by using the validation data set.

The final test image sequence is released later and contains images acquired in the same environment but including additional rooms not imaged previously. In addition, lighting conditions vary from the training to the validation and test sequences.

A. Data Set

There are three main data sets: training, validation and test. Training sequences are used to generate the classification systems and validation sets can be used to evaluate the performance of preliminary proposals. Training and validation frames are labelled with the room and, therefore, some preliminary scores can be obtained. Test frames are not labelled and the final score for a participant's proposal can only be obtained by submitting the results to the organizer's website.

The number of frames in the different data sets can be observed in Table I.

Sequence	Frames	Rooms	Lighting Conditions
2009 Edition			
Training 1	1034	5	Night
Training 2	915	5	Cloudy
Training 3	950	5	Sunny
Validation 1	952	5	Night
Validation 2	928	5	Cloudy
Validation 3	909	5	Sunny
Test	1600	6	Cloudy
2010 Edition			
Training Easy	8149	9	Cloudy
Training Hard	4535	9	Cloudy
Validation	4783	9	Night
Test	5102	12	Night

TABLE I
ROBOTVISION DATA SET INFORMATION

The main difference between the *KHL-IDOL2* (2009) and the *COLD-Stockholm* (2010) database is the information provided with the training image sequences. Images from the 2009 database were labelled with the room where the frame was taken and the complete $\langle x, y, \theta \rangle$ of the robot's pose. The map of the environment was also provided. With respect to the 2010 database, the information related to the robot's pose was removed and therefore only the room in which the frame was captured was provided.

All classes used to label training and validation frames are semantic terms extracted from indoor environments such as "corridor" or "printer area". Figure 1 shows several example images from the cloudy training sequence of the 2009 CLEF

Fig. 1. Examples of images from the CLEF 2009 database.

Fig. 2. Examples of images from the CLEF 2010@ICPR database.

edition and an image from the test sequence (bottom-right). For the first edition, only 5 different semantic categories were used for training and just one new room not imaged previously (toilet) was added for the test sequence.

The number of semantic categories used for the task was drastically increased for the 2010@ICPR edition (from 5 to 9). Three categories were kept from the 2009@ImageCLEF edition (corridor, printer area and kitchen), and six new categories were added (elevator, laboratory) or modified from the previous edition (student office, large office 1, small office and large office 2).

B. Performance Evaluation

Robot Vision task organizers use several rules to calculate the final score for each submitted run:

- +1.0 points for each correctly classified image (correct detection of an unknown room is treated in the same way as correct classification).
- -0.5 points for each misclassified image.
- +0.0 points for each image that was not classified (the algorithm refrained from the decision).

The final score will be a sum of points obtained for the test sequence. The test sequence consists of a single set for the 2009 edition and two sets (easy and hard) for the 2010@ICPR one. This score is used as a measure of the overall performance of the user proposals.

III. OTHER GROUP'S PROPOSALS

Seven groups registered and submitted at least one run for the 2009 edition of the Robot Vision task [18], with a total of 27 submitted runs. The number of participant groups increased to 8 for the 2010@ICPR edition [17], and the number of submitted runs also rose (29).

A. Glasgow - 2009

The multimedia Information Retrieval Group from the University of Glasgow achieved second position for the optional task using point matching technologies with rule-based decision techniques. Point extraction was performed by employing the Harris corner detector [8] and an illumination filter called Retinex [21] was applied to increase the number of interest points that can be extracted. The main drawback of this proposal is the hard assumption that all frames will contain similar content and geometric information. This assumption will be wrong with hard variations for the viewpoint the frames were taken from. The Harris corner detector will fail when facing changing environments and another invariant point extractor should be selected.

B. LSIS - 2009

Another interesting solution for the 2009 task was that proposed by the *Laboratoire des Sciences de l'Information et des Systemes (LSIS)*, La Garde-France [7]. This proposal constructs a Support Vector Machine (SVM) using two different types of features: the new Profile Entropy Features (PEF) and the generic Descriptor of Fourier (DF) [20]. PEF is proposed by the LSIS group and combines RGB colour and texture, yielding one hundred dimensions. The SVM classifier is generated by using 19 sub-classes created from the 5 original rooms. A total of 19 different binary SVM classifiers were generated and their outputs were combined to get the final decision. The SVM model generated on PEF achieved 4th position for the obligatory task but needed much less training time than the other systems (around 50 times less). The SVM model using only the Descriptor of Fourier obtained a negative score, indicating the importance of the features extracted (even using an efficient and robust classifier such as SVM).

C. CVG - 2010@ICPR

The winner of the obligatory task for the 2010@ICPR edition was the Computer Vision Group, from the ETH Zurich. This group presented a novel and interesting proposal [6] based on the use of visual words [5] computed from SIFT features. Each test frame is classified using a similarity ranking performed using the visual words, already extracted from the training frames. An additional and useful constraint is considered, based on a geometric verification. This verification

uses depth information and is computed using the planar 3-pt algorithm [14], which assumes that the robot is moving in a plane.

IV. TECHNIQUES

This section gives a brief introduction to all different techniques that were used in our proposal for the Robot Vision challenge. They all are well-known techniques that have proved their goodness in different scenarios. Monte Carlo was the selected localization method, using SIFT for the image processing (needed for the update phase). RANSAC was used to improve SIFT matching by discarding outliers.

A. SIFT

In order to perform a correct comparison between (at least) two frames representing the same scenario, appropriate features should be used. These features have to be invariant to changes in the position where the frames were captured. Frames to compare can be acquired over different lighting conditions but such changes should not affect the quality of the comparison. In addition to this, some elements in a dynamic environment can be removed, added or replaced.

All these factors mean that it makes no sense to use classical computer vision techniques based on the Hough transform [22] (such as line or square recognition) to solve the problem of robot localization using visual information. These techniques rely on specific scenario elements which (in the problem we are dealing with) are liable to be removed or replaced.

The most popular technique for extracting relevant features from images is the Scale-Invariant Feature Transform (SIFT) [11]. The main idea of this algorithm is to apply different transformations and study the points of the image which are invariant under these transformations. These extracted points can be used to perform object recognition by carrying out a matching between images representing the same object or scenario.

Features extracted are invariant to image scale and rotation, and they are also robust to noise and changes in viewpoint. An important characteristic of systems developed to perform object recognition using SIFT is that they are robust to partial object occlusion. On the other hand, the algorithm presents a high computational cost that is an important drawback for real-time systems.

In order to deal with the considerable processing time of the algorithm, an implementation of the algorithm over the graphics processor unit (GPU) was considered. The selected implementation (named "SiftGPU"¹) speeds up the process and allows us to perform a higher number of experiments.

B. RANSAC

Random Sample Consensus [4] (RANSAC) is a non-deterministic iterative method for estimating a mathematical model from a dataset. The idea behind RANSAC is to find a significant group of points which are all consistent with a specific model and reject the remaining points as outliers.

¹<http://www.cs.unc.edu/ccwu/siftgpu/>

In order to achieve this goal, RANSAC iteratively estimates models using a random subset of points and evaluates these models with the complete dataset. The algorithm ends when certain constraints (such as the obtained model accurately fitting the data) are overcome or after a maximum number of iterations is reached. The RANSAC paradigm is formally stated as follows:

- Given a model that requires a maximum of n data points to instantiate its free parameters, and a set of data points P such that the number of P is greater than n [$\#(P) \geq n$], randomly select a subset $S1$ of n data points from P and instantiate the model. Use the instantiated model $M1$ to determine the subset $S1^*$ of points in P that are within some error tolerance of $M1$. The set $S1^*$ is called the consensus set of $S1$.
 - If $\#(S1^*)$ is greater than some threshold t , which is a function of the estimate of the number of gross errors in P , use $S1^*$ to compute a new model $M1^*$.
 - If $(S1^*)$ is less than t , randomly select a new subset $S2$ and repeat the above process. If, after some predetermined number of trials, no consensus set with t or more members has been found, either solve the model with the largest consensus set found, or terminate in failure.

There are three free parameters in RANSAC:

- The error tolerance that defines whether or not a point is compatible with a model;
- The number of subsets that defines the number of iterations;
- The threshold t that defines the number of compatible points used to decide if the right model has been found.

Our proposal for the 2010@ICPR edition of the Robot Vision task uses RANSAC to improve the initial matching obtained with SIFT. Such initial matching obtains a high number of outliers that do not fit the real correspondence between two candidate images. Fig.3 illustrates the result of a matching between the invariant points extracted from two images and how the outliers (red lines) are discarded using RANSAC.

C. Monte Carlo Localization Method

The Monte Carlo Localization method [3] (denoted as MCL-method) is a method to determine the position of a robot given a map of its environment. This method is basically an implementation of the particle filter applied to robot localization and has been proposed for use within indoor dynamic environments. The MCL-method keeps information about different environment locations and allows us to represent uncertainty about the robot's localization.

The main alternative to the use of the MCL-method for developing a robot localization algorithm are (Extended) Kalman filters [13]. This alternative method was not considered for our 2009 edition proposal because it only keeps information about a single robot's location and it is necessary to know the robot's initial pose. Moreover, this method presents problems when the uncertainty about the pose increases significantly.

Fig. 3. SIFT matching between two images where outliers (red lines) are discarded.

Using a particle filter approach, the density of the localization is represented by a set of N random samples drawn from it. Each one of these samples or particles represents a robot's pose $\langle x, y, \theta \rangle$. Using the set of samples ($S_k = \{s_k^i; i = 1..N\}$) we can approximately reconstruct the density and estimate the robot's localization.

At each time-step k the MCL-method recursively computes the set of particles that is drawn from the information sensed at k . In the scope of ImageCLEF RobotVision, the information sensed from the environment is that retrieved from the image captured at time-step k .

The iterative MCL-method proceeds in two phases (where $\mathbf{u}_k$ denotes the movement order sent to the robot at time-step k and $\mathbf{z}_k$ represents the information sensed from the environment at k):

Prediction Phase This phase starts from a set of particles S_{k-1} computed in the previous iteration. The method applies the motion model to each particle s_{k-1}^i by sampling from the density $p(\mathbf{x}_k | s_{k-1}^i, \mathbf{u}_{k-1})$:

- (i) for each particle s_{k-1}^i : draw one sample s_k^i from $p(\mathbf{x}_k | s_{k-1}^i, \mathbf{u}_{k-1})$

We obtain a preliminary set of particles at time k S'_k , when we have not yet incorporated any sensor measurement.

Update Phase The second phase takes into account the measurement $\mathbf{z}_k$, and weights each of the samples in S'_k by the weight $m_k^i = p(\mathbf{z}_k | s_k^i)$, i.e. the likelihood of s_k^i given $\mathbf{z}_k$.

We then obtain S_k by resampling from this weighted set:

(ii) for $j=1..N$: draw one S_k sample s_k^j from $\{s_k^i, m_k^i\}$

The resampling selects, with higher probability, samples s_k^i that have a high likelihood associated with them, and in doing so a new set S_k is obtained that approximates a random sample from $p(\mathbf{x}_k|Z^k)$.

After the update phase, steps (i) and (ii) are repeated iteratively. The filter is initialized at time $k = 0$ with a random sample $S_0 = \{s_0^i\}$ from the prior $p(x_0)$.

While the method is being applied, the best particles should be duplicated and the worst particles should disappear from the set of samples. After some iterations, the algorithm should converge with all particles around the real robot's pose. This situation is shown in Fig. 4, where blue circles represent the particles for three time-steps. The size of the circumferences represents the weight of the particles $m_k^i = p(\mathbf{z}_k|s_k^i)$. It can be observed how most of the low-weight particles from the first iteration disappear in the third one. On the other hand, high-weighted particles are duplicated.

V. PROPOSAL

All our work is focused on the optional task because we want to develop robotic systems that can feasibly be used in the real world. For the obligatory task (mandatory) we presented the result of applying a subset of our processing. This subset was used to define the image processing necessary to perform the optional task. The temporal and spatial relationship between consecutive frames was not considered for the obligatory task, where the sequence of test frames could differ from the original one.

A. 2009 Proposal

Our proposal for the 2009 RobotVision@ImageCLEF competition was to develop a localization method based on the use of particle filters. We used an MCL method and (as was mentioned in Section IV-C) we needed a motion model (for the prediction phase) and an image processing algorithm to sense the information from the environment (and use this information within the update phase).

1) *Image Processing*: A preliminary pre-processing step was applied to extract all the available information from the training sequences. This process stored (for each training frame) the complete pose of the robot $\langle x, y, \theta \rangle$, labelled with the correctly classified room C_i and the set of m SIFT points extracted from that frame $[P_1, \dots, P_m]$. This feature extraction can be performed offline.

Once we had extracted all training information, we defined the way to classify each test frame. This classification was performed based on the similarity computed between each test frame and all training frames. For each test frame, we obtained a similarity ranking and classified that test frame with the label of the k most similar training frames. This process is similar to the k -nearest neighbor algorithm [2].

The main problem of this approach is the processing time necessary to perform a SIFT matching ($\cong 0.18sec$ per image). Another important drawback is the problem of invariant feature extraction (by using SIFT) when facing lighting changes. Extracted SIFT features heavily depend on lighting conditions under the frame was taken. Therefore, matchings between two frames taken from the same viewpoint (and room) but acquired under different lighting conditions will not obtain a high similarity score.

In addition to this slow processing based on similarity, we added an extra step based on the detection of natural environment landmarks. Robot localization by using environment landmark detection is commonly used in the scope of the RoboCup [1], [19]. We applied a Hough transform[22] to study the distribution of the lines and squares obtained. Environment landmarks such as the corridor ceiling or the external door located in the printer area were selected as unchanging sections. Some examples of these detections can be observed in Fig. 5, where the corridor class is detected. A corridor situation is defined as a frame with a large number of vertical lines (higher than 20) and two symmetric lines starting from the top of the image and representing the ceiling. Vertical lines will correspond with to doors, wardrobe and other elements likely to appear within a corridor. The execution time of this extra step on a current computer was lower than 0.005 seconds.

Fig. 5. Corridor detection. Purple lines are candidates and green lines are the correct ones.

Natural landmarks should be selected carefully, taking into account that these landmarks should be highly discriminative elements that are easy to detect under changing lighting conditions. All selected landmarks should have specific characteristics that cannot appear in new unknown rooms. We only used the corridor ceiling (larger than all the other room ceilings) and a window combination for the Printer Area.

Fig. 6. Printer Area detection. Purple squares are candidates and green squares are the correct ones.

2) *Localization Algorithm*: In order to reduce the number of comparisons and to take advantage of the similarity between consecutive test frames, we added a particle filter. It

Fig. 4. Particle convergence after three iterations of the MCL method, where circles represent particles and the size of circles is used for a particle's weight.

was necessary to define a particle representation and how the prediction and update phases are iteratively applied.

Particle Representation: Each particle is represented by three numerical values (representing the pose of the robot): x and y are used for position and θ for orientation. These parameters have continuous numeric values. The limits for x and y values are the boundary of the environment. In this case, values for the x component are between 0 and 1500 centimetres. Bounds for the y component are 0 and 2500 centimetres.

Prediction Phase: This step applies a movement model without knowing the movement order sent to the robot between test frames. We assume that the robot movement will be similar to that performed during the training (training frames were acquired while the robot was moving). The average robot velocity was estimated from the difference between the poses of the training sequence. We obtain the average linear and angular velocity, which are defined in centimetres (and degrees) per frame.

We add white noise in this step to represent some uncertainty in the obtained movement. This noise applies a random variation based on the particle's weight, obtaining higher variations for worst particles. Using this approach, best candidates will obtain minor changes.

Update Phase: This phase obtains the weight for each particle using the information extracted from the test frame

to classify and from the training sequence. Each particle is weighted by matching the SIFT points of the test frame with those extracted from the training frame taken from the pose the particle represents. We only have SIFT points extracted from a discrete number of poses in the environment and so we have to select the nearest training frame to the particle's pose.

Once we have weighted all particles, the robot's position is estimated using the particle information: x_i , y_i and weight w_i . The average robot position is obtained as a weighted sum, taking into account the particle weight:

$$\bar{x} = \sum_{i=0} x_i \times w_i, \quad \bar{y} = \sum_{i=0} y_i \times w_i$$

Each test frame will be labelled with the room belonging to the average robot position, or not labelled if this position is between two rooms.

We noticed that some test frames were not represented by any training frame, failing all the available SIFT matchings. The main consequence of this situation is that the weight of all the particles decreases continuously and they are spread over the environment. When facing this situation and finding new false positives, particles will converge to wrong areas and the robot's location will not be correctly obtained.

In order to avoid this situation and to escape from wrong convergences, we added an extra step to the original process (basic MCL method). This step performs a population initialization when the best particle's weight is below a certain threshold (similar to approaches to escape from local optima).

This first modification improved the system behaviour but the process became unstable and the algorithm presented problems to achieve convergence.

At this point, we decided to restrict the negative effect of population initializations, performing them over restricted environment areas. We assumed that it is possible to detect when our algorithm converges (all particles are close together) and that this event represents a perfect pose estimation (particles represent robot's current pose).

Future population initializations will be carried out over a restricted area. This area is defined as a circumference with radius r and centred at point $\langle x_a, y_a \rangle$. This radius r is a function of the instability level (obtained with the best particle's weight for the last n iterations) and the value of point $\langle x_a, y_a \rangle$ that was obtained under the last stable situation representing the robot's most reliable pose.

Fig. 7. Population re-initialization over a restricted area represented by the green circle.

A re-initialization is shown in fig 7, where a new population is created by spreading particles over the area covered by the green circle.

The stability of the algorithm could be estimated by studying the variation in the pose of the particle. The process will be stable when the variation obtained for the x and y components of the last n pose estimations is sufficiently small (all particles are spread over a small portion of the environment). The next step was to define a state for the algorithm based on its stability: When the algorithm is stable(i), no population initializations are performed. Otherwise, the initialization depends on the instability level. If the algorithm has been stable for the last few frames and it becomes unstable(ii), initialization is performed over a restricted area (a circumference centred at the most reliable robot position, obtained from previous iterations with a stable process). The size of this area depends on the instability level. If the algorithm has been unstable for the last few frames(iii) and a new initialization has to be applied, all the particles will be spread over the whole environment.

The whole process is shown in Fig. 8, where these three

situations (and the action associated to each case) are shown.

Fig. 8. General Processing Scheme.

B. 2010 Proposal

In this edition, the training information provided by the RobotVision@ICPR organizers prevented us from applying localization principles. Training frames were just labelled with the room where they were taken and the map of the environment was not provided. The task was reduced to a visual place classification problem and we decided to improve the classification techniques used for the 2009 edition.

First of all we adopted the GPU implementation of the SIFT algorithm (known as SiftGPU). This implementation speeded up the process for the system training and allowed us to perform a higher number of experiments and tests.

Training frame pre-processing. In order to reduce the amount of information to work with and to discard redundant frames, we added training sequence pre-processing. All training frames were converted to greyscale. After this conversion, we computed the difference between two frames as the absolute difference for the colour, pixel by pixel (an example of the difference is shown in Fig. 9). Each frame whose difference between it and the last non-removed frame was lower than a certain threshold (0.05% in all our experiments) was removed from the training sequence.

Once we had rejected all redundant information, we selected a subset of the most representative training frames. This process was performed as a k -medoids [9] clustering algorithm and the similarity between two frames was computed by matching the SIFT points extracted from them. An example of the training sequence pre-processing is illustrated in Fig. 10, where the first row of frames represents the complete training sequence, the second row is used to show which frames are removed (because they are redundant). Finally, the most representative frames are shown in the third row in Fig. 10.

Frame Classification. The complete process for classifying a test frame consists of three steps. First, SIFT points are extracted from the current frame. Second, we compute the

Fig. 10. An example of sequence pre-processing where redundant frames are discarded and best candidates are selected as representative.

Fig. 9. Difference computed between two frames

similarity between the current frame and the training frames by means of the percentage of matching points (SIFT + RANSAC). Finally, these similarities are used to classify the test frame.

The similarity value between a test frame and a training frame is obtained using SIFT matching and RANSAC. After all the SIFT matching points are obtained, RANSAC is applied to discard the outliers. The percentage of common points between both frames is stored as the similarity value.

Each test frame can be classified as a room, marked as unknown or not classified. A ranking with the best n similarity values and its associated rooms is obtained. We compute the sum of the similarity values separately for the different rooms in the ranking. The test frame will be classified as the room with the highest value when this value clearly exceeds all the other ranking rooms, otherwise it will not be classified. Unknown is used to denote a test frame acquired in a room

not included in the training rooms and will be used when the maximum similarity value is below a certain threshold (0.05 in the experiments). A complete classification process where the test frame is matched with all the selected training frames can be observed in Fig. 11. In this case the test frame should be labelled as a corridor.

Test Frame						
Best 6 training frames						
Room	Corridor	Corridor	Corridor	Kitchen	Corridor	Kitchen
% of common SIFT points	0.32	0.28	0.21	0.12	0.11	0.08

Fig. 11. Test frame classification using a similarity ranking of 6 training frames

For the optional task (where the algorithm can exploit the continuity of the sequence) we took into account the test frame we are going to classify and the last 4 test frames already classified. Test frames initially labelled as not classified were labelled as the room used to classify the last 4 frames when this room was the same. In addition to these considerations, we avoided passing from one room to another without using the corridor (the last test frame was labelled as not classified)

VI. RESULTS

Our proposals were evaluated by the RobotVision@ImageCLEF organizers. The performance of each algorithm was evaluated using a score that is computed as follows: for each correctly classified frame the score is updated by +1.0, for each misclassified one the score is

reduced by -0.5 . A non-classified frame does not alter the score.

Each run consisted of submitting the results for the test sequence, using the information provided by the training sequence.

Participants in 2009 were asked to use a specific training sequence acquired under night lighting conditions, but in 2010, two training sequences had to be used: easy and hard. The amount of information to work with was higher when using the easy training sequence. For the easy training set, the robot was driven through the environment following a similar path as for the test sequence. The direction followed by the robot in the hard training set was the opposite to that used for the easy one.

A. 2009 Preliminary Experiments

Before submitting any result file to the website, we performed a set of preliminary experiments using all the available training sequences. We generated three different classifiers, which were trained using the training sequences acquired under the three lighting conditions (cloudy, night and sunny). Each one of these classifiers was tested using three validation sequences (one for each lighting condition). These three validation sequences were selected from the set of validation sequences provided by the organization (*KHL-IDOL2* database). For each combination of training and validation, we applied our processing for the obligatory and optional task, storing the number of correctly classified, not classified and misclassified frames. A detailed view of the results obtained is presented in Table II.

From data collected in Table II (Obligatory task), we can conclude that there is a high dependency of the system on lighting changes. This dependency makes the algorithm obtain worse results when training the classifier with different lighting conditions than those used for testing. This situation is clearly revealed for the obligatory task, where the best score was always obtained using the same lighting conditions for training and testing.

This dependency is not so important when using the additional processing included in the optional task. For all the test sequences, (at least) 60% of frames were correctly classified. It can be noticed that scores obtained for the different test sequences are not so strongly dependent on training illumination conditions as those obtained for the obligatory task. By performing a comparison between these results and those obtained for the obligatory track, we can state that the MCL-method notably improved the results for the optional task.

B. 2009 Competition

All internal thresholds were tuned using the proposed validation sequences: the process was estimated as unstable when the best particle's weight was below 0.05 or when average variation for the x (or y) component was above 2 (or 3) meters. The maximum radius of the circumference used to initialize populations was 3 meters, and its value was calculated with $3 \cdot \bar{m}$ (where $\bar{m}$ denotes the average weight for all the population particles).

Different runs were submitted to the website and the best score obtained was 916.5 for the optional task and 511.0 for the obligatory one. Table III shows the complete results obtained for the 2009 competition, with the number of correctly classified (C), misclassified (M) and not classified frames (NC). We achieved 5th position for the obligatory task (the best score of 793.0 was obtained by the Idiap Research Institute from Switzerland).

Obligatory task - 5th position	
Score	511.0
Correctly Classified	676
Missclassified	330
Not Classified	684
Optional task - 1st position	
Score	916.5
Correctly Classified	1072
Missclassified	311
Not Classified	217

TABLE III
RESULTS FOR THE 2009 COMPETITION

Thanks to the use of a particle-filter algorithm, the final score for the optional task improved upon the results obtained for the obligatory one (from 511.0 to 916.5). The final score for the optional task was the highest one of all submitted runs from all participants, and the SIMD group of the University of Castilla-La Mancha was the winner for the optional task.

C. 2010 Competition

For the 2010 competition, each run consisted of submitting the results for the two training sets: easy and hard. The complete results are shown in Table IV, and as can be observed, our run achieved 3rd place for the obligatory task, for which 8 different research groups submitted results.

Obligatory task - 3rd position			
	Total	Easy	Hard
Score	3372.5	2000.0	1372.5
Correctly Classified	3886	2180	1706.0
Missclassified	1027	360	667
Not Classified	189	11	178
Optional task - 1st position			
	Total	Easy	Hard
Score	3881.0	2230.5	1650.5
Correctly Classified	4224	2332	1892
Missclassified	686	203	483
Not Classified	192	16	176

TABLE IV
RESULTS FOR THE 2010 COMPETITION

Our proposal again obtained 1st position for the optional task, for which 4 different groups submitted results.

VII. CONCLUSIONS AND FUTURE WORK

The current article presents all the proposals of our group (SIMD) for the first two editions of the RobotVision task,

TRAINING SEQUENCE	VALIDATION SEQUENCE												
	Obligatory task												
		Night (952 frames)				Cloudy (928 frames)				Sunny (909 frames)			
		Score	C	M	NC	Score	C	M	NC	Score	C	M	NC
	Night (1034 fr.)	531	643	224	85	285	433	265	230	265.5	421	311	177
	Cloudy (915 fr.)	270.5	426	311	215	404.5	538	267	123	420.5	534	227	148
	Sunny (950 fr.)	285.5	435	299	218	358.5	457	197	247	509	615	212	82
	Optional task												
		Night (952 frames)				Cloudy (928 frames)				Sunny (909 frames)			
		Score	C	M	NC	Score	C	M	NC	Score	C	M	NC
Night (1034 fr.)	837.5	861	47	44	534.0	635	202	91	476.5	560	167	182	
Cloudy (915 fr.)	600.5	695	189	68	680.5	748	135	45	733.5	774	81	54	
Sunny (950 fr.)	725.0	791	132	29	701.0	769	136	23	798.5	823	49	37	

TABLE II

SCORE, NUMBER OF CORRECT, INCORRECT AND NON CLASSIFIED FRAMES FOR DIFFERENT COMBINATIONS OF TRAINING AND VALIDATION FRAME SEQUENCES. PRELIMINARY RESULTS OBTAINED USING SEQUENCES FROM THE 2009 COMPETITION.

within the CLEF campaign. According to the results obtained for the optional task in both editions, visual place classification is clearly helped by the application of robotic localization principles.

Two well-known techniques, namely SIFT and RANSAC, were used to perform the matching or comparison between images. The disadvantage of SIFT's high processing time was solved by using a new implementation over the GPU. RANSAC has been properly used to discard the outliers and to improve the matching between SIFT points.

The percentage of common points between the test frame and the best training frame can be used to weight the classification. This information could be highly useful for estimating classification quality or performance.

For future work, we aim to develop an indoor localization system capable of being trained automatically. This system will use the information extracted from vision cameras and that obtained from robot actuators. This future line of research is related to the integration of both sources of information: visual and odometrical.

ACKNOWLEDGEMENTS

The authors acknowledge the financial support provided by the Spanish "Junta de Comunidades de Castilla-La Mancha (Consejería de Educación y Ciencia)" under PCI08-0048-8577 and PBI-0210-7127 Projects and FEDER funds.

REFERENCES

- [1] G. Adorni, S. Cagnoni, and M. Mordonini. Landmark-based robot self-localization: a case study for the robocup goal-keeper. In *Information Intelligence and Systems, 1999. Proceedings. 1999 International Conference on*, pages 164–171, 1999.
- [2] T. Cover and P. Hart. Nearest neighbor pattern classification. *Information Theory, IEEE Transactions on*, 13(1):21–27, 2002.
- [3] F. Dellaert, D. Fox, W. Burgard, and S. Thrun. Monte carlo localization for mobile robots. In *Proc. of the IEEE International Conference on Robotics*, 1999.
- [4] M. Fischler and R. Bolles. Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. 1981.
- [5] F. Fraundorfer, C. Wu, J. Frahm, and M. Pollefeys. Visual word based location recognition in 3d models using distance augmented weighting. In *Fourth International Symposium on 3D Data Processing, Visualization and Transmission*, volume 2, 2008.
- [6] F. Fraundorfer, C. Wu, and M. Pollefeys. Combining monocular and stereo cues for mobile robot localization using visual words. In *International Conference on Pattern Recognition (in press)*, 2010.
- [7] H. GLOTIN, Z. ZHAO, and E. DUMONT. Fast LSIS Profile Entropy Features for Robot Visual Self-Localization. *Working Notes of CLEF*, 2009.
- [8] C. Harris and M. Stephens. A combined corner and edge detector. In *Alvey vision conference*, volume 15, page 50. Manchester, UK, 1988.
- [9] L. Kaufman, P. ROUSSEEUW, D. D. o. M. Technische Hogeschool, and Informatics. Clustering by means of medoids. Technical report, Technische Hogeschool, Delft(Netherlands). Dept. of Mathematics and Informatics., 1987.
- [10] D. Lowe. Object recognition from local scale-invariant features. In *International Conference on Computer Vision*, volume 2, pages 1150–1157. Corfu, Greece, 1999.
- [11] D. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91–110, 2004.
- [12] J. Luo, A. Pronobis, B. Caputo, and P. Jensfelt. The KTH-IDOL2 database. Technical Report CVAP304, Kungliga Tekniska Hogskolan, CVAP/CAS, October 2006.
- [13] R. Negenborn. Robot localization and kalman filters. 2003.
- [14] D. Ortín and J. Montiel. Indoor robot motion based on monocular images. *Robotica*, 19(03):331–342, 2001.
- [15] A. Pronobis and B. Caputo. COLD: The CoSy Localization Database. *The International Journal of Robotics Research*, 28(5):588, 2009.
- [16] A. Pronobis and B. Caputo. The robot vision task. In W. B. Croft, H. Müller, P. Clough, T. Deselaers, and B. Caputo, editors, *ImageCLEF*, volume 32 of *The Information Retrieval Series*, pages 185–198. Springer Berlin Heidelberg, 2010.
- [17] A. Pronobis, M. Feroni, H. Christensen, and B. Caputo. The Robot Vision Track at ImageCLEF 2010. 2010.
- [18] A. Pronobis, L. Xing, and B. Caputo. Overview of the clef 2009 robot vision track. In C. Peters, B. Caputo, J. Gonzalo, G. Jones, J. Kalpathy-Cramer, H. Müller, and T. Tsikrika, editors, *Multilingual Information Access Evaluation II. Multimedia Experiments*, volume 6242 of *Lecture Notes in Computer Science*, pages 110–119. Springer Berlin / Heidelberg, 2010.
- [19] T. Röfer, T. Laue, and D. Thomas. Particle-filter-based self-localization using landmarks and directed lines. In A. Bredendfeld, A. Jacoff, I. Noda, and Y. Takahashi, editors, *RoboCup 2005: Robot Soccer World Cup IX*, volume 4020 of *Lecture Notes in Computer Science*, pages 608–615. Springer Berlin / Heidelberg, 2006.
- [20] F. Smach, C. Lemaître, J.-P. Gauthier, J. Miteran, and M. Atri. Generalized fourier descriptors with applications to objects recognition in svm context. *Journal of Mathematical Imaging and Vision*, 30:43–71, 2008.
- [21] Z. ur Rahman, D. J. Jobson, and G. A. Woodell. Retinex processing for automatic image enhancement. *Journal of Electronic Imaging*, 13(1):100–110, 2004.
- [22] H. V and P. C. Method and means for recognizing complex patterns, 1962. US Patent 3,069,654.