

Probabilistic Coarse-To-Fine Object Recognition

No Author Given

No Institute Given

Abstract. A coarse-to-fine probabilistic model for objects in images is presented. We build upon the framework introduced in [1, 2], where objects are composed of constellations of features, and features from a same object share a common reference frame. Feature appearance and pose are modeled by probabilistic distributions, the parameters of which are shared across features in order to allow training from few examples. Unlike [2] where the recognition algorithm attempts to propagate seed matches to build a correspondence with database models, the current study uses a coarse-to-fine strategy. A simple model voting followed successively by a Hough transform and by the RANSAC algorithm, are used as successive refinement blocks that discard incorrect matching hypotheses in a cascade. After each step, our belief in the remaining hypotheses - in terms of probability densities - is updated with the information obtained from the filtering procedure. We test our ideas with experiments on two image databases. We compare with D.Lowe's [1] and P.Moreels' [2] algorithms and demonstrate significantly better performance.

1 Comments

From the discussions with Francois Fleuret and Don Geman:

- The difference between the coarse-to-fine tree used by Francois and the ‘cascade’ that we are using, is due to the difference in the nature of the features. Francois uses features that are very ‘poor’. No appearance, the only information is the location of the edge with respect to the reference frame of the face. This allows him to test his set of edges against all possible poses (explored along a smart path with his coarse-to-fine strategy). In contrast, the features we are using are much richer. They carry appearance information, and the geometry is richer too, with orientation and scale. As a consequence, we are not searching the whole space of poses, but only the poses that collected the largest consensus of individual votes (similar to D.Lowe).
- SIFT features have a large degree of invariance. This invariance is useful early in the game, when we have no idea of the position of the model in the scene. However, the accuracy with which location, orientation and scale are computed, is fixed and never modified. The pose estimation can be sensitive to errors in particular in the orientation and scale. Worse, correct pairs of features can be discarded on the basis of erroneous values of orientation and scale. It could be useful to use less invariant features once a first estimate of the pose has been computed. E.g., map non-invariant regions according to the pose estimate, and use template matching between this transformed region and the patch actually observed in the second image.

2 Introduction

Recognizing objects in images is perhaps the most challenging problem currently facing machine vision researchers. Much progress has been made in the recent past both in recognizing individual objects [1], as well as in recognizing object categories [4, 5]. A number of ideas have proven to be key to recent progress. First of all: objects and categories may be represented as collections of features, or parts, each one of which encodes the distinctive appearance of a portion of the object. The mutual position of these features, as well as their appearance, contains information as to the identity of the object. Second: a (small) subset of the object’s features is often sufficient signal to infer its presence in the image: if one enforces both mutual position and appearance constraints one may rule out false alarms arising from features detected in random clutter. Robustness to occlusion and poor feature detection may be obtained by exploiting this fact. Third: very efficient approximate algorithms for matching features in the high dimensional space where the features’s appearance is represented have been discovered [9]. Similarly, there exist efficient algorithms for enforcing geometrical constraints [11]. The combination of these ideas and techniques has given us very efficient and robust object recognition algorithms.

Still, much progress still needs to be made to reach levels of performance that approach those of the human visual system. Error rates are still larger than

10% on fairly benign benchmark image sets. Furthermore, large classes of objects are still difficult to recognize or discriminate: e.g. objects containing repeated textures (e.g. furry teddy bears) and objects with smooth featureless surfaces (e.g. plain coffee mugs). In order to overtake some of the current challenges we need to be creative and generate novel image analysis ideas, e.g. new feature types. Another way to make progress is to place our recent discoveries on a firm theoretical footing. Much of what we know is still a ‘bag of tricks’ – we need to understand better the underlying principles in order to improve our designs and take full advantage of what we can learn from the statistics of images.

The goal of this study is to produce a consistent probabilistic interpretation of some of the most effective techniques we know for the recognition of individual objects [1]. The techniques we use are inspired by the work of Perona and collaborators on the probabilistic ‘constellation’ model [4, 5] for object categories. We are also inspired by D. Geman and collaborators’ work on coarse-to-fine searching [7]. An exploration of this topic was started by Moreels et al. [2]. The current study builds upon their work, and makes three contributions. First, we incorporate in the current framework a number of ‘atomic operations’, such as ‘vote counting’, Hough transform and RANSAC, which allow us to pursue a probabilistic search for the best interpretation of a given image in a coarse-to-fine fashion. Our previous work only used one type of operation: matching individual features; here we start with ‘statistical’ global measurements and eliminate a great number of scene interpretations with very little effort. Second, we benefit from a recent study measuring the variation in position and appearance of features in 3D objects imaged under different viewpoints and lighting conditions – various conditional probabilities in our algorithm are therefore based on careful empirical measurements [6]. Third, although this was not the main goal of our study, our experiments show that our new algorithms perform substantially better than both D.Lowe’s system [1] and previous work from Moreels&al. [2].

Section 3 describes the coarse-to-fine process used to generate hypotheses and sets of features assignments. Section 4 and 5 details the probabilistic model and parameters estimation used to score the hypotheses. Section 6 presents and discusses results, and section 7 contains our conclusions.

3 Hypothesis generation

3.1 Scene and database model

The task starts with a set of images of *model objects* that form the *database of models* M , and a query image or *test scene* F . Our goal is to identify the model objects present in the test scene, along with their pose, i.e. position, orientation and scale.

All test scenes and model objects, are represented by collections of distinctive features. Features are informative and stable parts of the image, detected by an interest point operator. In this work we use the popular combination of

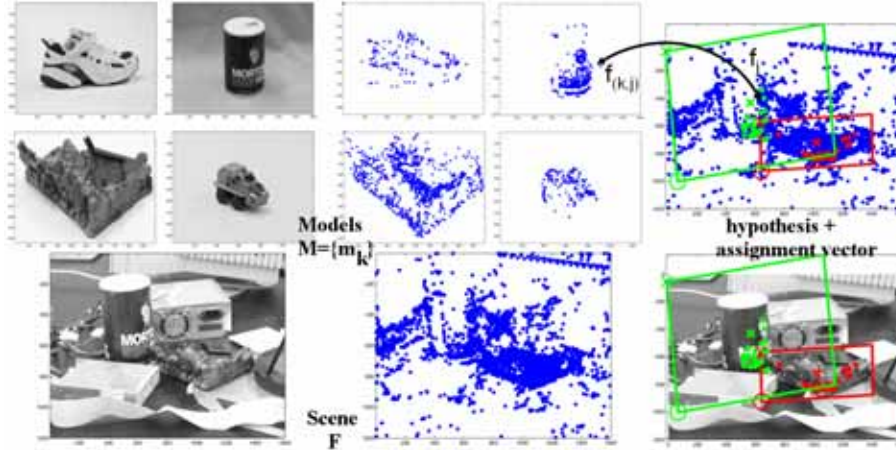


Fig. 1. The recognition problem. (left) Database models and test scene. (middle) Features extracted from models and test scene. (right) The choice of a set of models and poses forms a hypothesis, the choice of a set of features assignments forms an assignment vector.

difference-of-gaussians features and SIFT descriptor proposed by Lowe [1], although a few other options are equally good [6]. To simplify notations, we will also call *database* and denote by M the sets of features extracted in images of model objects, and also denote by F the set of features extracted from the scene image.

Models are indexed by k and denoted by m_k , while indices i and j are used respectively for test scene features and model features: f_i denotes the i -th scene feature, while $f_{(k,j)}$ denotes the j -th feature from the k -th model.

Each feature is described by its *appearance* as well as its *geometrical information*: position, scale and orientation in the image where it was detected. The *geometry* information associated to each feature contains position information, orientation and scale. It is denoted by \mathcal{X}_i for test feature f_i and $\mathcal{X}_{(k,j)}$ for model feature $f_{(k,j)}$. This geometric information is measured relatively to the standard *reference frame* of the image in which the feature has been detected. All features extracted from the same image share the same reference frame. The *appearance* information associated to a feature is a descriptor characterizing the local image appearance near this feature. It is denoted by \mathcal{A}_i for test feature f_i and $\mathcal{A}_{(k,j)}$ for model feature $f_{(k,j)}$.

A *hypothesis* H is an interpretation of a test scene. It is composed of a set of N_H models $\{m_k\}_{1 \leq k \leq N_H}$ and corresponding poses $\{\theta_k\}_{1 \leq k \leq N_H}$. H can contain no model (‘background hypothesis’), one model or multiple models.

An *assignment vector* V carries complementary information to a hypothesis: it assigns each feature from the test scene to a model feature or to clutter. The i -th component $V(i) = (k, j)$ denotes that the test feature f_i is matched to $f_{(k,j)}$, j -th feature from the k -th model m_k . $V(i) = 0$ (or $(0, 0)$) denotes the case where f_i is attributed to clutter.

3.2 Building blocks of the hypothesis generation process

Ideally, we should consider all combinations (*hypothesis* × *assignment vector*). As this exhaustive exploration is unrealistic in terms of computation time, we aim at identifying a small set of likely combinations (H, V).

We divide the matching process into a cascade of steps. At each step more information is considered, and the probability on the possible explanations of the test image is updated. We here describe the basic blocks in the hypotheses filtering process, the probabilistic interpretation will be introduced in sec.4. This sequence of steps can be considered as a coarse-to-fine matching method [7, 8].

- **Model voting.** The first screening of the hypothesis space is done by indexing the scene features into the database. This indexing is based on appearance only, and performed by a kd-tree structure, with backtracking to improve its accuracy [9]. Either one or more closest neighbors are selected to form candidate assignments for the scene feature (one neighbor only in the experiments from sec.6), and candidate matches are formed between scene features and model features to which they were indexed. The variable $\bar{N} = N_k$ indicates for each model, how many times the kd-tree returned a feature belonging to model m_k .

- **Coarse Hough transform.** We use the Hough transform [1, 10] as a first step in the exploration of the pose space. Since the features encode position, orientation and scale, a single pair $(f_i, f_{(k,j)})$ provides enough information to characterize a 2D similarity transform from model to test image, at this stage we restrict ourselves to similarity transforms. The Hough space of parameters is a space with 4 dimensions: translation in x , translation in y , rotation and change of scale. This space is discretized into coarse bins (half the size of the model image for location, 90 degrees in orientation, and a scale factor of 2 - note that this is significantly larger than the bins chosen in [1]), and the candidate matches are hashed into these bins. The choice of a coarse discretization makes the exploration of the Hough table a fast process. Besides, the coarse discretization causes the boundary-related hashing issues to be less evident than in [1]. The counterpart of having large bins, is that this step is not accurate enough to fully characterize hypotheses. Variable \tilde{N} denotes the number of candidate matches falling in each $(model, pose)$ bin. $\tilde{N} = \{N_{(k,b)}\}_{k,b}$ where k indexes the model image and b the pose bin.

- **Hypothesis refinement via RANSAC.** Once the fraction of inliers has been increased by the Hough transform, a RANSAC algorithm [11] is applied to each bin that collected a sufficient number of matches (defined by a threshold T_{hough}). Triplets of matches are sampled randomly. For each of these triplets, pose parameters for an affine transform H are computed using the (x,y) locations. The largest consensus set and the corresponding hypothesis are selected as candidates for this bin.

- **Choice of features assignments.** The previous steps are concerned with hypotheses H , i.e. identity and pose of the models present in the test scene. Concurrently to the RANSAC step, we want to decide which candidate feature assignments should be accepted and which features should be attributed to clutter. The set of accepted features can then be used to refine the pose parameters,

as in [1] For each candidate match $f_i \leftrightarrow f_{(k,j)}$ we compute the likelihood ratio LR_i of $f_{(k,j)}$ being a good match for f_i , versus this feature being a clutter detection (LR_i is described in sec.4). If $R_i > 1$ the match is inserted into V , in the alternative f_i is considered a clutter detection. For a given hypothesis H , this results into an assignment vector $V(F, H)$ which is a deterministic function of the hypothesis H and the data F .

4 Probabilistic interpretation of the coarse-to-fine search

4.1 Probabilistic decomposition

The probabilistic treatment that we propose reflects the coarse-to-fine strategy described in sec.3.2. We want to compare hypotheses scores given the observations made in the test image and given a specific database of models. I.e., we want to evaluate $P(H|F, M)$ and $P(H, V|F, M)$. We have

$$P(H, V|F, M) = \frac{P(F, H, V|M)}{P(F|M)} \quad (1)$$

where $P(F|M)$ is a prior on features observations. We now examine $P(F, H, V|M)$, which we will call *score*:

$$\begin{aligned} P(F, H, V|M) &= P(F, H, V, \tilde{N}, \bar{N}|M) = \\ &P(F|V, \tilde{N}, \bar{N}, H, M) \cdot P(V|\tilde{N}, \bar{N}, H, M) \cdot P(\tilde{N}|\bar{N}, H, M) \cdot P(\bar{N}|H, M) \cdot P(H|M) \end{aligned} \quad (2)$$

where the additional variables \bar{N} and \tilde{N} are defined in sec. 3.2 (these variables are deterministic functions of F, H, V, M).

We first describe the meaning of each one of these terms. In sec. 5 we explain in detail how each term is computed.

- $P(H|M)$. This is a prior on all possible hypotheses. It contains information on which objects are most likely present, together with their most probable pose. In principle, $P(H|M)$ should also encode ‘context’, i.e. which objects are likely to show up in combination, and what is their likely mutual position.

- $P(\bar{N}|H, M)$ predicts the number of features associated to each model during the initial search phase, so this is a “bag of words” model. If the model is present in the test image, the expected number of features in the test scene is directly related to the number of features in models. This term takes into account scale information present in H : objects seen at a smaller resolution will generate fewer features than objects seen from a close viewpoint. This term is updated by the initial model voting step described in sec. 3.2.

- $P(\tilde{N}|\bar{N}, H, M)$. This term models the spread of points in the Hough space. If all features were detected with exact position, scale and orientation, all correct matches should fall in the same bin, namely the bin with pose parameters closest to those specified by H . Errors in the measurement of features’ location, orientation and scale, as well as model inaccuracy causes these matches to spread in

neighbor bins as well. This term is updated during the Hough transform step described in sec. 3.2.

- $P(V|\tilde{N}, \bar{N}, H, M)$. The assignment vector V specifies in detail which test feature is associated to which model feature or to a clutter detection. Note that geometry and appearance information on the test features (variable F) is not present in this term. Similarly to $P(\bar{N}|H, M)$, only counts of features matches are taken into account. Therefore, this term predicts how many features in the test image, will be eventually put in correspondence with model features, given the initial counts provided by \bar{N} and \tilde{N} .

- $P(F|V, \tilde{N}, \bar{N}, H, M) = P(F|V, H, M)$. This term is most important, as it compares appearance and geometry of each feature with the values predicted by the hypothesis H , once the detailed assignments are known (variable V). The discrepancy in appearance between f_i and $f_{(k,j)}$ is attributed to a combination of measurement noise and modeling error. The discrepancy between the pose of f_i and the one predicted by $f_{(k,j)}$ (together with the pose of model k as described in H) is also explained as an observational/model error. This term and the previous term are computed at each iteration of the RANSAC process described in sec. 3.2

4.2 Progressive score update

As mentioned above, apart from $P(H|M)$ each term in the score decomposition (2) is added by a specific stage of the refinement process described in 3.2.

The values obtained after update by the model voting and the Hough transform step are compared to user-specified thresholds to filter out incorrect hypotheses.

The next score update is performed at each iteration of the RANSAC algorithm: for a given seed triplet, it computes the hypothesis pose, the corresponding assignment vector $V(F, H)$, and updates the score $P(F, H, V|M)$ with the terms $P(V|\tilde{N}, \bar{N}, H, M)$ and $P(F|V, H, M)$.

Test images that contain multiple model objects are handled by adding objects sequentially to the hypothesis. We investigate first the presence of the object that collected the highest number of votes during the model voting step. The Hough transform and RANSAC are performed on the bins related to this object. Once the best hypothesis related to this object is identified, the candidate correspondences identified as ‘good matches’ are discarded, and the search continues with the remaining candidate correspondences and the object that collected the next highest number of votes during model voting. This sequential addition of model ends when all models that collected a significant number of votes (4 votes in sec 6) have been explored.

5 Parameters estimation

5.1 Models for the components of eq.(2)

- $P(H|M)$: the number of models present in the scene can be modeled by a Poisson process. The intensity of this process is the average number of models in test scenes. Since we assume independence between models, this term becomes

$$P(H|M) = P(N_H|M) \cdot \prod_{k \in H} P(\theta_k|M) \quad (3)$$

where $k \in H$ denotes that H specifies model m_k to be present in the scene. H also specifies the pose parameters of this model, θ_k . $P(\theta_k|M)$ is set to a constant when the model pose predicted by H intersects the test image, and to zero otherwise. This is a crude approximation, valid if the models are equally likely to be photographed in any pose. A more elaborate model would make use of contextual information: clouds in a landscape are likely to be detected near the top of the scene, while cars in a street shot are unlikely to be detected upside down.

- $P(\bar{N}|H, M)$ - since we assume independence between models,

$$P(\bar{N}|H, M) = \prod_{k \in H} P(N_k|H, M) \cdot \prod_{k \notin H} P(N_k|H, M) \quad (4)$$

For each model m_k :

1. If the hypothesis H specifies that model m_k is present in the test scene, its features should be detected. In fact, features are more or less stable across changes in viewpoint, lighting condition, etc. [6]. We model the probability of detecting a feature f_j^k from m_k in the test scene, by a constant p_1 independent of the feature and independent of the model. Note that this is an approximation, motivated by the fact that we are learning only from one example. In a more precise description, p_1 should be modeled as depending on f_j^k . On the other hand, p_1 should depend on the pose change specified by H (a feature is more likely to be stable if the change in model pose is small). We model only the dependency on scale change $r_s = s_{scene}/s_{model}$ between model and scene. The number of feature detections is proportional to the image area, therefore we choose $p_1 = K * r_s^2$ if $K * r_s^2 \leq 0.5$, $p_1 = p_1^{max} = 0.5$ otherwise. We take $K = 0.2$, this is consistent with typical numbers of matches obtained by Lowe's algorithm [1] and with features stability results from 5.2.
2. If the hypothesis H specifies that model m_k is absent from the test scene, there is still a small chance p_2 that it appears in the list of model counts N_k . This is due to the inaccuracy of the indexing process between scene features and the database.

The number of features attributed to the clutter is modeled by a Poisson distribution. The mean $\lambda(A)$ of this distribution is proportional to the image

area A , and estimated using a training set of images picked randomly on the internet. In the end, we have

$$P(\bar{N}|H, M) = \prod_{k \in H} \frac{n_k!}{N_k!(n_k - N_k)!} p_1^{N_k} (1 - p_1)^{n_k - N_k} \cdot \prod_{k \in H} \frac{n_k!}{N_k!(n_k - N_k)!} p_2^{N_k} (1 - p_2)^{n_k - N_k} \cdot \text{Pois}(\lambda(A), N_0) \quad (5)$$

where n_k is the total number of features in model m_k , and N_k is the number of times that the indexing process ('model voting', sec. 4.1) returned a feature from model m_k . The binomial coefficients are a consequence of this stage not being concerned with features identities but only with model counts.

- $P(\tilde{N}|\bar{N}, H, M)$ - this density is characterized by the probability densities of the ratios $r_b = N_b/N = N_b/\sum_b N_b$, where b indexes the bins in the discretized Hough space mentioned in sec.3 (the index k has here been omitted, N is the number of model hits from the previous stage - we have $\sum_b N_b = N$, or equivalently $\sum_b r_b = 1$). The densities $P(r_b|H, M)$ were estimated using Monte-Carlo simulations on the ground truth matches obtained with the setup described in sec.5.2. Pairs of candidate matches are formed by appearance-match between two views. Each match hashes into a bin of the discretized Hough space. Bins are indexed relatively to the bin b_0 corresponding to the ground truth transformation between both views. If the matching process was perfectly accurate, all matches would hash into b_0 . In practice, matches are spread in adjacent bins as well. Monte-Carlo simulations form histograms over $[0, 1]$ for the value of r_b in each bin b . For the bins dimensions defined in 3, 48% of the matches hash into b_0 , and bins located further than second-order neighbors of b_0 don't collect a significant number of matches.

- $P(V|\tilde{N}, \bar{N}, H, M)$ - for model m_k present in H , the identity of the Hough space bin of interest is specified by θ_k and denoted by $b(\theta_k)$. Among the candidate correspondences that hashed into this bin, some are correct and some should be rejected. If p_3 is the fraction of correct matches, we can model $P(V|\tilde{N}, \bar{N}, H, M)$ by a binomial distribution. We obtain

$$P(V|\tilde{N}, \bar{N}, H, M) = \prod_{k \in H} \frac{N_{(b(\theta_k), k)}!}{N_k^V!(N_{(b(\theta_k), k)} - N_k^V)!} p_3^{N_k^V} (1 - p_3)^{(N_{(b(\theta_k), k)} - N_k^V)} \quad (6)$$

Note the difference with eq.(5): here the counts of features are specified by the assignment vector V (hence the notations N_k^V), while in (5) they are specified by the vote counts \bar{N} from the Hough transform. Besides, the maximum value allowed for the feature count is the number of features $N_{(b(\theta_k), k)}$ that passed the previous tests, whereas in (5) the maximum feature count was the total number of features in model m_k .

NOTE: is there any difference between the scene features that were considered to be clutter detections at the model voting stage (and modeled by a Poisson distribution) and the features that were rejected only at the last stage (more

complex distribution) ? They are all clutter features after all. But the second sort of clutter features was good enough to survive until the last step.

- $P(F|V, \tilde{N}, \bar{N}, H, M)$. With the conditional feature independence mentioned above,

$$P(F|V, \tilde{N}, \bar{N}, H, M) = \prod_{V(i) \neq 0} p_{fg}(f_i|H, f_{V(i)}) \cdot \prod_{V(i)=0} p_{bg}(f_i) \quad (7)$$

where p_{fg} is the foreground probability of the observed feature’s appearance and pose if the candidate match is correct, whereas p_{bg} is the background probability of its appearance and pose if the feature was actually a clutter detection.

If $V(i) \neq 0$, f_i and $f_{V(i)}$ are believed to be one and the same object part, respectively in the test scene and in a model image. Differences measured between them are an observation noise due to the imaging system as well as distortions caused by viewpoint or lighting conditions changes. This ‘foreground’ probability p_{fg} encodes differences in appearance of the descriptors, but also in geometry, i.e. position, scale, orientation (geometry is omitted e.g. in [12] due to the use of a richer descriptor that captures the object shape). Assuming independence between appearance information and geometry information, denoted respectively by \mathcal{A} and \mathcal{X} , we have

$$p_{fg}(f_i|f_{V(i)}, H) = p_{fg, \mathcal{A}}(\mathcal{A}_i|\mathcal{A}_{V(i)}, H) \cdot p_{fg, \mathcal{X}}(\mathcal{X}_i|\mathcal{X}_{(k,j)}, H) \quad (8)$$

The error in appearance is measured by comparing the appearance descriptors of the scene and model features. We model the density $p_{fg, \mathcal{A}}(\mathcal{A}_i|\mathcal{A}_{V(i)}, H)$ by a full covariance Gaussian distribution. The parameters of this distribution are learned from statistics on pairs of ground truth features formed using the calibrated experiments from sec.5.2.

The error in geometry is measured by comparing the position observed in the test image, with the predicted value that would be observed if the model feature was to be transformed according to the pose parameters specified by H . We model the density $p_{fg, \mathcal{X}}(\mathcal{X}_i|\mathcal{X}_{V(i)}, H)$ on this error by the product of Gaussian distributions for position, orientation and scale, the parameters of these are also learned from statistics on ground truth matches.

If $V(i) = 0$, f_i is believed to be a clutter detection. Similar appearance and location densities $p_{bg, \mathcal{A}}$ and $p_{bg, \mathcal{X}}$ for the features associated to clutter are estimated, this time from random images.

As mentioned in sec. 4, we accept or reject matches in a candidate assignment vector based on the likelihood ratio of the match being correct, versus the feature being a clutter detection.

$$LR_i = \frac{p_{fg, \mathcal{A}}(\mathcal{A}_i|\mathcal{A}_{(k,j)}, H) \cdot p_{fg, \mathcal{X}}(\mathcal{X}_i|\mathcal{X}_{(k,j)}, H)}{p_{bg, \mathcal{A}}(\mathcal{A}_i) \cdot p_{bg, \mathcal{X}}(\mathcal{X}_i)} \quad (9)$$

It is important to note that the parameters for the densities $p_{fg, \mathcal{A}}$, $p_{fg, \mathcal{X}}$, $p_{bg, \mathcal{A}}$, $p_{bg, \mathcal{X}}$ are shared across features, instead of having one set of parameters for each feature as in [3–5]. This results in an important decrease of the parameters that have to be learned, at a slight cost in the model expressiveness.

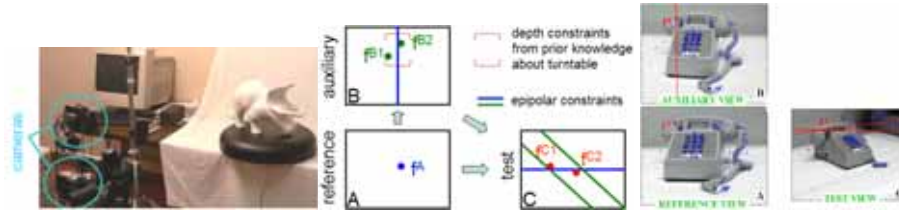


Fig. 2. Experimental 3-view setup for generating ground truth features matches between a reference view and a test view. (left) Setup - the third view is generated when the turntable is rotated. (middle) Diagram for the triple epipolar constraint. Since the viewpoints for A and B are very close, a reliable match for feature f_A can be identified in B using the epipolar constraint from f_A and depth constraints. In image C , the epipolar constraints from images A and B intersect. Statistically, only one or zero features in C satisfy these constraints. c) example of matching process for one feature.

5.2 Ground truth matches

In order to learn the parameters of the foreground and background densities, we need sets of ground truth matches. Besides, we need statistics on how stable features are with respect to viewpoint change. These measurements can be performed by a user clicking on features in pairs of images, but this is tedious. In order to automate the ground truth identification process, we used a stereo rig and a computer-controlled turntable (see fig.2, [6]). Three-dimensional object were photographed from calibrated viewpoints obtained by rotating the turntable. For a triplet of views of a same object, correspondences between features were established using epipolar constraints. First, these ground truth features matches provide statistics on the number of stable features. Second, they also provide information on changes of appearance between two features representing the same object location in two different images. A third measurement is the inaccuracy in feature position, due to the feature detector and the viewpoint change: with the epipolar constraints, for a given feature in a reference view, we can predict its position in other views. The geometric inaccuracy is the deviation from this predicted position, to the actual observed location of the corresponding feature (reprojection error).

One can argue that the lighting conditions for these measurements are different from natural light used in the experiments from sec. 6. However, we believe that the measurements were carried with a range of lighting condition that was wide enough, that the results are useful even when applied to images taken in outdoors environment (ground truth measurements performed with photographic lights with and without diffusers, and neon lights).

6 Experimental results

6.1 Setting

The recognition method presented above was tested on the same datasets used in [2]-sec.4 in order to compare performances. One dataset contains 49 train-

ing and 101 test images of kitchen items and objects of everyday use, against a background of concrete. The background is very grainy and generates lots of clutter detections (more than two thirds of the features detections were background detections). The second set (Ponce Lab, UIUC) contains office pictures, 161 images for the training set and 51 test images. The test scenes contained between zero and five objects from the learning set, for a total of 178 occurrences in the first set and 79 in the second set. Lighting conditions were: sunny outdoors environment for the first set, indoors pictures for the second set. The viewpoint changed significantly between pictures containing a same object.

6.2 Results

The method presented here was compared against the local-to-global approach presented in [2] and Lowe’s voting approach [1]. The implementation of Lowe’s system used for the first set was his own software, as used by Evolution Robotics, while for the second set only our own implementation of his system was available.

Computation time was of the order of 10 seconds per image on a Pentium 4 2.8GHz, which is of the same order of magnitude or a bit lower than the 25s reported in [2]. The computation time of our implementation of D.Lowe’s system is slightly faster (8-9 seconds per image). The limited speed of the current system despite our coarse-to-fine architecture, is due to the number of iterations allowed for the RANSAC step, taken to 1000 in our experiments. In [2], the limiting factor is the number of steps that add one match to the best hypothesis and update its score, while in [1] the speed is determined by the number of bins in the Hough transform.

Fig.3 displays the ROC curves obtained for the 3 methods. The threshold being varied to generate the ROC is the average residual error between the predicted features locations according to the hypothesis, and the positions actually observed.

The detection rate of our method is comparable to the other methods. The objects that were missed are mainly uniform objects, or object with a small footprint. Their coverage by features is too sparse, and the features too unstable, to generate the geometrically consistent combinations required by the RANSAC step.

On the other hand, the false alarm rate is significantly lower than with the methods from [1, 2]. We believe that this is due to the efficiency of the RANSAC procedure at rejecting geometrically inconsistent hypotheses.

Compared to the local-to-global approach from [2], the current method is less prone to false alarms since all the available information is used from the start, while [2] can be misled by incorrect seed matches that appeared very promising in terms of appearance. Compared to Lowe’s method we obtain fewer false alarms as well, as Lowe’s method is a compromise : it trades false alarms for detections via the choice of the bin size in Hough space (smaller bins cause fewer false alarms, while larger bins cause more detections). In our case, the Hough transform used in the second step uses very large bins (to enable high detection

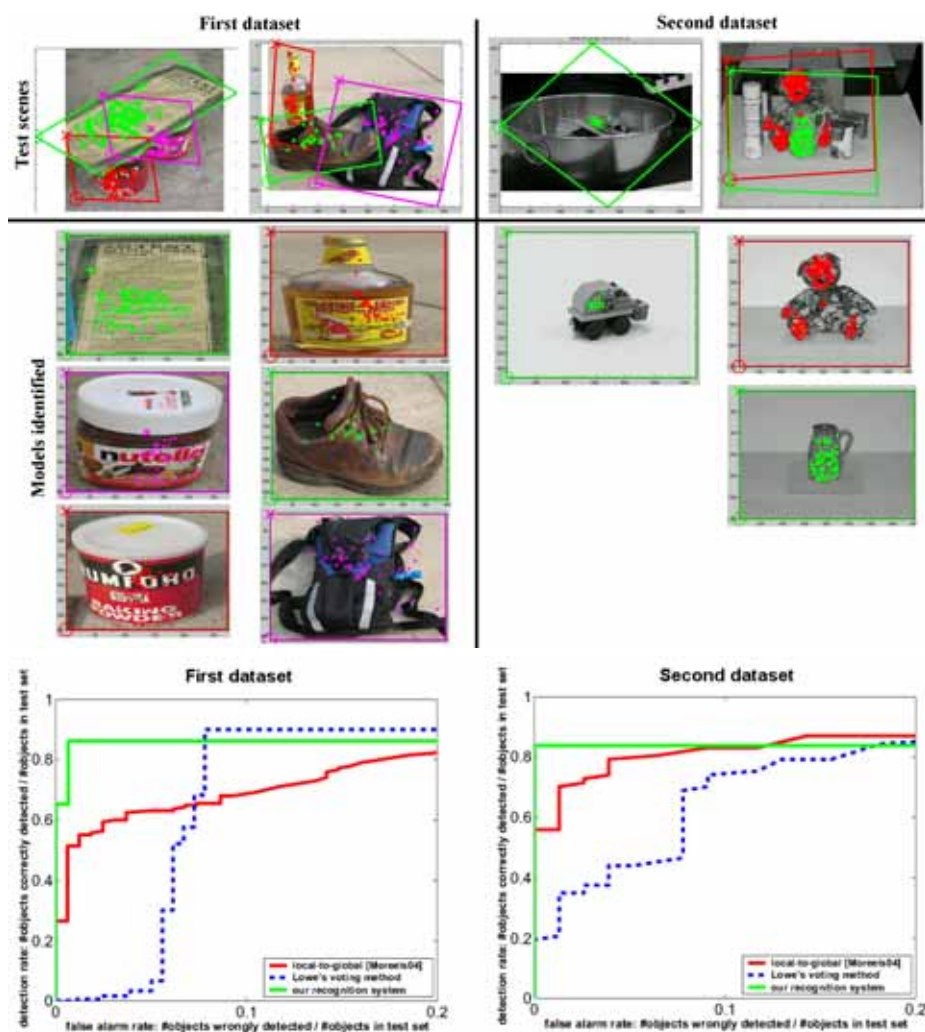


Fig. 3. (Top) A few matching examples from both datasets. Our system shows good robustness to occlusions, viewpoint change and change of scale. Datasets available from www.vision.caltech.edu/ (Bottom) ROC curves for the same two datasets. We compared our method to [2] (red curve) and [1] (blue curve). The system presented here yielded detection rates comparable to the other methods, but significantly lower false-alarm rates.

rates), as we rely only partly on it for false alarm rejections, the remaining false alarms are left to RANSAC to eliminate.

7 Conclusion

We presented a consistent probabilistic framework for individual object recognition. The search for the best interpretation of a given image is performed with a coarse-to-fine strategy. The early stages take into account only global counting variables that are inexpensive to compute. We benefit here from Kd-tree search and Hough transform, which result in first estimates of the objects likely contained in the scene and their pose. A large fraction of irrelevant hypotheses are discarded at a very low computational cost. Further steps refine the hypotheses and specify individual features assignments. At each step the probability of all possible interpretations is updated. The geometric consistency is efficiently enforced by the RANSAC estimator. The search procedure results in a small set of hypotheses whose probability is computed. Besides, the conditional densities used here are estimated using extensive measurements on ground truth matches between images from real 3D objects.

We tested this recognition method against two state-of-the-art systems. The detection rates of the three methods were comparable, but the system presented here showed a significantly lower false-alarm rate.

References

1. D.G. Lowe, "Distinctive Image Features from Scale-Invariant Keypoints", *Int. J. Comp. Vis.*, 60(2), pp. 91-110, 2004.
2. P. Moreels and P. Perona, "Common-Frame Model for Object Recognition", *Proc. NIPS*, 2004.
3. M.C. Burl, M. Weber, P. Perona, "A Probabilistic Approach to Object Recognition Using Local Photometry and Global Geometry", *Proc. Europ. Conf. Comp. Vis.*, pp.628-641, 1998.
4. M. Weber, M. Welling and P. Perona, "Unsupervised Learning of Models for Recognition", *Proc. Europ. Conf. Comp. Vis.*, 2000.
5. R. Fergus, P. Perona, A. Zisserman, "Object Class Recognition by Unsupervised Scale-invariant Learning", *IEEE. Conf. on Comp. Vis. and Patt. Recog.*, 2003.
6. P. Moreels and P. Perona, "Evaluation of Features Detectors and Features Descriptors based on 3D objects", *ICCV*, 2005.
7. F. Fleuret and D. Geman, "Coarse-to-fine face detection", *IJCV*, 41, 85-107, 2001.
8. D. Geman and B. Jedynak, "An active testing model for tracking roads from satellite images", *IEEE Trans. Pattern Anal. Mach. Intell.*, 18, 1-14, 1996.
9. J.S. Beis and D.G. Lowe, "Shape Indexing Using Approximate Nearest-neighbour Search in High-dimensional Spaces", *Proc. IEEE. CVPR*, pp.1000-1006, 1997.
10. D.H. Ballard, "Generalizing the hough transform to detect arbitrary shapes", *Pattern Recognition*, 13(2):111-122, 1981.

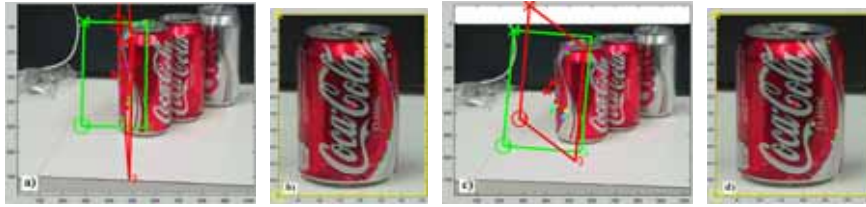


Fig. 4. Gain obtained from using virtual features. a)-b) Test image and identified model image. The estimated affine frame pose (red) is skewed, since all matches (colored dots) lie in a narrow band. Note that if we estimate a similarity transform (green frame) in place of an affine transform, the frame doesn't show this perturbation as similarities capture only one scale factor. However, similarities are less expressive. c)-d) Use of the virtual matches 'stabilizes' the estimated pose.

11. M. Fischler and R. Bolles, "Random sample consensus: a paradigm for model fitting with application to image analysis and automated cartography", *Comm. ACM* 24, 381-395, 1981.
12. S. Belongie, J. Malik, J. Puzicha, "Shape Matching and Object Recognition Using Shape Contexts", *IEEE Trans. Patt. Anal. Mach. Intell.*, 24(24), pp.509-522, 2002.
13. J. Friedman, J. Bentley, R. Finkel, "An Algorithm for Finding Best Matches in Logarithmic Expected Time", *ACM. Trans. Math. Software*, vol. 3, pp. 209-226, 1977.

Appendix - Combining RANSAC with orientation and scale information

RANSAC only makes use of the (x,y) features coordinates, as does D.Lowe's algorithm [1]. However, the features carry orientation and scale information as well. Now, let's remember that a 2D similarity transform can be characterized either by a single match with location, orientation and scale, or by 2 matches containing location only. Therefore, we introduce *virtual matches* to 'convert' orientation&scale information into location information. Starting from a given match between f_i and $f_{(k,j)}$, a virtual test feature g_i is selected randomly in the test scene. The corresponding virtual model feature $g_{(k,j)}$ is obtained by the similarity predicted by the match $(f_i, f_{(k,j)})$. The location information from the pair of matches $\{(f_i, f_{(k,j)}), (g_i, g_{(k,j)})\}$ carries the same pose information as does $(f_i, f_{(k,j)})$ with location, orientation, scale. The virtual pairs are added to the initial candidate matches to produce a richer set before performing RANSAC. One virtual match is added for each 'real' match, so that the role of the orientation and scale information is similar to the role of the location. As shown in Fig.4, the virtual matches are useful to provide regularization when all 'real' candidate matches lie in a thin band, causing triangles formed during RANSAC to be nearly singular.