Simultaneous object recognition and segmentation from single or multiple model views

International Journal of Computer Vision 67(2), 159–188, 2006

DOI: 10.1007/s11263-005-3964-7

Simultaneous Object Recognition and Segmentation from Single or Multiple Model Views*

VITTORIO FERRARI
Computer Vision Group (BIWI), ETH Zuerich, Switzerland
ferrari@vision.ee.ethz.ch

TINNE TUYTELAARS
ESAT-PSI, University of Leuven, Belgium
Tinne.Tuytelaars@esat.kuleuven.ac.be

LUC VAN GOOL
Computer Vision Group (BIWI), ETH Zuerich, Switzerland ; ESAT-PSI, University of Leuven, Belgium
vangool@vision.ee.ethz.ch

Received September 21, 2004; Revised April 4, 2005; Accepted May 3, 2005

First online version published in January, 2006

Abstract. We present a novel Object Recognition approach based on affine invariant regions. It actively counters the problems related to the limited repeatability of the region detectors, and the difficulty of matching, in the presence of large amounts of background clutter and particularly challenging viewing conditions. After producing an initial set of matches, the method gradually explores the surrounding image areas, recursively constructing more and more matching regions, increasingly farther from the initial ones. This process covers the object with matches, and simultaneously separates the correct matches from the wrong ones. Hence, recognition and segmentation are achieved at the same time. The approach includes a mechanism for capturing the relationships between multiple model views and exploiting these for integrating the contributions of the views at recognition time. This is based on an efficient algorithm for partitioning a set of region matches into groups lying on smooth surfaces. Integration is achieved by measuring the consistency of configurations of groups arising from different model views. Experimental results demonstrate the stronger power of the approach in dealing with extensive clutter, dominant occlusion, and large scale and viewpoint changes. Non-rigid deformations are explicitly taken into account, and the approximative contours of the object are produced. All presented techniques can extend any view-point invariant feature extractor.

1. Introduction

The modern trend in Object Recognition has abandoned model-based approaches (e.g. Bebis et al., 1995), which require a 3D model of the object as input, in favor of appearance-based ones, where some example images suffice. Two kinds of appearance-based methods exist: *global* and *local*. Global methods build an object representation by integrating information over an entire image (e.g. Cyr and Kimia, 2001; Murase and Nayar, 1995; Swain and Ballard, 1991), and are therefore very sensitive to background clutter

*This research was supported by EC project VIBES, the Fund for Scientific Research Flanders, and the IST Network of Excellence PASCAL.

and partial occlusion. Hence, global methods only consider test images without background, or necessitate a prior segmentation, a task which has proven extremely difficult. Additionally, robustness to large viewpoint changes is hard to achieve, because the global object appearance varies in a complex and unpredictable way (the object's geometry is unknown). Local methods counter problems due to clutter and occlusion by representing images as a collection of features extracted based on local information only (e.g. Selinger and Nelson, 1999). After the influential work of Schmid (1996), who proposed the use of rotation-invariant features, there has been important evolution. Feature extractors have appeared (Lowe, 2004; Mikolajczyk and Schmid, 2001) which are invariant also under scale changes, and more recently recognition under general viewpoint changes has become possible, thanks to extractors adapting the complete *affine* shape of the feature to the viewing conditions (Baumberg, 2000; Matas et al., 2002; Mikolajczyk and Schmid, 2002; Schaffalitzky and Zisserman, 2002; Tuytelaars et al., 1999; Tuytelaars and Van-Gool, 2000). These *affine invariant* features are particularly significant: even though the global appearance variation of 3D objects is very complex under viewpoint changes, it can be approximated by simple affine transformations on a local scale, where each feature is approximately planar (a *region*). Local invariant features are used in many recent works, and provide the currently most successful paradigm for Object Recognition (e.g. Lowe, 2004; Mikolajczyk and Schmid, 2002; Obrdzalek and Matas, 2002; Rothganger et al., 2005; Tuytelaars and Van-Gool, 2000). In the basic common scheme a number of features are extracted *independently* from both a model and a test image, then characterized by invariant descriptors and finally matched.

In spite of their success, the robustness and generality of these approaches are limited by the repeatability of the feature extraction, and the difficulty of matching correctly, in the presence of large amounts of clutter and challenging viewing conditions. Indeed, large scale or viewpoint changes considerably lower the probability that any given model feature is re-extracted in the test image. Simultaneously, occlusion reduces the number of visible model features. The combined effect is that only a small fraction of model features has a correspondence in the test image. This fraction represents the maximal number of features that can be correctly matched. Unfortunately, at the same time extensive clutter gives rise to a large number of non-object features, which disturb the matching process. As a final outcome of these combined difficulties, only a few, if any, correct matches are produced. Because these often come together with many mismatches, recognition tends to fail.

Even in easier cases, to suit the needs for repeatability in spite of viewpoint changes, only a sparse set of *distinguished* features (Matas et al., 2002) are extracted. As a result, only a small portion of the object is typically covered with matches. Densely covering the visible part of the object is desirable, as it increases the *evidence* for its presence, which results in higher detection power. Moreover, it would allow to find the contours of the object, rather than just its location.

Simultaneous recognition and segmentation. In the first part of the paper we tackle these problems with a new, powerful technique to match a model view to the test image which no longer relies solely on matching viewpoint invariant features. We start by producing an initial large set of unreliable region correspondences, so as to maximize the number of correct matches, at the cost of introducing many mismatches. Additionally, we generate a grid of regions densely covering the model image. The core of the method then iteratively alternates between *expansion* phases and *contraction* phases. Each expansion phase tries to construct regions corresponding to the coverage ones, based on the geometric transformation of nearby existing matches. Contraction phases try to remove incorrect matches, using filters that tolerate non-rigid deformations.

This scheme anchors on the initial matches and then *looks around* them trying to construct more. As new matches arise, they are exploited to construct even more, in a process which gradually *explores* the test image, recursively constructing more and more matches, increasingly farther from the initial ones. At each iteration, the presence of the new matches helps the filter taking better removal decisions. In turn, the cleaner set of matches makes the next expansion more effective. As a result, the number, percentage and extent of correct matches grow with every iteration. The two closely cooperating processes of expansion and contraction gather more evidence about the presence of the object *and* separate correct matches from wrong ones *at the same time*. Hence, they achieve simultaneous recognition and segmentation of the object.

By constructing matches for the coverage regions, the system succeeds in covering also image areas which are not interesting for the feature extractor or not

discriminative enough to be correctly matched by traditional techniques. During the expansion phases, the shape of each new region is adapted to the local surface orientation, allowing the exploration process to follow curved surfaces and deformations (e.g. a folded magazine).

The basic advantage of our approach is that each single correct initial match can expand to cover a smooth surface with *many* correct matches, even when starting from a large number of mismatches. This leads to filling the visible portion of the object with matches. Some interesting direct advantages derive from it. First, robustness to scale, viewpoint, occlusion and clutter are greatly enhanced, because most cases where traditional approaches generate only a few correct matches are now solvable. Secondly, discriminative power is increased, because decisions about the object's identity are based on information densely distributed over the entire portion of the object visible in the test image. Thirdly, the approximate boundary of the object in the test image is suggested by the final set of matches. Fourthly, non-rigid deformations are explicitly taken into account.

Integrating multiple model views. When multiple model views are available, there usually are significant overlaps between the object parts seen by different views. In the second part of the paper, we extend our method to capture the relationships between the model views, and to exploit these for integrating the contributions of the views during recognition. The main ingredient is the novel concept of a *group of aggregated matches* (GAM). A GAM is a set of region matches between two images, which are distributed over a smooth surface of the object. A set of matches, including an *arbitrary* amount of mismatches, can be partitioned into GAMs. The more matches there are in a GAM, the more likely it is that they are correct. Moreover, the matches in a GAM are most often all correct, or all incorrect. When evaluating the correctness and interrelations of sets of matches, it is convenient to reason at the higher perceptual grouping level that GAMs offer: no longer consider unrelated region matches, but the collection of GAMs instead. Hence, GAMs become the atomic unit, with their size carrying precious information. Moreover, the computational complexity of a problem can be reduced, because there are considerably fewer relevant GAMs than region matches.

Concretely, multiple-view integration is achieved as follows. During modeling, the model views are connected by a number of region-tracks. At recognition time, each model view is matched to the test image, and the resulting matches are partitioned into GAMs. The coherence of a configuration of GAMs, possibly originating from different model views, is evaluated using the region tracks that span the model views. We search for the most consistent configuration, covering the object as completely as possible, and define a confidence score which strongly increases in the presence of compatible GAMs. In this fashion, the detection power improves over the simple approach of considering the contribution of each model view independently. Moreover, incorrect GAMs are discovered because they do not belong to the best configuration, thus improving the segmentation.

Paper structure. Sections 2 to 8 cover the first part: the image-exploration technique to match a model view to the test image. The integration of multiple model views is described in the second part, Sections 9 to 12. A discussion of related work can be found in Section 14, while experimental results are given in Section 13. Finally, Section 15 closes the paper with conclusions and possible directions for future research. Preliminary versions of this work have appeared in Ferrari et al. (2004a, b).

2. Overview of Part I: Simultaneous Recognition and Segmentation

Figure 2(a) shows a challenging example, which is used as case-study throughout the first part of the paper. There is a large scale change (factor 3.3), out-of-plane rotation, extensive clutter and partial occlusion. All these factors make the life of the feature extraction and matching algorithms hard.

A scheme of the approach is illustrated in Fig. 1. We build upon a multi-scale extension of the extractor of Tuytelaars and Van-Gool (2000). However, the method works in conjunction with any affine invariant region extractor (Baumberg, 2000; Matas et al., 2002; Mikolajczyk and Schmid, 2002). In the first phase (*soft matching*), we form a large set of initial region correspondences. The goal is to obtain some correct matches

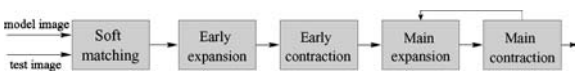


Figure 1. Phases of the image-exploration technique.

also in difficult cases, even at the price of including a large majority of mismatches. Next, a grid of circular regions covering the model image is generated (coined *coverage regions*). The *early expansion* phase tries to propagate these coverage regions based on the geometric transformation of nearby initial matches. By *propagating* a region, we mean constructing the corresponding one in the test image. The propagated matches and the initial ones are then passed through a novel local filter, during the *early contraction* phase, which removes some of the mismatches. The processing continues by alternating faster expansion phases (*main expansion*), where coverage regions are propagated over a larger area, with contraction phases based on a global filter (*main contraction*). This filter exploits both topological arrangements and appearance information, and tolerates *non-rigid deformations*.

The 'early' phases differ from the 'main' phases in that they are specialized to deal with the extremely low percentage of correct matches given by the initial matcher in particularly difficult cases.

3. Soft Matching

The first stage is to compute an initial set of region matches between a *model image* I_m and a *test image* I_t.

The region extraction algorithm (Tuytelaars and Van-Gool, 2000) is applied to both images independently, producing two sets of regions Φ_m, Φ_t, and a vector of invariants describing each region (Tuytelaars and Van-Gool, 2000). Test regions Φ_t are matched to model regions Φ_m in two steps, explained in the next two subsections. The matching procedure allows for *soft matches*, i.e. more than one model region is matched to the same test region, or vice versa.

3.1. Tentative Matches

For each test region $T \in \Phi_t$ we first compute the Mahalanobis distance of the descriptors to all model regions $M \in \Phi_m$. Next, the following appearance similarity measure is computed between T and each of the 10 closest model regions:

$$\overline{\text{sim}}(M,T) = \text{NCC}(M,T) + \left(1 - \frac{\text{d}\overline{\text{RGB}}(M,T)}{100}\right) \tag{1}$$

where NCC is the normalized cross-correlation between the regions' greylevel patterns, while $\text{d}\overline{\text{RGB}}$ is the average pixel-wise Euclidean distance in *RGB* color-space after independent normalization of the 3 colorbands (necessary to achieve photometric invariance). Before computation, the two regions are aligned by the affine transformation mapping T to M. This mixed measure is more discriminative than NCC alone, which is the most common choice in the literature (Obrdzalek and Matas, 2002; Mikolajczyk and Schmid, 2002; Tuytelaars and Van-Gool, 2000). NCC mostly looks at the *pattern structure*, and discards valuable color information. A green disc on a red background, and a bright blue disc on a dark blue background would be very similar under NCC. $\text{d}\overline{\text{RGB}}$ captures complementary properties. As it focuses on *color* correspondence, it would correctly score low the previous disc example. However, it would confuse a green disc on a bright green background with a green cross on a bright green background, a difference which NCC would spot. By summing these two measures, we obtain a more robust one which alleviates their complementary shortcomings.

Each of the 3 test regions most similar to T above a low threshold t_1 are considered tentatively matched to T. Repeating this operation for all regions $T \in \Phi_t$, yields a first set of *tentative matches*. At this point, every test region could be matched to either none, 1, 2 or 3 model regions.

3.2. Refinement and Re-Thresholding

Since all regions are independently extracted from the two images, the geometric registration of a correct match is often not optimal. Two matching regions often do not cover exactly the same physical surface, which lowers their similarity. The registration of the tentative matches is now *refined* using our algorithm (Ferrari et al., 2003), that efficiently looks for the affine transformation that maximizes the similarity. This results in adjusting the region's location and shape in one of the images. Besides raising the similarity of correct matches, this improves the quality of the forthcoming *expansion* stage, where new matches are constructed based on the affine transformation of the initial ones.

After refinement, the similarity is re-evaluated and only matches scoring above a second, higher threshold t_2 are kept.[1] Refinement tends to raise the similarity of correct matches much more than that of mismatches. The increased *separation* between the similarity distributions makes the second thresholding more effective. At this point, about 1/3 to 1/2 of the tentative matches are left.

3.3. Motivation

The obtained set of matches usually still contains *soft matches*, i.e. more than one region in Φ_m is matched to the same region in Φ_t, or vice versa. This contrasts with previous works (Baumberg, 2000; Lowe, 2004; Mikolajczyk and Schmid, 2002; Obrdzalek and Matas, 2002; Tuytelaars and Van-Gool, 2000), but there are two good reasons for it. First, the scene might contain repeated, or visually similar elements. Secondly, large viewpoint and scale changes cause loss of resolution which results in a less accurate geometric correspondence and a lower similarity. When there is also extensive clutter, it might be impossible, based *purely* on local appearance (Schaffalitzky and Zisserman, 2002), to decide which of the best 3 matches is correct, as several competing regions might appear very similar, and score higher than the correct match. A classic 1-to-1 approach may easily be distracted and fail to produce the correct match.

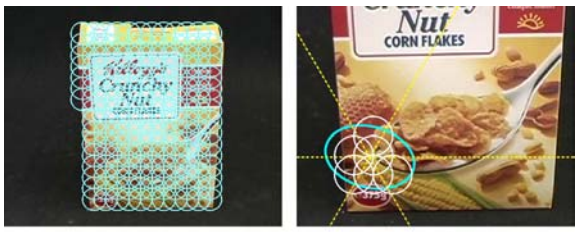
The proposed process outputs a large set of plausible matches, all with a reasonably high similarity. The goal is to maximize the number of correct matches, even at the cost of accepting a substantial fraction of mismatches. This is important in difficult cases, when only a few model regions are re-extracted in the test image, because each correct match can start an expansion which will cover significant parts of the object.

Figure 2(a) shows the case-study, for which 3 correct matches out of 217 are found (a *correct-ratio* of 3/217). The large scale change, combined with the modest resolution (720×576), causes heavy image degradation which corrupts edges and texture. In such conditions only a few model regions are re-extracted in the test image and many mismatches are inevitable. In the rest of the paper, we refer to the current set of matches as the *configuration* Γ.

How to proceed? Global, robust geometry filtering methods, like detecting outliers to the epipolar geometry through RANSAC (Torr and Murray, 1997) fail, as they need a minimal portion of inliers of about 1/3 (Chum et al., 2003; Lowe, 2004). Initially, this may very well not be the case. Even if we could separate out the few correct matches, they would probably not be sufficient to draw reliable conclusions about the presence of the object. In the following sections, we explain how to gradually increment the number of correct matches and simultaneously decrease the number of mismatches.

a b

Figure 2. (a) Case-study, with model image (*top*), and test image (*bottom*). (b) A close-up with 3 initial matches. The two model regions on the left are both matched to the same region in the test image. Note the small occluding rubber on the spoon.



a b

Figure 3. (a) The homogeneous coverage Ω. (b) A support region (*dark*), associated sectors (*lines*) and candidates (*bright*).

4. Early Expansion

4.1. Coverage of the Model Image

We generate a grid Ω of overlapping circular regions densely covering the model image I_m (Fig. 3(a)). In our implementation the grid is composed of a first layer of regions of radius 25 pixels, spaced 25 pixels, and a second layer with radius 13 pixels and spaced 25 pixels.[2] No regions are generated on the black background. According to various experiments, this choice of the parameters is not crucial for the overall recognition performance. The choice of the exact grid pattern, and consequently the number of regions in Ω, trades segmentation quality for computational cost, and could be selected based on the user's desires.

At this point, none of the regions in Ω is matched to the test image I_t. The expansion phases will try to

construct in I_t as many regions corresponding to them as possible.

4.2. Propagation Attempt

We now define the concept of *propagation attempt* which is the basic building-block of the expansion phases and will be used later. Consider a region C_m in model image I_m without match in the test image I_t and a nearby region S_m, matched to S_t. If C_m and S_m lie on the same physical facet of the object, they will be mapped to I_t by similar affine transformations. The *support* match (S_m, S_t) *attempts to propagate* the *candidate* region C_m to I_t as follows:

1. Compute the affine transformation A mapping S_m to S_t.
2. Project C_m to I_t via A: $C_t = AC_m$.

The benefits of exploiting previously established geometric transformations was also noted by Schaffalitzky and Zisserman (2002).

4.3. Early Expansion

Propagation attempts are used as a basis for the first expansion phase as follows. Consider as supports $\{S^i = (S^i_m, S^i_t)\}$ the soft-matches configuration Γ, and as candidates Λ the coverage regions Ω. For each support region S^i_m we partition I_m into 6 circular sectors centered on the center of S^i_m (Fig. 3(b)).

Each S^i_m attempts to propagate the closest candidate region in each sector. As a consequence, each candidate C_m has an associated subset $\Gamma_{C_m} \subset \Gamma$ of supports that will *compete* to propagate it. For a candidate C_m and each support S^i in Γ_{C_m} do:

1. Generate C^i_t by attempting to propagate C_m via S^i.
2. Refine C^i_t. If C^i_t correctly matches C_m, this adapts it to the local surface orientation (handles curved and deformable objects) and perspective effects (the affine approximation is only valid on a local scale).
3. Compute the color transformation $T^i_{RGB} = \{s_R, s_G, s_B\}$ between S^i_m and S^i_t. This is specified by the scale factors on the three colorbands.
4. Evaluate the quality of the refined propagation attempt, after applying the color transformation T^i_{RGB}

$$sim_i = sim\big(C_m, C^i_t, T^i_{RGB}\big) = \mathrm{NCC}\big(T^i_{RGB}C_m, C^i_t\big) + \left(1 - \frac{\mathrm{dRGB}(T^i_{RGB}C_m, C^i_t)}{100}\right)$$

Applying T^i_{RGB} allows to use the unnormalized similarity measure sim, because color changes are now compensated for. This provides more discriminative power over using $\overline{\mathrm{sim}}$.

We retain C^{best}_t, with $best = \arg\max_i sim_i$, the best refined propagation attempt. C_m is considered successfully propagated to C^{best}_t if $sim_{\mathrm{best}} > t_2$ (the matching threshold). This procedure is applied for all candidates $C_m \in \Lambda$.

Most support matches may actually be mismatches, and many of them typically lie around each of the few correct ones (e.g. several matches in a single soft-match, Fig. 2(b)). In order to cope with this situation, each support concentrates its efforts on the nearest candidate in each direction, as it has the highest chance to undergo a similar geometric transformation. Additionally, every propagation attempt is refined before evaluation. Refinement raises the similarity of correctly propagated matches much more than the similarity of mispropagated ones, thereby helping correct supports to win. This results in a limited, but controlled growth, maximizing the chance that each correct match propagates, and limiting the proliferation of mispropagations. The process also restricts the number of refinements to at most 6 per support (contains computational cost).

For the case-study, 113 new matches are generated and added to the configuration Γ. 17 of them are correct and located around the initial 3 (Fig. 4(a)). The correct-ratio of Γ improves to 20/330 (Fig. 4(b)), but it is still very low.

5. Early Contraction

The early expansion guarantees good chances that each initial correct match propagates. As initial filter, we discard all matches that did not succeed in propagating any region. The correct-ratio of the case-study improves to 20/175 (no correct match is lost), but it is still too low for applying a global filter. Hence, we developed the following local filter.

Figure 4. (a) Early propagation generates 17 correct matches (*bright*) out of 113. These are located around the initial 3 correct matches (*dark*). (b) The configuration after early expansion has 20 correct matches (*bright*) and 310 mismatches (*dark*).

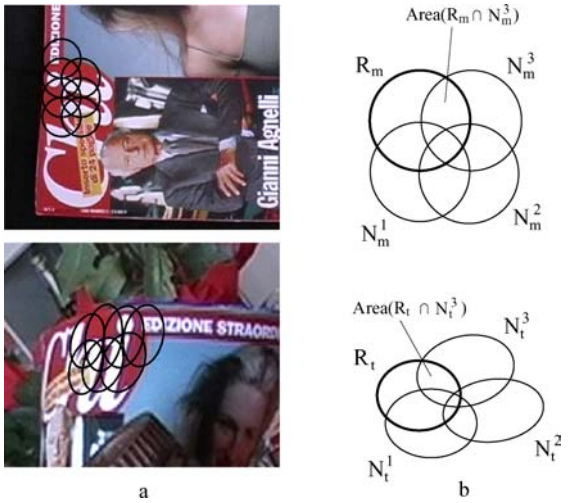


Figure 5. Surface contiguity filter. (a) The pattern of intersection between neighboring correct region matches is preserved by transformations between the model and the test images, because the surface is contiguous and smooth. (b) The filter evaluates this property by testing the conservation of the area ratios.

A local group of regions in the model image have uniform shape, are arranged on a grid and intersect each other with a specific pattern. If all these regions are correctly matched, the same regularities also appear in the test image, because the surface is contiguous and smooth (regions at depth discontinuities cannot be correctly matched anyway). This holds for curved or deformed objects as well, because the affine transformation varies slowly and smoothly across neighboring regions (Fig. 5(a)). On the other hand, mismatches tend to be randomly located over the image and to have different shapes.

We propose a novel local filter based on this observation. Let $\{N_m^i\}$ be the neighbors of a region R_m in the model image. Two regions A, B are considered neighbors if they intersect, i.e. if $\text{Area}(A \cap B) > 0$. Only neighbors which are actually matched to the test image are considered. Any match (R_m, R_t) is removed from Γ if

$$\sum_{\{N_m^i\}} \left| \frac{\text{Area}(R_m \cap N_m^i)}{\text{Area}(R_m)} - \frac{\text{Area}(R_t \cap N_t^i)}{\text{Area}(R_t)} \right| > t_s \quad (2)$$

with t_s some threshold.[3] The filter, illustrated in Fig. 5(b), tests the preservation of the pattern of intersections between R and its neighbors (the ratio of areas is affine invariant). Hence, a removal decision is based solely on *local* information. As a consequence, this filter is unaffected by the current, low overall ratio of correct matches.

Shape information is integrated in the filter, making it capable of spotting insidious mismatches which are roughly correctly located, yet have a wrong shape. This is an advantage over the (semi-) local filter proposed by (Schmid, 1996), and later also used by others (Schaffalitzky and Zisserman, 2002; Sivic and Zisserman, 2003), which verifies if a minimal amount of regions in an area around R_m in the model image also match near R_t in the test image.

The input regions need not be arranged in a regular grid, the filter applies to a general set of (intersecting) regions. Note that isolated mismatches, which have no neighbors in the model image, will not be detected. The algorithm can be implemented to run in $O((|\Gamma|+x)\log(|\Gamma|))$, with $x \ll |\Gamma|^2$ the number of region intersections (Ferrari, 2004, pp. 202–203).

Applying this filter to the case-study brings the correct-ratio of Γ to 13/58, thereby greatly reducing the number of mismatches.

6. Main Expansion

The first early expansion and contraction phases brought several additional correct matches and removed many mismatches, especially those that

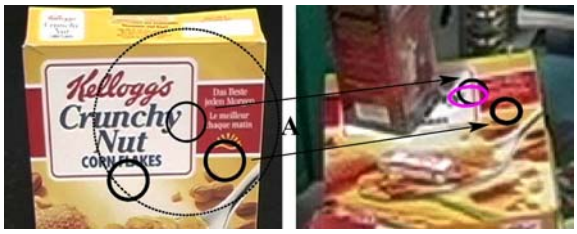


Figure 6. *Left*: a candidate (*thin*) and 2 of 20 supports within the large circular area. *Right*: the candidate is propagated to the test image using the affine transformation A of the support on the right (*thick*). Refinement adapts the shape to the perspective effects (*brighter*). The other support is mismatched to a region not visible in this close-up.

concentrated around the correct ones. Since Γ is cleaner, we can now try a faster expansion.

All matches in the current configuration Γ are removed from the candidate set $\Lambda \leftarrow \Lambda \backslash \Gamma$, and are used as supports. All support regions S_m^i in a circular area[4] around a candidate C_m compete to propagate it:

1. Generate C_t^i by attempting to propagate C_m via S^i.
2. Compute the color transformation T_{RGB}^i of S^i.
3. Evaluate $sim_i = \text{sim}(C_m, C_t^i, T_{RGB}^i)$.

We retain C_t^{best}, with $best = \arg\max_i sim_i$ and refine it, yielding C_t^{ref}. C_m is considered successfully propagated to C_t^{ref} if $\text{sim}(C_m, C_t^{\text{ref}}) > t_2$ (Fig. 6). This scheme is applied for each candidate.

In contrast to the early expansion, many more supports compete for the same candidate, and no refinement is applied *before* choosing the winner. However, the presence of more correct supports, now tending to be grouped, and fewer mismatches, typically spread out, provides good chances that *a* correct support will win a competition. In this process each support has the chance to propagate many more candidates, spread over a larger area, because it offers help to all candidates within a wide circular radius. This allows the system to grow a *mass* of correct matches. Moreover, the process can jump over small occlusions or degraded areas, and costs only one refinement per candidate. For the case-study, 185 new matches, 61 correct, are produced, thus lifting the correct-ratio of Γ up to 74/243 (31%, Fig. 9, second row).

7. Main Contraction

At this point the chances of having a sufficient number of correct matches for applying a global filter are much better. We propose here a global filter based on a topological constraint for triples of region matches. In contrast to the local filter of Section 5, this filter is capable of finding also isolated mismatches. The next subsection introduces the property on which the filter is based, while the following two subsections explain the filter itself and discuss its qualities.

7.1. The Sidedness Constraint

Consider a triple (R_m^1, R_m^2, R_m^3) of regions in the model image and their matching regions (R_t^1, R_t^2, R_t^3) in the test image. Let $\mathbf{c_v^j}$ be the center of region R_v^j ($v \in \{m, t\}$). The function

$$\text{side}\big(R_v^1, R_v^2, R_v^3\big) = \text{sign}\big(\big(\mathbf{c_v^2} \times \mathbf{c_v^3}\big)\mathbf{c_v^1}\big) \qquad (3)$$

takes value -1 if c_v^1 is on the right side of the directed line $\mathbf{c_v^2} \times \mathbf{c_v^3}$, going from c_v^2 to c_v^3, or value 1 if it's on the left side. The equation

$$\text{side}\big(R_m^1, R_m^2, R_m^3\big) = \text{side}\big(R_t^1, R_t^2, R_t^3\big) \qquad (4)$$

states that $\mathbf{c}^1$ should be on the same side of the line in both views (Fig. 7). This *sidedness constraint* holds for all correctly matched triples of coplanar regions, because in this case property (3) is viewpoint invariant. The constraint is valid also for most non-coplanar triples. A triple violates the constraint if at least one of the three regions is mismatched, or if they are not coplanar and there is important camera translation in the direction perpendicular to the 3D plane containing their centers (*parallax-violation*). This can create a parallax effect strong enough to move $\mathbf{c}^1$ to the other side of the line. Nevertheless, this phenomenon typically affects only a small minority of triples. Since the camera can only translate in one direction between two views, the resulting parallax can only corrupt few triples, because those on planes oriented differently will not be affected.

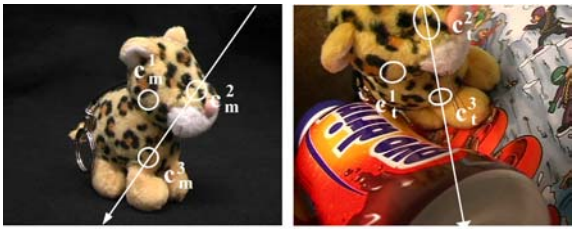


Figure 7. Sidedness constraint. $\mathbf{c}^1$ should be on the same side of the directed line from $\mathbf{c}^2$ to $\mathbf{c}^3$ in both images.

The region matches violate or respect Eq. (4) independently of the order in which they appear in the triple. The three points should be cyclically ordered in the same orientation (clockwise or anti-clockwise) in the two images in order to satisfy (4).

Topological configurations of points and lines were also used by Tell and Carlsson (2002), in the wide-baseline stereo context, as a mean for guiding the matching process.

7.2. *Topological Filter*

A triple including a mismatched region has higher chances to violate the sidedness constraint. When this happens, it indicates that probably at least one of the matches is incorrect, but it does not tell which one(s). While one triple is not enough to decide, this information can be recovered by considering all triples simultaneously. By integrating the weak information each triple provides, it is possible to robustly discover mismatches. The key idea is that we expect incorrectly located regions to be involved in a higher share of violations.

The constraint is checked for all unordered triples (R^i, R^j, R^k), $R^i, R^j, R^k \in \Gamma$. The share of violations for a region match R^i is $\mathrm{err}_{\mathrm{topo}}(R^i) =$

$$\frac{1}{v} \sum_{R^j, R^k \in \Gamma \setminus R^i,\, j>k} \left|\mathrm{side}\left(R^i_m, R^j_m, R^k_m\right) - \mathrm{side}\left(R^i_t, R^j_t, R^k_t\right)\right| \tag{5}$$

with $v = (n-1)\,(n-2)/2$, $n = |\Gamma|$. $\mathrm{err}_{\mathrm{topo}}(R^i) \in [0, 1]$ because it is normalized w.r.t. the maximum number of violations v any region can be involved in.

The topological error share (5) is combined with an appearance term, giving the total error

$$\mathrm{err}_{\mathrm{tot}}(R^i) = \mathrm{err}_{\mathrm{topo}}(R^i) + \left(t_2 - \overline{\mathrm{sim}}\left(R^i_m, R^i_t\right)\right)$$

The filtering algorithm starts from the current set of matches Γ, and then iteratively removes one match at a time as follows:

1. (Re-)compute $\mathrm{err}_{\mathrm{tot}}(R^i)$ for all $R^i \in \Gamma$.
2. Find the worst match R^w, with $w = \arg\max_i \mathrm{err}_{\mathrm{IFtot}}(R^i)$
3. If $\mathrm{err}_{\mathrm{tot}}(R^w) > 0$, remove R^w from Γ. R^w will not be used for the computation of $\mathrm{err}_{\mathrm{topo}}$ in the next iteration. Iterate to 1.
   If $\mathrm{err}_{\mathrm{tot}}(R^w) \leq 0$, or if all matches have been removed, then stop.

At each iteration the most probable mismatch R^w is removed. During the first iterations several mismatches are still present. Therefore, even correct matches might have a moderately large error, as they take part in triples including mismatches. However, mismatches are likely to have an even larger error, because they are involved in the very same triples, plus other violating ones. Hence, the worst mismatch R^w, the region located in I_t farthest from where it should be, is expected to have the largest error. After removing R^w all errors decrease, including the errors of correct matches, because they are involved in less triples containing a mismatch. After several iterations, ideally only correct matches are left. Since these have only a low error, due to occasional parallax-violations, the algorithm stops.

The second term of $\mathrm{err}_{\mathrm{tot}}$ decreases with increasing appearance similarity, and it vanishes when $\mathrm{sim}(R^i_m, R^i_t) = t_2$, the matches acceptance threshold. The removal criterion $\mathrm{err}_{\mathrm{tot}} > 0$ expresses the idea that topological violations are accepted up to the degree to which they are compensated by high similarity. This helps finding mismatches which can hardly be judged by only one cue. A typical mismatch with similarity just above t_2, will be removed unless it is perfectly topologically located. Conversely, correct matches with $\mathrm{err}_{\mathrm{topo}} > 0$ due to parallax-violations are in little danger, because they typically have good similarity. Including appearance makes the filter more robust to low correct-ratios, and remedies the potential drawback (parallax-violations) of a purely topological filter.

In order to achieve good computational performance, we store the terms of the sum in function (5) during the first iteration. In the following iterations, the sum is quickly recomputed by retrieving and adding up the necessary terms. This makes the computational cost almost independent of the number of iterations. The algorithm can be implemented to run in $O(n^2 \log(n))$, based on the idea of constructing, for each point, a list with a cyclic ordering of all other points (a complete explanation is given in Ferrari (2004, pp. 208–211).

7.3. *Properties and Advantages*

The proposed filter has various attractive properties, and offers several advantages over detecting outliers to the epipolar geometry through RANSAC (Torr and Murray, 1997), which is traditionally used in the matching literature (Matas et al., 2002; Mikolajczyk and Schmid, 2002; Schaffalitzky and Zisserman, 2002a, 2002b; Tuytelaars and Van-Gool, 2000). In the

Figure 8. Sidedness constraints hold also for deformed objects. The small arrows indicate 'to the right' of the directed lines $A \rightarrow B$, $B \rightarrow C$, $C \rightarrow D$, $D \rightarrow A$.

following, we refer to it as RANSAC-EG. The main two advantages are (more discussion in Ferrari (2004, pp. 75–77):

It allows for non-rigid deformations. The filter allows for non-rigid deformations, like the bending of paper of cloth, because the structure of the spatial arrangements, captured by the sidedness constraints, is stable under these transformations. As Fig. 8 shows, sidedness constraints are still respected even in the presence of substantial deformations. Other filters, which measure a geometrical distance error from an estimated model (e.g. homography, fundamental matrix) would fail in this situation. In the best case, several correct matches would be lost. Worse yet, in many cases the deformations would disturb the estimation of the model parameters, resulting in a largely random behavior. The proposed filter does not try to capture the transformations of all matches in a single, overall model, but it relies instead on simpler, weak properties, involving only three matches each. The discriminative power is then obtained by integrating over all measurements, revealing their strong, collective information.

It is insensitive to inaccurate locations. The regions' centers need not be exactly localized, because err_{topo} varies slowly and smoothly for a region departing from its ideal location. Hence, the algorithm is not affected by perturbations of the region's locations. This is precious in the presence of large scale changes, not completely planar regions, or with all kinds of image degradation (motion blur, etc.), where localization errors become more important. In RANSAC-EG instead, the point must lie within a tight band around the epipolar line. Worse yet, inaccurate localization of some regions might compromise the quality of the fundamental matrix, and therefore even cause rejection of many accurate regions (Zhang et al., 1995). In Ferrari (2004, pp. 84–85) we report experiments supporting this point, where the topological filter could withstand large random shifts on the regions' locations (about 25 pixels, in a 720×576 image).

7.4. Main Contraction on the Case-Study

After the main expansion, the correct-ratio of the case-study was of 74/243. Applying the filter presented in this section brings it to 54/74, which is a major improvement (Fig. 9 second row). 20 correct matches are lost, but many more mismatches are removed (149). The further processing will recover the correct matches lost and generate even more.

8. Exploring the Test Image

The processing continues by iteratively alternating main expansion and main contraction phases.

1. Do a main expansion phase. All current matches Γ are used as supports. This produces a set of propagated region matches Υ, which are added to the configuration: $\Gamma \leftarrow (\Gamma \cup \Upsilon)$.
2. Do a main contraction phase on Γ. This removes matches from Γ.
3. If at least one newly propagated region survives the contraction, i.e. if $|\Upsilon \cap \Gamma| > 0$, then iterate to point 1, after updating the candidate set to contain $\Lambda \leftarrow (\Omega \backslash \Gamma)$, all original candidate regions Ω which are not yet in the configuration. Stop if no newly propagated regions survived, or if all regions Ω have been propagated (i.e. if $\Omega \subset \Gamma$).

In the first iteration, the expansion phase generates some correct matches, along with some mismatches. Because a correct match tends to propagate more than a mismatch, the correct ratio increases. The first main contraction phase removes mostly mismatches, but might also lose several correct matches: the amount of noise (percentage of mismatches) could still be high and limit the filter's performance. In the next iteration, this cleaner configuration is fed into the expansion phase again which, less distracted, generates more correct matches and fewer mismatches. The new correct matches in turn help the next contraction stage in taking better removal decisions, and so on. As a result, the number, percentage and spatial extent of correct matches increase at every iteration, reinforcing

Figure 9. Evolution of Γ for the case-study. Top-rows: correct matches; bottom rows: mismatches.

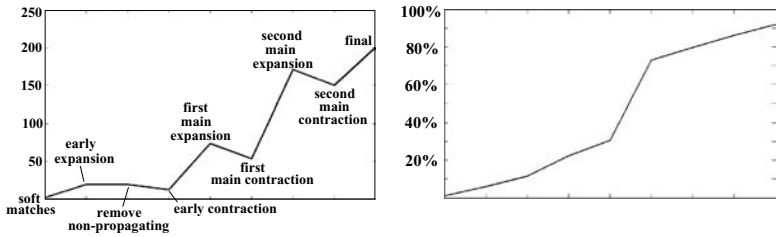


Figure 10. *Left*: the number of correct matches for the case-study increases at every iteration (compare the points after each contraction phase). *Right*: the steady growth in the percentage of correct matches best illustrates the increasing confidence in the presence of the object (from 1.4% after soft-matching, to 91.8% after the last iteration!).

the confidence about the object's presence and location (Fig. 10). The two goals of separating correct matches and gathering more information about the object are achieved *at the same time*.

Correct matches erroneously killed by the contraction step in an iteration get another chance during the next expansion phase. With even fewer mismatches present, they are probably regenerated, and this time have higher chances to survive the contraction (higher correct-ratio, more positive evidence present).

Thanks to the refinement, each expansion phase adapts the shape of the newly created regions to the local surface orientation. Thus the whole exploration process follows curved surfaces and deformations.

The exploration procedure tends to 'implode' when the object is not in the test image, typically returning only a few matches. Conversely, when the object is present, the approach fills the visible portion of the object with many high confidence matches. This yields high discriminative power and the qualitative shift from only *detecting* the object to knowing its extent in the image and which parts are occluded. Recognition and segmentation are two aspects of the *same* process.

In the case-study, the second main expansion propagates 141 matches, 117 correct, which is better than the previous 61/185. The second main contraction starts from 171/215 and returns 150/174, killing a lower percentage of correct matches than in the first iteration. After the 11th iteration 220 matches cover the whole visible part of the object (202 are correct). Figure 9 depicts the evolution of the set of matches Γ. The correct matches gradually cover more and more of the object, while mismatches decrease in number. The system reversed the situation, by going from only very few correct matches in a large majority of mismatches, to hundreds of correct matches with only a few mismatches. Notice the accuracy of the final segmentation, and in particular how the small occluding rubber has been correctly left out (Fig. 9 bottom-right).

9. Overview of Part II: Integrating Multiple Model Views

The image-exploration technique presented in the first part of the paper matches each single model view to the test image independently. In this second part, we capture the relationships among multiple model views, and integrate their contributions at recognition time.

In the next section, we introduce an algorithm for partitioning a set of region matches between two images into groups lying on smooth surfaces (termed *groups of aggregated matches*, or GAMs). GAMs are at the heart of the approach, and enjoy two fundamental properties. First, the matches in a GAM are most often all correct, or all incorrect. Second, it is very unlikely for mismatches to form large GAMs (i.e. composed of many matches). Hence, the size of a GAM informs about the probability of it being correct. Because of these properties, it is convenient to reason in terms of GAMs, rather than individual matches. Our multiple view integration scheme relates GAMs arising from different model views, and considers *them* as atomic units, without descending to the matches level.

Sections 11 and 12 present the multiple-view integration approach. In the initial modeling stage, the model views are matched to each other, in order to build a large number of *region-tracks*, densely connecting them (Section 11). At recognition time, we match each model view to the test image and partition the resulting sets of matches into GAMs (Section 12). By following the model tracks, a GAM originating from a certain model view can be transfered to another model view.

Hence, we can measure the geometric consistencies of pairs of GAMs, and integrate these into a global score which quantifies the goodness of some subset (*configuration*) of all GAMs, even if they originate from different model views. We search for the configuration that maximizes the score function. The maximal score represents the system's confidence in the presence of the object and strongly increases in the presence of compatible GAMs. Therefore, the detection power is better than when considering model views in isolation, and the segmentation improves because several incorrect GAMs are typically left out of the best configuration.

10. Groups of Aggregated Matches (GAMs)

This section describes an incremental grouping algorithm to partition a set of two-view matches into GAMs.

10.1. Affine Dissimilarity

The grouping process is driven by the similarity between the affine transformations that map the regions from one view to the other. Consider three points on each region: the center p_0 and two more points p_1, p_2 on the boundary. These points have previously been put in correspondence by the matching algorithm. The following function measures to which degree the affine transformation of a region match R is also valid for another match Q (Fig. 11):

$$D(R, Q) = \frac{1}{6}\left(\sum_{i=0..2} \left\| A^R_{1,2} Q^i_1 - Q^i_2 \right\| + \sum_{i=0..2} \left\| A^R_{2,1} Q^i_2 - Q^i_1 \right\| \right) \qquad (6)$$

where $A^R_{a,b}$ is the affine transformation mapping R from view a to view b, and R^i_v is point p_i of region R in view v. By averaging over the two regions, we obtain the *affine dissimilarity*

$$D_A(R, Q) = \frac{1}{2}(\mathrm{D}(R, Q) + \mathrm{D}(Q, R)) \qquad (7)$$

between (the affine transformations of) R and Q. This measure is symmetric in the regions *and* in the views. This brings stability and helps dealing fairly with large scale changes. Two region matches have a high affine dissimilarity if either is a mismatch, or if they lie on different surfaces.

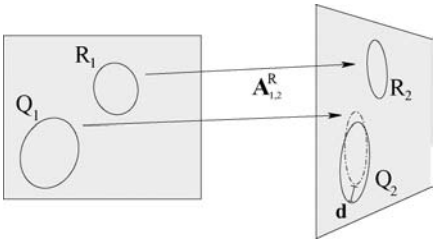


Figure 11. Affine dissimilarity. **d** is one term in function (6).

10.2. Constructing GAMs

The matches are partitioned by the following algorithm, which starts a GAM from a single match and then grows it by iteratively adding matches. The algorithm starts with the set Ω of region matches.

1. A match is removed from Ω and put in a new GAM Γ.
2. Search Ω for a region with affine dissimilarity to the GAM below a certain threshold.The search proceeds from the closest to the farthest to the GAM, according to the spatial distance

$$\frac{\sum_{R\in\Gamma} \mathrm{d}\left(R^0_1, Q^0_1\right)}{|\Gamma|}$$

   This is the average Euclidean distance (d) of a region Q to the regions composing the GAM, measured in the first view. The affine dissimilarity between a region Q and the GAM Γ is $\sum_{R\in\Gamma} w_R D_A(R, Q)$. This is the weighted mean of the affine dissimilarities to each region in the GAM, with weights w_R set inversely proportional to the square of the distances between the regions.
3. As soon as a suitable region is found, it is added to the GAM and the search stops. The region is removed from Ω, and the algorithm iterates to 2. If no such region is found, the current GAM is closed. The algorithm goes back to 1, where a new GAM is created and then grown. The process terminates when Ω is empty.

Figure 12 shows an example run (*Felix*). Matches A, B, C, D, E, F are distributed over the curved magazine surface, while G, I, J over the planar plate on the left

Figure 12. Felix scene. Top: 9 Matches. Bottom: Close-up on match H; the 'a' of 'Happy' is mismatched to 'Birthday'. The GAM constructor successfully finds the two groups (dish, magazine) and isolates the mismatch in a third, singleton one.

of the image. Region H, covering the 'a' of 'Happy' in the left image, is mismatched to the 'a' of 'Birthday' in the right image (the correct corresponding region is not visible). The algorithm starts by creating a GAM containing region A alone. In the next iteration, the nearest region B is added to the GAM, and then C, D, E, F are added one at the time, in this order. No other region has a sufficiently similar affine transformation, so the GAM $\{A, B, C, D, E, F\}$ is closed. A new GAM formed by region G is started, and then region I is added. The next nearest region H is a mismatch and has a quite dissimilar affine transformation, so it doesn't join the GAM in the second iteration. Instead, J is picked up, and the GAM is closed as $\{G, I, J\}$. Finally, H is put in a singleton GAM, and the algorithm terminates.

The algorithm groups two regions in the same GAM if they have a similar affine transformation or if there is some region with coherent intermediate affine transformation spatially located between them. In other words, the affine transformation can vary gradually from a region to the next within a GAM. Hence, a GAM can cover not only a planar, but also a curved or even a continuously deformed surface (like bending of paper or cloth). The fact that the method doesn't prescribe a fixed neighborhood area where to grow renders it capable of grouping also spatially sparse and discontiguous subsets of correct matches.

In principle, the composition of a GAM might depend on the choice of its first region in step 1. However, the near-to-far growing order and the distance-based weighting make the algorithm highly order-independent. This is confirmed by experiments on several scenes, where the composition of the GAMs was stable (variations of about 1%) in spite of random permutations of the input regions.

10.3. Fundamental Properties

The GAM decomposition has two fundamental properties:

1. It is unlikely for mismatches to form large GAMs. Mismatches have *independent*, random affine transformations, uniformly spread in the large 6D affine transformation space. Thus, the more mismatches you consider, the less likely they will respect the constructor's criterion, that their affine transformations vary gradually from a region to the next. A set of mismatches has widely varying, inconsistent transformations. More precisely, the probability that N mismatches are grouped in the same GAM is expected to decrease roughly exponentially with N. On the other hand, several correct matches lying on the same surface will form a larger GAM, because of their coherent affine transformations. Therefore, the number of matches in a GAM relates to its probability of being correct.

2. A GAM is most often composed of either only correct matches or only mismatches. The reasons lie again in the randomness of mismatches' transformations. Suppose a correct GAM is being grown, and at some iteration the algorithm has to decide whether to add a nearby mismatch. This is unlikely to happen as the mismatch has little chances to offer a suitable affine transformation. Even in this case, the probability to add a second mismatch is again equally low. The total probability quickly drops with the number of added mismatches. As a result, correct GAMs are composed of correct matches only, or they contain only very few mismatches (typically 1 or 2).

As a combined effect of the two properties, mismatches are scattered over many small GAMs, while correct matches typically concentrate in a few larger GAMs. This brings the major advantage to organizing individual matches into GAMs: if a GAM contains many matches we know it is very probably correct. Small GAMs are most of the time mismatches, and sometimes they are minor groups of correct matches located on a small, or difficult to match, surface. Beside informing about correctness, the sizes of GAMs correlate with relevance: the larger a GAM is, the more important it is, because it covers a larger surface.

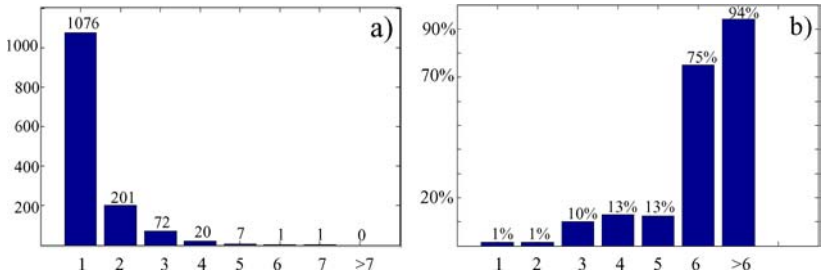


Figure 13. (a) Number of incorrect GAMs in function of their size (x-axis). (b) Percentage of correct GAMs.

The above properties are the reason of existence of GAMs and make them valuable as an intermediate grouping level on which to base powerful higher level algorithms. These need no longer consider each individual match, but can reason about complete GAMs instead, because matches and mismatches are separated into different GAMs. Hence, GAMs are seen as the new atomic units.

GAMs can be used beyond the object recognition context. In another work (Ferrari et al., 2004), we propose a GAM-based algorithm for simultaneously estimating the epipolar geometry between two images and filtering mismatches, which works in the presence of very high percentages of mismatches.

Experimental assessment. In order to assess the validity of the fundamental properties, we have matched 14 image pairs, run the GAM constructor, and measured size and composition of all resulting GAMs. The images come from diverse sources and contain planar, curved, as well as deformed surfaces. Seven pairs are wide-baseline stereo cases (WBS), while the others are object recognition cases, with the first image being a model view and the second a test image. The two kinds of data differ in several aspects. The recognition pairs present larger occlusion, scale change, and clutter. The WBS pairs feature a more complex geometry, with many fragmented surfaces, in contrast to the often compact objects in the recognition pairs. Six of the recognition cases come from our dataset (Section 13 results), while one is the teddybear used in the independent work of Rothganger et al. (2005). The WBS cases include three classic examples used in many papers: the Valbonne church (Schaffalitzky and Zisserman, 2002) the Graffiti wall (Mikolajczyk and Schmid, 2002) and the Dunster toy house (Pritchett and Zisserman, 1998). The region correspondences are produced by one-to-one matching for the WBS cases, and by soft-matching for the object recognition cases (Section 3).

In total there are 2253 matches, which have been partitioned into 1428 GAMs. 1378 of them are formed purely of mismatches, while there are 50 GAMs containing all 415 correct matches. We call the former *incorrect GAMs* and the latter *correct GAMs*. Since the overall ratio of correct matches is only 18.4%, the statistics are relevant and truly summarize the behavior of the GAM constructor.

Figure 13(a) plots the number of incorrect GAMs as a function of their size. The exponential decrease is clearly visible. There is only one incorrect GAM of size 6, and *none* larger than 7. This confirms the first fundamental property: it is unlikely for mismatches to form large GAMs. The second property is confirmed as well: 96.4% of all non-singleton GAMs are composed of either only correct matches or only mismatches (as the property trivially holds for singleton GAMs, they are not counted). The property is also almost fulfilled by the remaining GAMs, as they contain all correct matches, but one (2.4%) or two (1.2%). No GAM mixed more than two mismatches with a correct match, therefore meeting the expectations.

The relation between the size of a GAM and its probability of being correct is illustrated in Fig. 13(b), which plots the percentage of correct GAMs of size N, for various N. The chances that a GAM is correct quickly grow with its size, and is 94% for $N > 6$.

10.4. Example GAMs

Figure 14 shows some examples. The first is the well-known *Graffiti*, introduced in Mikolajczyk and Schmid (2002). The constructor algorithm grouped in a single GAM 71 matches spread over the whole wall, despite evident perspective effects. The matches are produced by the standard approach of Tuytelaars and Van-Gool (2000). The other example consists of two images of *Coleo*, a plush toy with a complex shape composed by

Figure 14. *Top*: Graffiti scene. A large GAM covers the whole wall, effectively bridging the perspective effect (only centers are shown). *Middle*: two GAMs on two very different views of Coleo. Bottom: close-up on some matches of the back-arm GAM. The geometric transformations vary over a wide range, but change gradually among spatially neighboring regions.

several curved surfaces. We matched the images with the image-exploration technique presented in part I, and fed the GAM constructor with the resulting region correspondences. There are many more correspondences than one would obtain by conventional matching, and they densely cover the parts of the object visible in both images. When applied to this input, the GAM decomposition is most interesting, because the constructor has enough prime matter to build GAMs covering larger areas, even if curved or deformed. Despite the very different viewpoints, the exploration algorithm produced about 120 correct matches, densely covering the parts visible in both views. The two largest GAMs correspond well to the principal contiguous surfaces, which are the head and the back-arm complex. Some of the matches among the latter GAM are shown in the close-ups. The regions are all circles of the same size in the left image, because they are part of one layer of the coverage generated in subsection 4.1. The contiguous variation of the regions' shapes in the right image mirrors the changes in affine transformation due to the varying surface orientation. Although the *range* of the transformations is very wide, the GAM grouper succeeded in grouping these matches in a large GAM, exploiting the *gradual* changing of the transformation from a region to the next.

11. Modeling from Multiple Views

Let's now turn to the central question of this part of the paper: how to exploit the relationships between multiple model views for recognition. In the modeling stage, the relationships are captured by a dense set of *region-tracks*. Each such track is composed by the image regions of a single physical surface patch along the model views in which it is visible. The tracks should densely connect the model views, because they will be used during recognition in order to establish connections among GAMs matched from different model views to the test image (Section 12).

This section explains how to build the model region-tracks, starting from the bare set of M unordered model images. First, dense two-view matches are produced between all pairs of model images. All pairwise sets of matches are then integrated into a single multi-view model. This process can be regarded as a specialized, dense counterpart of other sparse multi-view matching schemes, such as Schaffalitzky and Zisserman (2002; Ferrari et al. (2003).

In the following sections, we explain the method on 8 model views, taken at about 45 degrees during a complete tour of an example object (named *Coleo*, see next figures).

Dense two-view correspondences. A dense set of region correspondences between every two model views v_i, v_j is obtained using a simplified variant of the image-exploration technique (part I). More precisely, it uses a simple one-to-one nearest neighbor approach for the initial matching instead of the soft-matching phase, and there are no 'early' phases (Sections 4 and 5). The system directly goes to the 'main' phases after the initial matching (Sections 6 and 7). The use of this faster, less powerful version is justified because matching model views is easier than matching to a test image: there is no background clutter, and the object appears at approximately the same scale.

Let's recall that the image-exploration technique constructs correspondences for many overlapping circular regions, arranged on a grid completely covering the first model view v_i (*coverage regions*, see Section 4.1). The procedure yields a large set of reliable correspondences, densely covering the parts

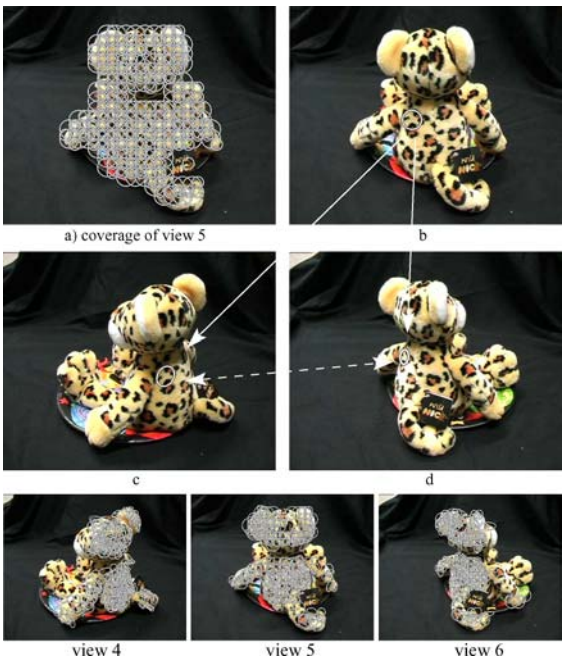


Figure 15 (a) Coverage regions for model view 5. (b) One of the coverage regions. (*c*+*d*) the corresponding regions constructed by the image-exploration algorithm in views 4 and 6. These direct matches 5 → 4 and 5 → 6 induce a three-view track across views 4, 5, 6. Hence, the transitive match 4 → 6 is implied. *Bottom*: 242 3-view tracks through views 4, 5, 6.

of the object visible in both views. Please note that the image-exploration matcher is not symmetric in the views, as it tries to construct correspondences in the second view, for the coverage regions of the first view (we say that it matches v_i *to* v_j, noted $v_i \rightarrow v_j$).

Dense multi-view correspondences. Once two-view region correspondences have been produced for all ordered pairs of model views (v_i, v_j), $i \neq j$, they can be organized into multi-view region tracks. When matching a view v_i to any other model view, we always use the same set of coverage regions. Therefore, each coverage region, together with the regions it matches in the other views, induces a region track (Fig. 15). Note that if a region is matched from view v_i to view v_j, and also from view v_i to view v_k, then it is implicitly matched between v_j and v_k as well, because it will be part of the same track. These *transitive matches* actively contribute to the inter-view connectedness, as they often link parts of the object that are harder to match directly. The final set of region tracks constitutes our object model. Figure 15 shows all 3-view tracks passing through views 4, 5, 6, after building the model from all 8 views.

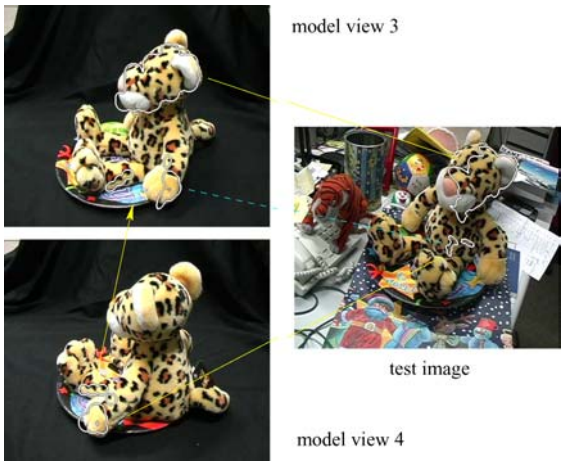


Figure 16. A correct GAM (head), matched from view 3, and an incorrect one (paw) from view 4. The paw GAM is transferred from model view 4 to model view 3 (arrow) via the model's connections.

12. Recognition from Multiple Views

Given a test image, the system should determine if it contains the modeled object. The first step is to match each model view of the object to the test image separately. For this purpose, the image-exploration technique is used again, this time in its full version. Each resulting set of region matches is then partitioned into GAMs. Each correct GAM usually corresponds to (part of) an object facet (Figs. 16, 17; only contours are shown).

However, at this stage, there is no guarantee that all GAMs are correct. As a result, there usually are some inconsistencies between GAMs. For instance, a GAM correctly matches the head of Coleo in Fig. 16 from model view 3 to the test image. Furthermore, there is another GAM erroneously matching the paw in model view 4 to the chest in the test image. Since the model views are interconnected by the model tracks, we know the correspondences of the regions on the paw between views 3 and 4. Therefore we consider the second GAM to match the chest in the test image to the paw in model view 3. Now both GAMs match model view 3 to the test image, and their (geometric) inconsistency can be measured and discovered.

Just as it finds conflicting GAMs, the system can notice compatible ones (Fig. 17). This is a good reason for considering them as more reliable and therefore

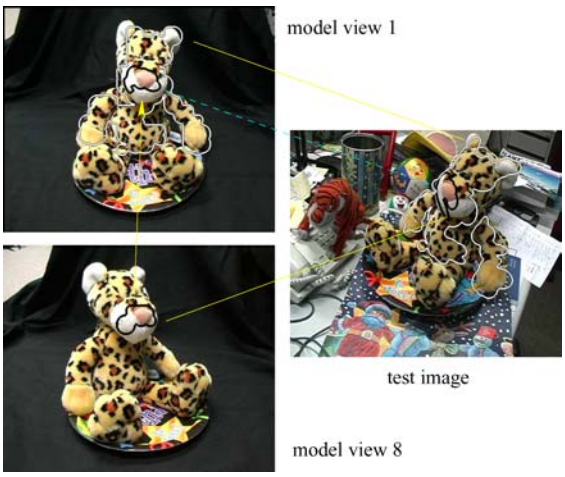


Figure 17. Two compatible (and correct) GAMs. The nose GAM (*black*) is initially matched from model view 8, and is transferred to model view 1. Note how the other GAM (*white*) is very large and covers the head, arms and chest. A GAM can extend over multiple facets when the combination of viewpoints and surface orientations make the affine transformations of the region matches vary smoothly even across facet edges. In these cases, the resulting GAMs are larger and therefore more reliable and relevant.

to reinforce the system's belief in the presence of the object. This leads to the main advantage in evaluating GAM compatibilities: the reliability of the recognition decision is enhanced, because higher scores can be assigned in positive cases (i.e. when the object is in the test image). As a secondary advantage, incorrect GAMs can be detected and removed, thus improving the segmentation.

In this section, we explain how to realize these ideas. For every pair of GAMs, we compute a compatibility score, quantifying the consistency of their spatial arrangement. In simple cases, the two GAMs are matched from the same model view and the score can be directly computed. In the more interesting cases where each GAM is from a different model view, we first *transfer* one of the GAMs to the model view of the other, by using the connections embedded in the model tracks. Next, the pairwise scores are integrated in a single *configuration score*. This varies as a function of the *configuration*, the subset of all GAMs which are considered correct. The score favors configurations containing large, compatible GAMs. This is justified because larger GAMs are more likely to be correct. A Genetic Algorithm is used to maximize the configuration score. The maximum yields the final recognition score and reveals which GAMs are deemed incorrect. The recognition score increases in the presence of compatible GAMs, thereby improving recognition performance.

The recognition score, and the decisions to remove GAMs, are based on a *global* analysis of the situation. This considers simultaneously relationships among all pairs of GAMS, coming from all model views. It is computationally feasible because there are much less GAMs (a few tens) than region matches (hundreds to thousands). This is an advantage of reasoning on the higher perceptual grouping level offered by GAMs. The system no longer needs to consider each single region individually, but it can rely on a meaningful organization instead. The following subsections describe the elements of the above scheme in more detail.

12.1. GAM Transfer

Consider a GAM matched from a model view v_i to the test image, and another GAM matched from a different model view v_j. Before computing the compatibility score for this GAM pair, they must be put in a common model view. Only then the geometrical coherence of their relative arrangement can be evaluated. A GAM is transferred from v_i to v_j as follows:

1. Determine the set of model regions Λ covering the same part of v_i as the GAM[5]. Remove from Λ all regions which are not part of a model track passing through v_j. The model can now predict the location and shape of the GAM in v_j.
2. Compute the affine transformations mapping each region of Λ from v_i to v_j (Fig. 18).
3. Project each GAM region to v_j via the affine transformation of the nearest region of Λ. Thereby, we have established a region-to-region correspondence for the GAM between the test image and model view v_j.

When transferring a GAM, it is like making a model-based prediction. The pairwise compatibility score (next subsection) evaluates to which degree the two GAMs are consistent with this prediction. This idea is essential: in this way the system exploits the relationships among the model views, in order to conclude more than what is possible from the mere collection of all GAMs. During modeling, the system learned the structure of the object in the form of region tracks, and it brings this insight to bear at recognition time by imposing order on the GAMs.

Note that a GAM cannot be transferred if the model regions it covers in view v_i are not visible in view v_j

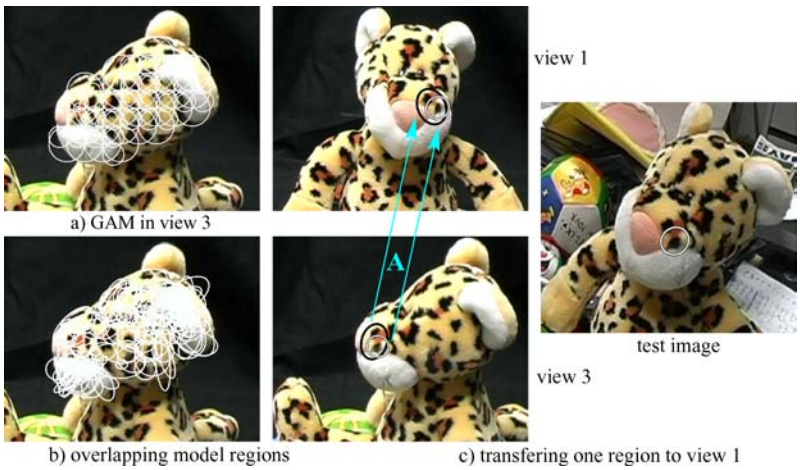


Figure 18. The GAM transfer mechanism. (a) The GAM to be transfered, which is originally matched from view 3 to the test image. (b) The set Λ of overlapping model regions. (c) One of the GAM regions (*white*) is transferred from view 3 to view 1, via the affine transformation of the nearest region of Λ (*black*). We now know the correspondence between view 1 and the test image.

(Λ is empty). In these cases, the compatibility score is not computed, and a neutral score is assigned instead.

12.2. Pairwise Compatibility Score

We evaluate here the geometric consistency of a pair of GAMs. Both GAMs are matched between the test image and a model view v_i. If at least one GAM is incorrect, we wish this measure to be low.

The compatibility score is based on the *sidedness constraint* for triples of region matches, introduced in Section 7. We check the constraint for all triples formed by a region from a GAM and two regions from the other GAM. The percentage of triples respecting the constraint is our choice for the compatibility score of the GAM pair.

The key idea is that if a region is picked from an incorrect GAM, we expect most triples in which it takes part to violate the constraint. Note that no triple is composed of regions from a single GAM. This is important when exactly one of the GAMs is correct. In these cases, most triples based only on the correct GAM will respect the constraint, and would therefore falsely raise the score.

The proposed score tolerates a substantial amount of non-rigid deformation. This preserves the system's capability of recognizing deformable objects. Moreover, it is insensitive to inaccurately localized region matches (Section 7.3). The score can penalize conflicting GAMs, but also highlight compatible pairs of GAMs. Although based on comparing region matches, it captures the compatibility of the GAMs as a whole.

12.3. Configuration Score

The compatibility scores are computed for all pairs of GAMs, and combined here in a single *configuration score*.

The compatibility scores range in [0, 1]. Based on a threshold t, we linearly transform the interval $[0, t]$ to $[-1, 0]$ and the interval $[t, 1]$ to $[0, 1]$. The values then range in $[-1, 1]$. In all experiments, the threshold $t = 0.2$ splits the original range into positive and negative parts. Positive scores now indicate that two GAMs are likely to belong together, while negative ones indicate incompatibility.

Let a *configuration* C be a subset of the available GAMs. What is the score of a configuration? It should be high when containing large, mutually compatible GAMs. It should be lower in the presence of incompatible ones. These two forces, pairwise corroboration and individual size, are combined into the following configuration score

$$S(C) = \sum_{P \in C} \left(\text{Size}(P) + \sum_{Q \in C \setminus P} (\text{Comp}(P, Q) \cdot \text{Size}(Q)) \right) \tag{8}$$

with $\text{Size}(P)$ the number of regions in GAM P, and $\text{Comp}(P, Q) \in [-1, 1]$ the pairwise compatibility

scores. We are interested in the maximum value of $S(C)$, and in the configuration for which it occurs. The maximum value is used as recognition criterion, to decide whether the object is in the test image. As argued before, larger GAMs are trusted more (first summation term). The second term makes the contribution of each GAM heavily dependent on its compatibility with the others, especially the larger ones. A GAM whose negative compatibilities lower S will be left out. Smaller GAMs can also be part of the maximum configuration, depending on how compatible they are with the others. An important effect of the second summation term is that the total score can be *much higher* than the mere sum of the sizes of all correct GAMs. This reflects the key idea that compatible configurations are worth more because they more reliably indicate the presence of the object. This increases the separation between scores in positive and negative cases, thus improving discriminative power.

The GAMs not selected by the best configuration are deemed incorrect and discarded. This decision is based on a global analysis. Typically, several incorrect GAMs are detected thanks to their incompatibility with GAMs matched to other model views. Such a case couldn't have been discovered by looking at the GAM's model view in isolation. This is another benefit of our proposal for integrating multiple model views. Finally, note how we treat a GAM as a unit: either we keeps all its matches, or none.

12.4. Maximization by Genetic Algorithm

We now need to find the configuration which maximizes function (8). Unfortunately, we can't try them all out, as there are 2^n possible configurations of n GAMs. Moreover, a function in the form of (8) cannot be maximized by graph-cuts methods, as shown by Kolmogorov and Zabih (2002).

We designed a Genetic Algorithm (GA) to find an approximation of the solution. GAs offer an elegant and flexible framework for optimizing functions of any form. We represent a configuration by a binary indicator vector I of length n. If $I(p) = 1$, the pth GAM is in the configuration. The *fitness function* $F(I)$ is defined equivalent to $S(C)$. The GA follows several steps:

1. *Initialize*. Create a random, uniformly distributed population of binary n-vectors. The size of this population is $l = \text{ceil}(\sqrt{2n})^2$. Since this enforces $\sqrt{l}$ to be an integer, it simplifies the later crossover.
2. *Fitness*. Evaluate the fitness function $F(I)$ for each individual. Stop if the best individual is identical as in the previous generation.
3. *Crossover.* Consider the best $\sqrt{l}$ individuals. Derive the next generation by crossing over all pairs of them. Crossing over two individuals means keeping the identical bits and randomly choosing the different bits. This amounts to producing $l - \sqrt{l}$ new individuals, and copying the current best $\sqrt{l}$.
4. *Mutation*. Each bit of each individual in the new population is switched with probability 0.1. This avoids that the algorithm explores only the part of the search space spanned by the best individuals.
5. *Iterate*. Iterate to point 2.

In various experiments[6] this GA proved effective by approximating the true exhaustive search solution to less than 1 small GAM difference on average, in comparisons with up to $n = 20$ GAMs. It is also very time efficient, as it solves cases with $n = 20$ within some seconds (exhaustive search needs more than 1 hour), and scales well, taking less than one minute for $n = 60$, a problem size for which the real optimum cannot be computed. One of the reasons for this performance is the nature of the optimization problem itself. In the vast majority of cases where the object is in the test image, the GAMs sizes are very non-uniformly distributed, with some large GAMs, and a greater number of smaller ones. Moreover, the value of function (8) raises more when large GAMs are in C, and even much more with compatible large GAMs. As a result, the search space has a strong non-flat shape, and usually features high peaks for C containing at least some of the largest GAMs. These characteristics significantly ease the task of the GA.

13. Results

The next two Sections present results for the image-exploration technique (part I) applied to an object recognition dataset taken by the authors, and within a video retrieval application. Subsection demonstrates the improvements brought by integrating the contributions of multiple model views (part II).

13.1. Recognition on Our Dataset

The dataset in this Section[7] consists of 9 model objects and 23 test images. In total, the objects appear 43

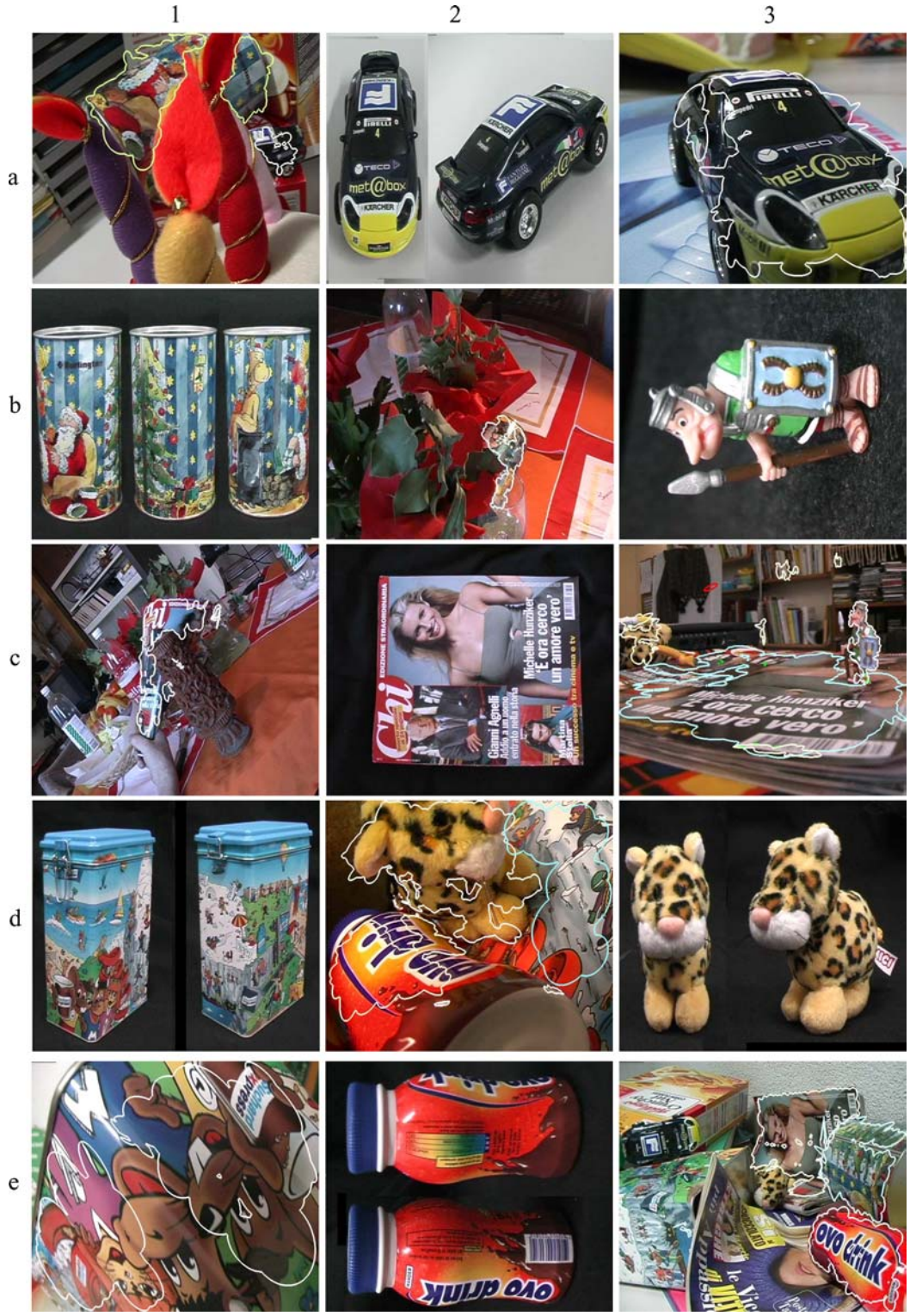


Figure 19. Recognition results (see text).

times, as some test images contain several objects. To facilitate the discussion, the images are referred to by their coordinates as in Fig. 19, where the arrangement is chosen so that a test image is adjacent to the model object(s) it contains. There are 3 planar objects, each modeled by a single view, including a *Kellogs* box[8] and two magazines, *Michelle* (Figure 2c) and *Blonde* (analog model view). Two objects with curved shapes, *Xmas* (b1) and *Ovo* (e2), have 6 model views. *Leo* (d3), *Car* (a2), *Suchard* (d1) feature more complex 3D shapes and have 8 model views. Finally, one frontal view models the last 3D object, *Guard* (b3). Multiple model views are taken equally spaced around the object. The contributions from all model views of a single object are combined by superimposing the area covered by the final set of matched regions (to find the contour), and by summing their number (detection criterion). All images are shot at a modest resolution (720×576) and all experiments are conducted with the same set of parameters. In general, in the test cases there is considerable clutter and the objects appear smaller than in the models (all model images have the same resolution as the test images and they are shown at the same size).

Tolerance to non-rigid deformations is shown in c1, where *Michelle* is simultaneously strongly folded and occluded. The contours are found with a good accuracy, extending to the left until the edge of the object. Note the extensive clutter. High robustness to viewpoint changes is demonstrated in c3, where *Leo* is only half visible and captured in a considerably different pose than any of the model views, while *Michelle* undergoes a very large out-of-plane rotation of about 80 degrees. *Guard*, occluding *Michelle*, is also detected in the image, despite a scale change of factor 3. In d2, *Leo* and *Ovo* exhibit significant viewpoint changes, while *Suchard* is simultaneously scaled by factor 2.2 and 89% occluded. This very high occlusion level makes this case challenging even for a human observer. A scale change of factor 4 affecting *Suchard* is illustrated in e1. In figure la, *Xmas* is divided in two by a large occluder. Both visible parts are correctly detected by the presented method. On the right side of the image, *Car* is found even if half occluded and very small. *Car* is also detected in spite of a considerable viewpoint change in a3. The combined effects of strong occlusion, scale change and clutter make b2 an interesting case. Note how the boundaries of *Xmas* are accurately found, and in particular the detection of the part behind the glass. As a final example, 8 objects are detected at the same time in e3 (for clarity, only 3 contours are shown). Note the correct segmentation of the two deformed magazines and the simultaneous presence of all the aforementioned difficulties.

Figure 20(b) presents a close-up on one of 93 matches produced between a model view of *Xmas* (left) and test case b2 (right). This exemplifies the great appearance variation resulting from combined viewpoint, scale and illumination changes, and other sources of image degradation (here a glass). In these cases, it is very unlikely for the region to be detected by the initial region extractor, and hence traditional methods fail. This figure also illustrates the accuracy of the correspondences generated by the expansion phases.

As a proof of the method's capability to follow deformations, we processed the case in Fig. 20(c) starting with only one match (dark). 356 regions, covering the whole object, were produced. Each region's shape fits the local surface orientation (for clarity, only 3 regions are shown).

The performance of the system was quantified by processing all pairs of model-object and test images, and counting the resulting number of region matches. The highest ROC curve in Fig. 20(a) depicts the detection rate versus false-positive rate, while varying the detection threshold from 0 to 200 matches. An object is detected if the number of produced matches, summed over all its model views, exceeds this threshold. The method performs very well, and can achieve 98% detection with 6% false-positives. For comparison, we processed the dataset also with 4 state-of-the-art affine region extractors (Baumberg, 2000; Mikolajczyk and Schmid, 2002; Obrdzalek and Matas, 2002; Tuytelaars and Van-Gool, 2000), and described the regions with the SIFT (Lowe, 2004) descriptor,[9] which has recently been demonstrated to perform best (Mikolajczyk and Schmid, 2003). The matching is carried out by the 'unambiguous nearest-neighbor' approach[10] advocated in Baumberg (2000) and Lowe (2004): a model region is matched to the region of the test image with the closest descriptor if it is closer than 0.7 times the distance to the second-closest descriptor (the threshold 0.7 has been empirically determined to optimize results). Each of the central curves illustrates the behavior of a different extractor. As can be seen, none is satisfactory, which demonstrates the higher level of challenge posed by the dataset and therefore suggests that our approach can broaden the range of solvable Object Recognition cases. Closer inspection reveals the source of failure: typically only very few, if any, correct matches are produced when the object is present, which in turn is due to

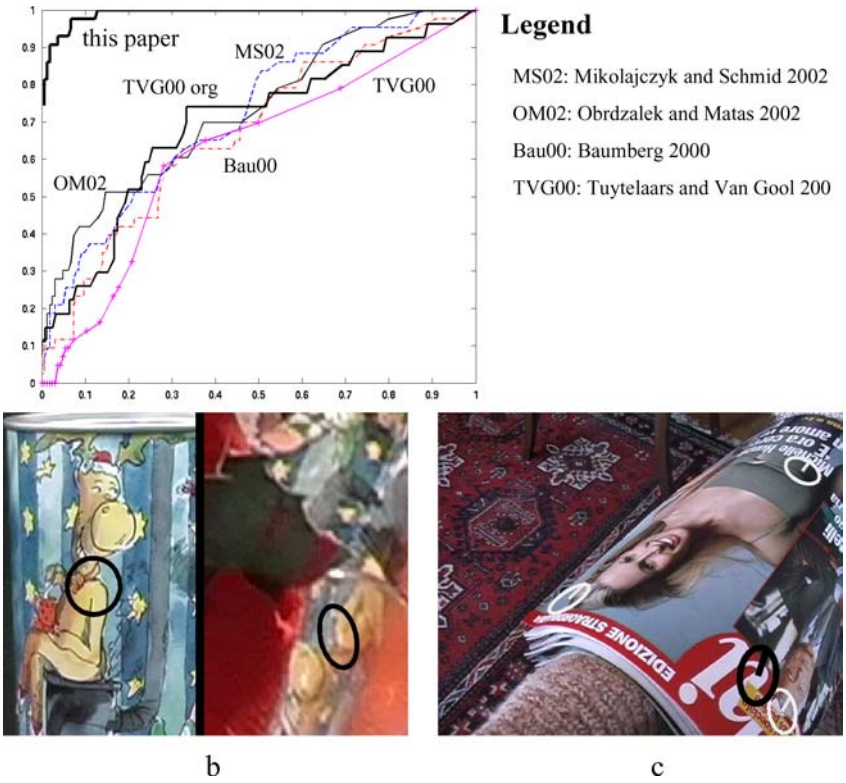


Figure 20. (a) ROC plot. False-positives on the *X*-axis, detection rate on the *Y*-axis. (b) Close-up on one match of case b2. (c) Starting from the black region only, the method covers the magazine with 365 regions (3 shown).

the lack of repeatability and the inadequacy of a simple matcher under such difficult conditions. The important improvement brought by the proposed method is best quantified by the difference between the highest curve and the central thick curve, representing the system we started from Tuytelaars and Van-Gool (2000) ('TVG00 org' in the plot).

Figure 21(a) shows a histogram of the number of final matches (recognition score) output by our system. The scores assigned when the object is in the test image

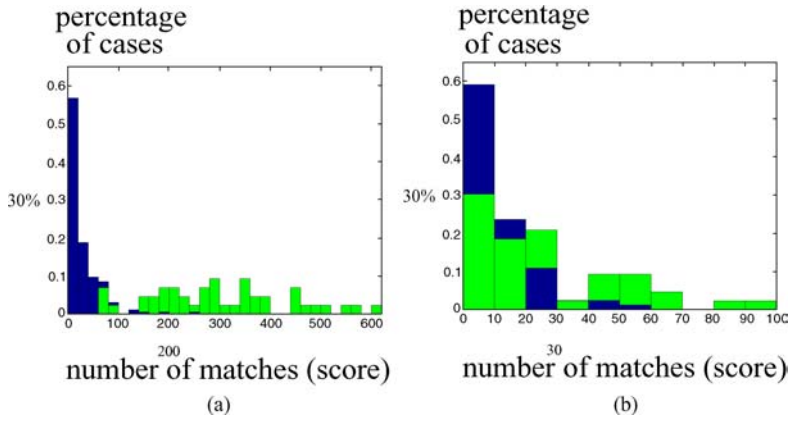


Figure 21. Distribution of scores (percentage; bright = positive cases; dark = negative cases). (a) For our method. (b) For the traditional matching of the regions of Obrdzalek and Matas (2002).

(positive cases) are much higher than when the object is absent (negative cases). This very good separation brings discriminative power and is due to the combination of two effects. First, the exploration process tends to implode in negative cases, because the expansion phases can do little and the contraction phases eat up most of the matches. Conversely, the method fills the object with matches when it is present, as expansions can prosper on much fertile surface. As a comparison with the traditional methods, the standard matching of regions of Obrdzalek and Matas (2002) based on the SIFT descriptor, yields two hardly separable distributions (Fig. 21 b), and hence the unsatisfactory performance in the ROC plot. Similar histograms are produced based on the other feature extractors (Baumberg, 2000; Mikolajczyk and Schmid, 2002; Tuytelaars and Van-Gool, 2000).

As last comparison, we consider the recent system (Rothganger et al., 2005) which constructs a 3D model of each object prior to recognition. We asked the authors to process our dataset. As they reported, because of the low number of model views, their system couldn't produce meaningful models, and therefore couldn't perform recognition. Conversely, we have processed the dataset of Rothganger et al. (2005) with our complete system (including GAMs and multi-view integration). It performed well, and achieved 95% detection rate for 6% false-positives (see Rothganger (2005) for more details).

13.2. Video Retrieval

In this experiment, the goal is to find a specific object or scene in a test video. The object is only given as delineated by the user in one model image. In Sivic and Zisserman (2003) another region-based system for video object retrieval is presented. However, it focuses on different aspects of the problem, namely the organization of regions coming from several shots, and weighting their individual relevance in the wider context of the video. At the feature level, their work still relies solely on regions from standard extractors.

Because of the different nature of the data, the system differs in a few points from the object recognition one. At recognition time the test video is segmented into shots, and a few representative keyframes are selected in each shot by the algorithm of Osian and Van-Gool (2004). The object is then searched in each keyframe separately, by a simplified version of the image-exploration technique. Specifically, it has a simple one-to-one nearest neighbor approach for the initial matching instead of the soft-matching phase, there are no 'early' phases, and there is only one layer of coverage regions. This simpler version runs faster (about twice as fast), though it is not as powerful. It takes about 2 minutes to process a (object, keyframe) pair on a common workstation (2.4 Ghz PC).

We present results on challenging, real-world video material, namely television news broadcast provided by the RTBF Belgian television. The data comes from 4 videos, captured on different days, each of about 20 minutes. The keyframes have low resolution (672×528) and many of them are visibly affected by compression artifacts, motion blur and interlacing effects. We selected 13 diverse objects, including locations, advertising products, logos and football shirts, and delineated each in one keyframe. Each object is searched in the keyframes of the video containing its model-image. On average, a video has 325 keyframes, and an object occurs 7.4 times. The number of keyframes not containing an object (negatives), is therefore much greater than the number of positives, allowing to collect relevant statistics. A total of 4236 (object, keyframe) image pairs have been processed.

Figure 22 show some example detections. A large piece of quilt decorated with various flags (a2) is found in a3 in spite of non-rigid deformation, occlusion and extensive clutter. An interesting application is depicted in b1-b2-b3. The shirts of two football teams are picked out as query objects (b2), and the system is asked to find the keyframes where each team is playing. In b1 the Fortis shirt is successfully found in spite of important motion blur (close-up in a1). Both teams are identified in b3, where the shirts appear much smaller and the Dexia player is turned 45 degrees (viewpoint change on the shirt). The keyframe in c1 instead, has not been detected. Due to the intense blur, the initial matcher does not return any correct correspondence. Robustness to large scale changes and occlusion is demonstrated in a4, where the UN council, modeled in b4, is recognized while enlarged by a scale factor 2.7, and heavily occluded (only 10% visible). Equally intriguing is the image of Figure 4c, where the UN council is seen from an opposite viewpoint. The large painting on the left of b4 is about the only thing still visible in the test keyframe, where it appears on the right side. The system matched the whole area of the painting, which suffers from out-of-plane rotation. As a last example, a room with Saddam Hussein is found

Figure 22. Video retrieval results. The parts of the model-images not delineated by the user are blanked out.

in Figure 3c (model in c2). The keyframe is taken under a different viewpoint and substantially corrupted by motion blur.

The retrieval performance is quantified by the *detection rate* and *false-positive rate*, averaged over all objects. An object is detected if the number of final matches, divided by the number of model coverage regions, exceeds 10% (detections of model-keyframes are not counted). The system performs well, by achieving an average detection rate of 82.4%, for a false-positive rate of 3.6%. As a comparison, we repeated the experiment with (Tuytelaars and Van-Gool, 2000), the method we started from. It only managed a 33.3% detection rate, for a false-positive rate of 4.6%, showing that our approach can substantially boost the performance of standard affine invariant matching procedures.

13.3. Multiple-View Integration

Example cases. We present a few examples on Coleo, to illustrate the behavior of the multiple-view integration scheme. Coleo features a complex geometry composed by several curved surfaces. Moreover, it is covered by ambiguous texture, formed by many small variations on the same basic pattern, which challenge the matching process. The model is built from only 8 views.

On the example of Fig. 16 and 17, the system initially produces 33 GAMs. Only 9 of the GAMs are correct, but 4 of them are very large (more than 60 matches) and contain the majority of the correctly matched regions. The multi-view integration scheme selects 10 GAMs in the configuration with the maximal score. All 9 correct GAMs are included, while all but one of the 24 erroneous GAMs are successfully detected and discarded. The final recognition score is 1770, which is three times as much as the total number of matches within the correct GAMs (596). Hence the confidence about the presence of the object is significantly boosted, compared to the simpler approach taken in Section 13 which just accumulates the number of matches from all model views as score. Moreover, when the object is not in the test image, the confidence score is decreased. As combined effect, the scores assigned in the two cases are more separated, which leads to enhanced discriminative power. Figure 23(a) shows the final segmentation, as the total area covered by the 10 selected GAMs.

Figure 23. Coleo cases. (a) The example used in part II. (b) Deformed case. The raised arm and the deformed chest are successfully detected. The minor background blobs are due to a few incorrect GAMs. (c) A challenging case with viewpoint remarkably different from any model view. (d) Some of the removed GAMs. (e) Close-up on some of the matches of case b. The regions are all circles in the left image because they are part of the homogeneous coverage. The shapes of the constructed correspondences (*right*) automatically adapt to the changing surface orientation.

A challenging case is shown in Fig. 23(c). The viewpoint is from above, and remarkably different from any model view. The object appears twice smaller than in the model views, and is partially occluded by a ball (head) and a plush wildcat (front). 37 GAMs are initially produced, out of which 5 are correct and quite large (43 matches on average). Most of the 32 wrong ones are composed by few matches. Our method selects all 5 correct GAMs, and 3 small incorrect ones, thereby effectively removing the large majority of mismatches (93%). The recognition score is 581, which is 2.6 times the number of matches in all correct GAMs (216). Note the quality of the segmentation, which includes even parts of the tail and the left paw. Figure 23 d shows some of the removed GAMs.

In the case of Fig. 23(b) Coleo is non-rigidly deformed. One arm is raised (left of the image), the paws face each other and the chest is being compressed. Nevertheless, the system could identify the object (configuration score 1270), and included in the segmentation also the arm and the chest. The paws were missed, because too occluded (right paw) and turned so as to hide the bottom part, mostly visible in the model views (left paw). A closer look at the chest allows to fully appreciate the behavior of the image-exploration technique (Fig. 23(e)). The pressure applied by the finger causes considerable distortions of the texture pattern. The system responds by altering the shape of each region in the test image, so as to mirror the wide variation of the local surface orientation.

Figure 24. Effects of additional model views. One of the 4 additional views (*left*), and segmentation for the case of Figure 23c, when using 12 model views (*right*). Notice the improvement, e.g. the head is more complete, and the left paw is included.

Effect of additional model views. Although the above reported cases are solved satisfactorily based on 8 model views, it is interesting to inspect the effects of including more model views. Figure 24 shows one of the 4 additional model views, which are taken from above at 90 degrees intervals. Matching also these new model views to the test image of Fig. 23(c) results in a total of 60 GAMs, including 9 correct. 8 correct GAMs, and 10 incorrect ones, are selected by the best configuration, giving a score of 2498, almost 5 times the total size of correct GAMs (511). Not only the score is much higher than when using 8 model views (581), but especially the ratio to the number of correct matches is larger (it was 2.6 before). The score grows faster than linearly with the number of compatible GAMs, realizing the idea that since compatible GAMs reveal consistent hypotheses, the system's confidence should quickly grow with them. When more model views are available, their larger overlap leads to a greater number of GAMs and a higher degree of their mutual corroboration. More model views means more cooperation and the proposed approach can effectively measure it. Besides, the segmentation also marginally improves, and now covers the left paw and more of the head.

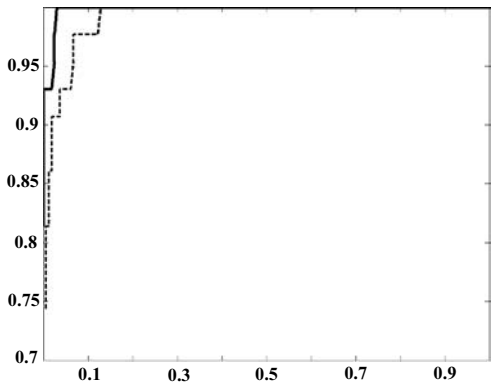


Figure 25. ROC plot. Adding the multiple-view integration layer brings significant improvement (*thick line*) on our dataset.

While including 10 incorrect GAMs might seem a lot, it must be noted that the other 41 incorrect ones are filtered out. Moreover, the 10 retained GAMs contain only a few matches each (3.8 on average) and their total size makes up only 11% of the mismatches within all 51 incorrect GAMs.

13.3.1. Impact on our dataset. In order to test the effects of the multiple-view integration scheme on a larger scale, we have applied it to the whole dataset of Section 13.1. We have first built models for all 9 objects, via the procedure of Section 11. Then, the outputs of all image-exploration matching processes for every pair of object and test images have been integrated as explained in Section 12. Notice how the scheme seamlessly accommodates for objects having only one model image. In these cases, it naturally reduces to an advanced two-view filter, which verifies the mutual compatibilities of GAMs matched between the model view and the test image. The parameters are kept the same throughout the whole experiment. The ROC plot in Fig. 25 shows important improvement over the one obtained without multiple-view integration. The system now attains the excellent performance of 100% detection, for 3% false-positives.

14. Related Work

Part I: simultaneous object recognition and segmentation. The presented technique belongs to the category of appearance-based object recognition. Since it can extend any approach which matches affine invariant regions between images, it is tightly related to this class of methods. The novelties and improvements brought by our approach are enumerated in the introduction section and demonstrated in the result Section 13.

Beyond the realm of local invariant features, there are a few works which are related to ours, in that they also combine recognition with segmentation. Leibe and Schiele (2004) present a method to detect an unknown object instance of a given category and segment it from a test image. The category (e.g. 'cows') is learnt from example instances (images of particular cows). However, the method does not support changes in camera viewpoint or orientation. In Yu et al. (2002), low-level grouping cues based on edge responses, high-level cues from a part detector and spatial consistency of detected parts, are combined in a graph partitioning framework. The scheme is shown to recognize and segment a human body in a cluttered image. However, the part detectors need a considerable number of training examples, and the very 'parts' to be learned are manually indicated ('head', 'left arm', etc.). Moreover, there is no viewpoint, orientation or scale invariance. Both methods are suited for categorization, and not specialized in the recognition of a particular object instance.

While we believe our approach to be essentially new, some components are clearly related to earlier research. The filter in Section 7 is constructed around the sidedness constraint. A similar constraint, testing the cyclic ordering of points, was used for wide-baseline matching in Tell and Carlsson (2002). Moreover, the 'propagation attempt' at the heart of the expansion phases is an evolution of the idea of 'growing matches' proposed by Pritchett and Zisserman (1998), Schaffalitzky and Zisserman (2002a, b). While they use existing affine transformations only to *guide the search* for further matches, our approach actively *generates* new regions, which have not been originally extracted. This is crucial to counter the repeatability problems stated in the introduction. Previously, a different, pixel-by-pixel propagation strategy was proposed in Lhuillier and Quao (2002), but it is applicable only in case of small differences between the images.

Part II: integrating multiple model views The GAM idea is similar in spirit to the work of Selinger and Nelson (1999), who advocate the benefits of an intermediate perceptual grouping level between primitives and views. Unlike in their work, here the primitives being grouped are region matches, rather than contour fragments. Moreover, GAMs are inherently a two-view concept, whereas contour fragments are defined in individual views. Very recently, Lazebnik et al. (2004)

have proposed to cluster nearby matches into semi-local groups, coined 'affine parts'. Since all matches in one such part are rigidly mapped by a *single* affine transformation, they are limited to cover semi-local planar areas. In contrast, GAMs are more general as they can cover any smooth surface, be it large, curved or deformed.

Since finding GAMs is not a goal *per se*, but rather an intermediate representation to enable higher level algorithms, their relation to the research world is better understood when considering our approach to integrating the contributions of multiple model views for recognition. If we take a step back from local invariant regions, and look at the wider world of appearance-based Object Recognition, we find much research on modeling 3D objects using multiple training viewpoints. For example in the works on aspect graphs (Cyr and Kimia, 2001), or on appearance eigenspaces (Murase and Nayar, 1995). However, when turning our attention to local invariant regions, we notice that nearly all works focus on one model image, or use multiple model images just independently, without trying to relate them or exploit their interplay (e.g. Ferrari et al., 2004; Lazebnik et al., 2004; Obrdzalek and Matas, 2002; Schmid, 1996, 1999). Only very few such earlier works try to capture and exploit the relationships among the model views. In Lowe (2001), similar model views are clustered, and links are made between corresponding features in adjacent clusters. By following the links, a feature from the test image votes for the view to which it is matched, and for the adjacent ones. The system gains robustness, because the votes are not dispersed among neighboring model views. In comparison to that work, we believe that our approach offers deeper integration among the model views. Multiple views actively *cooperate*: by reciprocally (in)validating GAMs arising from different views, they corroborate, or inhibit, the hypotheses of correspondence among parts of the object surface they represent. Moreover, the system arrives at a global recognition score, based on all GAMs and their mutual compatibility as expressed by the model views. This score grows in presence of compatible GAMs, thereby explicitly taking into account that hypotheses shared by multiple model views more reliably indicate the presence of the object. The very organization of region matches into GAMs, which become the new unit of reasoning, is a difference and novelty of our approach.

In Rothganger et al. (2005), a high degree of multiple-view integration is reached by building a 3D model of the object, prior to recognition. The method imposes two-view and multiview geometric constraints on subsets of matches, and obtains partial reconstructions by factorization. These partial reconstructions are then registered in a global frame by aligning points common to overlapping subsets. In contrast, our method does not build a 3D model. This has the advantage that the selection of model views is less constrained. Indeed, not all features need to be visible in at least two or three views, and the method can work also with a single view, or with disjoint views. Moreover, there is no danger of degenerate cases such as views showing only a single planar part. As an additional advantage, our method does not make rigidity assumptions and is capable of recognizing objects undergoing non-rigid deformations.

15. Conclusion and Outlook

In the first part of the paper we have presented an approach to object recognition capable of solving particularly challenging cases. Its power roots in the 'image exploration' technique. Every single correct match can lead to the generation of many correct matches covering the smooth surface on which it lies, even when starting from an overwhelming majority of mismatches. Hence, the method can boost the performance of any algorithm which provides affine regions correspondences, because very few correct initial matches suffice for reliable recognition. Moreover, the approximate boundaries of the object are found during the recognition process, and non-rigid deformations are explicitly taken into account, two features lacking in competing approaches (e.g. Baumberg, 2000; Lowe, 2004; Mikolajczyk and Schmid, 2002; Obrdzalek and Matas, 2002; Rothganger et al., 2005; Schaffalitzky and Zisserman, 2002; Tuytelaars and Van-Gool, 2000).

The second part of the paper introduced the GAM concept, and extended the recognition scheme to exploit the relationships among multiple model views to integrate their contributions during recognition. This increases the discriminative power due to the higher scores in positive cases. Moreover, the segmentation quality improves due to the removal of spurious region matches. Multi-view integration is achieved without rigidity assumptions, and without constructing a 3D model. The heart of the approach, GAMs, are capable of covering planar, curved or smoothly deformed surfaces, and posses two fundamental properties

which reveal valuable for the design of higher-level algorithms. GAMs are useful in several contexts of computer vision. In Ferrari (2004) and Ferrari et al. (2004) they are used in a powerful two-view filter, robust to very high amounts of mismatches. In a sense, GAMs also form an alternative to the elusive concept of 'object parts', in that they offer a perceptual unit between the local features and the global object.

Some individual components of the scheme, like the topological filter and GAMs, are useful in their own right, and can be used profitably beyond the scope of this paper.

In spite of the positive points expressed above, our approach is not without limitations. One of them is the computational expense: in the current implementation, a 2.4 Ghz computer takes about 4–5 minutes, on average, to process a pair of model and test images. Although we plan a number of speedups, the method is unlikely to reach the speed of the fastest other systems (the system of Lowe (2001, 2004) is reported to perform recognition within seconds). As another limitation, our method is best suited for objects which have some texture, much like the other recognition schemes based on invariant regions. Uniform objects (e.g. a balloon) cannot be dealt with and seem out of the reach of this kind of approaches. They should be addressed by techniques based on contours (Cyr and Kimia, 2001; Selinger and Nelson, 1999). Hence, a useful extension would be to combine some sort of 'local edge regions' with the current textured regions. Another interesting evolution would be to make the multiple-view integration scheme more active. Currently all model views are first matched to the test image, with the integration happening only afterwards. However, we could start by matching to a single view only and then employ the model connections to decide if and which other model view to try out. Finally, using several types of affine invariant regions simultaneously, rather than only those of Tuytelaars and Van-Gool (2000), would push the performance further upwards.

Notes

1. The R, G, B colorbands range in $[0, 255]$, so $\overline{\text{sim}}$ is within $[-4.41, 2]$. A value of 1.0 indicates good similarity. In all experiments the matching thresholds are $t_1 = 0.6$, $t_2 = 1.0$.
2. These values are for an image of 720×576 pixels, and are proportionally adapted for images of other sizes.
3. This is set to 1.3 in all our experiments.
4. In all experiments the radius is set to 1/6 of the image size.
5. This is implemented by selecting the model regions which strongly overlap (more than 70%) with the image area covered by the union of the GAM's regions.
6. These experiments are reported in full detail in Ferrari (2004, pp. 193–195).
7. The dataset is available at www.vision.ee.ethz.ch/∼ferrari.
8. The kellogs box is used throughout the paper as a case-study.
9. All region extractors and the SIFT descriptor are implementations of the respective authors. We are grateful to Jiri Matas, Krystian Mikolajczyk, Andrew Zisserman, Cordelia Schmid and David Lowe.
10. We have also tried the standard approach, used in Mikolajczyk and Schmid (2001, 2003), Obrdzalek and Matas (2002), Tuytelaars and Van-Gool (2000), which simply matches two nearest-neighbors if their distance is below a threshold, but it produced slightly worse results.

References

Baumberg, A. 2000. Reliable feature matching across widely separated views. In *ICCV*, pp. 774–781.

Bebis, G., Georgiopoulos, M., and Lobo, N.V. 1995. Learning geometric hashing functions for model-based object recognition. In *ICCV*, pp. 543–548.

Chum, O., Matas, J., and Obdrzalek, S. 2003. Epipolar geometry from three correspondences. In *Computer Vision Winter Workshop*.

Cyr, C. and Kimia, B. 2001. 3D object recognition using similarity-based aspect graph. *ICCV*, 254–261.

Ferrari, V. 2004. Affine Invariant Regions ++. PhD Thesis, Selected Readings in Vision and Graphics, Springer Verlag, Zuerich, CH. www.vision.ee.ethz.ch/∼ferrari

Ferrari, V., Tuytelaars, T., and Van-Gool, L. 2003. Wide-baseline multiple-view correspondences. *CVPR*, I:718–728.

Ferrari, V., Tuytelaars, T., and Van-Gool, L. 2004. Integrating multiple model views for object recognition. *CVPR*.

Ferrari, V., Tuytelaars, T., and Van-Gool, L. 2004. Simultaneous object recognition and segmentation by image exploration. *ECCV*, I:40–54.

Kolmogorov, V. and Zabih, R, 2002. What energy functions can be minimized via graph cuts ? *ECCV*, III:65–78.

Lazebnik, S., Schmid, C., and Ponce, J. 2004. Semi-local affine parts for object recognition. *BMVC*, II:779–788.

Leibe, B. and Schiele, B. 2004. Scale-invariant object categorization using a scale-adaptive mean-shift search, *DAGM*, 145–153.

Lhuillier , M. and Quan, L. 2002. Match propagation for image-based modeling and rendering, *PAMI*, 24(8).

Lowe, D. 2001. Local feature view clustering for 3D object recognition. *CVPR*, 682–688.

Lowe, D. 2004. Distinctive image features from scale-invariant keypoints. *IJCV*, 60(2): 91–110.

Matas, J., Chum, O., Urban, M., and Pajdla, T. 2002. Robust wide baseline stereo from maximally stable extremal regions. *BMVC*, 384–393.

Mikolajczyk, K. and Schmid, C. 2001. Indexing based on scale-invariant interest points. *ICCV*, I:525–531.

Mikolajczyk, K. and Schmid, C. 2002. An affine invariant interest point detector. *ECCV*, 128–142.

Mikolajczyk, K. and Schmid, C. 2003. A performance evaluation of local descriptors. *CVPR*, II:257–263.

Murase, H. and Nayar, S. 1995. Visual learning and recognition of 3d objects from appearance. *IJCV*, 14(1).

Obrdzalek, S. and Matas, J. 2002. Object recognition using local affine frames on distinguished regions. *BMVC*, 414–431.

Osian, M. and Van-Gool, L. 2004. Video shot characterization. *Machine Vision and Applications*, 15(3): 172–177.

Pritchett, P. and Zisserman, A.1998. Wide baseline stereo matching. *ICCV*, 754–760.

Rothganger, F., Lazebnik, S., Schmid, C., and Ponce, J. 2005. 3D object modeling and recognition using local affine-invariant image descriptors and multi-view spatial constraints. *IJCV*, in press.

Schaffalitzky, F. and Zisserman, A. 2002. Automated scene matching in movies. In *Workshop on Content-Based Image and Video Retrieval*, pp. 186–197.

Schaffalitzky, F. and Zisserman, A. 2002. Multi-view matching for unordered image sets. *ECCV*, I:414–427.

Schmid, C. 1996. Combining greyvalue invariants with local constraints for object recognition. *CVPR*, 872–877.

Schmid, C. 1999. A structured probabilistic model for recognition. *CVPR*, II:485–490.

Selinger, A. and Nelson, R.C. 1999. A perceptual grouping hierarchy for appearance-based 3d object recognition. *Computer Vision and Image Understanding*, 76(1): 83–92.

Sivic, J. and Zisserman, A. 2003. Video google: A text retrieval approach to object matching in videos. *ICCV*.

Swain, M.J., and Ballard, B.H. 1991. Color indexing. *IJCV*, 7(1): 11–32.

Tell, D. and Carlsson, S. 2002. Combining appearance and topology for wide baseline matching. *ECCV*, I:68–81.

Torr, P.H.S. and Murray, D.W. 1997. The development and comparison of robust methods for estimating the fundamental matrix. *IJCV*, 24(3): 271–300.

Tuytelaars, T. and Van-Gool, L. 2000. Wide baseline stereo based on local, affinely invariant regions. *BMVC*.

Tuytelaars, T., Van-Gool, L., Dhaene, L., and Koch, R. 1999. Matching affinely invariant regions for visual servoing. In *Intl. Conference on Robotics and Automation*, 1601–1606.

Yu, S.X., Gross, R., and Shi, J. 2002. Concurrent object recognition and segmentation by graph partitioning. *NIPS*.

Zhang, Z., Deriche, R., Faugeras, O., and Luong, Q. 1995. A robust technique for matching two uncalibrated images through the recovery of the unknown epipolar geometry. *Artificial Intelligence*, 78: 87–119.