Sub-linear Indexing for Large Scale Object Recognition

Štěpán Obdržálek and Jiří Matas
Center for Machine Perception, Czech Technical University Prague

Abstract

Realistic approaches to large scale object recognition, i.e. for detection and localisation of hundreds or more objects, must support sub-linear time indexing. In the paper, we propose a method capable of recognising one of N objects in $log(N)$ time.

The ”visual memory” is organised as a binary decision tree that is built to minimise average time to decision. Leaves of the tree represent a few local image areas, and each non-terminal node is associated with a ’weak classifier’. In the recognition phase, a single invariant measurement decides in which subtree a corresponding image area is sought.

The method preserves all the strengths of local affine region methods – robustness to background clutter, occlusion, and large changes of viewpoints. Experimentally we show that it supports near real-time recognition of hundreds of objects with state-of-the-art recognition rates. After the test image is processed (in a second on a current PCs), the recognition via indexing into the visual memory requires milliseconds.

1 Introduction

In recent years, research in object recognition has progressed rapidly. Methods based on correspondences of invariantly detected regions have achieved robustness to background clutter, occlusion, and large changes of viewpoint. Impressive results, albeit for certain classes of objects, have been reported [13, 2, 4].

Realistic approaches to recognition, detection and localisation of objects from large collections must support sub-linear indexing, i.e. the ability to associate current visual input with objects represented in the memory, at a speed that does not significantly depend on the number of images and objects already seen. Any technique that compares the current visual input one-by-one with stored models is linear in the number of known objects. Such recognition techniques, solving effectively a sequence of two-image matching problems, will have, sooner or later, an unacceptable response time. Searching and indexing are well-studied subjects, and two sub-linear methods dominate the field – hashing and tree search. This paper presents an approach that achieves sub-linear, real-time recall by

[*] The authors were supported by the Czech Science Foundation under project GACR 102/03/0440 and by the European Commission under project IST-004176.

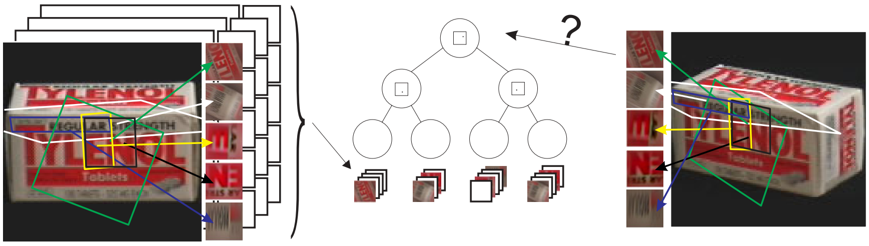

Figure 1: An example of features stored in the visual memory.

representing the visual memory as a binary decision tree organised to minimise average time to decision.

The novel features of the proposed method, and the method itself, is easier explained if the reader is familiar with the state-of-the-art approach of D. Lowe [4]. In Lowe's recognition method, processing of an image starts by extracting square patches invariant to similarity transformations. Next, each patch is described by the SIFT feature – a 128-dimensional vector consisting of sixteen eight-bin weighted histograms of gradient orientations. In the training stage, descriptors from all training images are organised in a kD-tree. Correspondence between patches from the training images and a query image are established as follows. First, descriptors are computed on the query image. For each descriptor, the kD-tree returns one stored patch if its descriptor is significantly closer to the queried descriptor than other stored descriptors, else no match is reported. The matched pairs of query and kD-tree patches form tentative correspondences, which are confirmed or rejected in subsequent verification and consistency checks (these are not relevant for this paper). Lowe's method can be summarised as: (i) detect local patches in an invariant manner, (ii) represent them by a fix-sized feature vectors, (iii) search efficiently for nearest neighbours of the vectors in the kD-tree and (iv) find geometrically consistent groups of correspondences of local coordinate frames.

The first step of the proposed LAF-TREE method is in principle the same as in Lowe's approach. We chose a different type of transformation covariant regions – the maximally stable extremal regions (MSERs [6]), but any affine (scale) invariant processes could be used[1]. The LAF-TREE method establishes local coordinate systems (local affine frames – LAFs) by constructions described in [10].

The novelty of the LAF-TREE approach is in the departure, in Step (ii), from the "compute a fixed-size feature vector on a fixed patch (= measurement region [7])" paradigm. In this paradigm, the local reference frame is described by a function of pixel values from a patch whose size and shape, if expressed in the local frame coordinates, is fixed. In Lowe's approach, the shape is a square of a predefined size. It is clear that a fixed measurement region will lead to difficulties when recognising certain classes of objects, e.g. "wire-like" objects as bicycles where any square neighbourhood includes background. Perhaps more significantly, a measurement region of a certain size will be too big for some frames, e.g. including parts of background or discontinuities, and yet it will be too small for other frames whose descriptors will not be discriminative.

The problem of a better-than-fixed measurement region seems insurmountable. How

[1] Executables of a number of covariant region detectors (including MSERs) are available on the web at `http://www.robots.ox.ac.uk/~vgg/research/affine`

can possibly be the measurement region adapted unless we know what we are looking at? *We finesse the problem by interleaving the processes of recognising the frame and deciding where to measure next.* The frame is recognised by descending a decision-measurement tree where each decision not only reduces the number of potential corresponding frames represented in the tree, but also defines which measurements are taken next. More precisely, a binary tree is formed in the learning stage. For each non-terminal node, a binary valued measurement-decision function, called a 'weak classifier', is selected from a large pool according to an optimisation criterion. The criterion is a lower bound on the expected time to decision. The term 'weak classifier' stresses the obvious analogies with discrete AdaBoost - a classifier is selected by a greedy algorithm, it could be any binary function of pixel values and, as will be shown later, it is not required to make unequivocal decisions.

Establishing tentative correspondences with the decision-measurement tree has a number of favourable properties. The advantages of a data-specific measurement region have already been mentioned. From a computational point of view, efficiency of recognition is increased since only a small fraction of potential measurements is evaluated. In case that a measurement is close to a decision boundary, or not available at all as in the case when it is taken from the background, robustness of the search is easily achieved by inserting the learned frame in both subtrees. With this modification, the search in the recognition stage descend always into only a single branch, guaranteeing that a leaf of the tree is reached in $\log(N)$ steps, where N is the number of frame instances stored in the tree. Last but not least, the learning process explicitly takes into account geometric uncertainty and image statistics to minimise the response time (see Section 2). The final recognition step (iv), the verification of the presence of objects by finding geometrically consistent groups of correspondences, is not time-critical, since the number of tentative matches per frame is small. It can be implemented by RANSAC or a voting scheme.

Related work. Decision trees were successfully applied to various recognition problems, see e.g. [8] for a survey. Our problem differs from the bulk of published work in not having the data labelled. Unsupervised learning techniques are exploited, the trees provide an automatic clustering of image areas by their visual similarity. Our work was inspired by Lepetit and Fua [2], whose approach, however, differs in several areas. First, they set the tree size (and the number of trees, since they are using multiple randomised trees) by hand, while in our approach the tree size is a function of image database content. Second, our measurements are invariant to affine deformations of the image (due to the LAF constructions), thus a 3D model, or synthetically warped 2D images capturing the appearance variations, are not needed. We also explicitly consider image noise and background segmentation of the measurements, while Lepetit et al. synthetically generate noisy patches and patches with random background. Finally we present experiments on datasets containing hundreds of objects, while the method of Lepetit (and also of Lowe) have been demonstrated on only a few objects.

Marée et al. [5] is using various forms of decision trees to recognise randomly generated image windows. The Video Google system by Šivic and Zisserman [13] is able of indexing of a full-length movie. A clustering of descriptors of local features is employed to reduce the recall time complexity by a constant factor. The work by Nene and Nayar [9] supports real-time recognition using a space slicing search, but it is restricted to segmented objects. Neither object occlusion, cluttered background nor multiple objects in a scene are handled.

2 Recognition with Decision-Measurement Trees

This section describes the decision-measurement tree which is used to represent the ”visual memory”. Generally, a decision tree is a tree structure where a simple test (a weak classifier) is assigned to each non-terminal node. Each leaf corresponds to a volume of observation space, that is defined by the sequence of decisions made on the path from the tree root to that particular leaf. During the recall phase, the tree is traversed according to the decisions at non-terminal nodes, until a leaf node is reached. The elements in the leaf do not necessarily match the query – being in the same volume does not imply proximity – and an additional evaluation of a similarity measure is necessary to distinguish matching and non-matching elements. Recall can be viewed as a sequential reduction of the set of candidate correspondences until a subset of a small predefined cardinality (called 'leaf capacity') is reached. The elements remaining in the subset are sequentially searched for matches.

Although we currently employ only one simple type of weak classifiers, multiple types can be freely combined within the tree. Our classifiers are binary functions $d_{\overline{\mathbf{x}},\Theta_{\overline{\mathbf{x}}}} : A \rightarrow \{\mathrm{L},\mathrm{R}\}$, which threshold a single pixel value. Vector $\overline{\mathbf{x}}$ is specifying the measurement location, $\Theta_{\overline{\mathbf{x}}}$ is a scalar threshold on the value at $\overline{\mathbf{x}}$, A is a local affine frame, and $\{\mathrm{L},\mathrm{R}\}$ are the decisions to search left and right subtrees respectively.

Due to noise, the decisions are ambiguous for values close to the thresholds $\Theta_{\overline{\mathbf{x}}}$. The ambiguity can be solved in the recognition phase by descending both subtrees, as e.g. in the classical kD-tree algorithm. In an alternative approach, the elements are in ambiguous cases stored redundantly in both subtrees. There is then no need to backtrack or split the tree search during recognition (recall), all uncertainties are addressed in the training phase. This approach allows for faster retrieval at the expense of memory needed for the redundant representation. Since our motivation is to achieve high recognition speeds, we have adopted the second approach. The design leads to a straightforward retrieval algorithm (see Algorithm 1). The retrieval is very fast since for each query frame A only one evaluation of a weak classifier (thresholding of a single pixel value) is performed at each tree level. The depth of the tree is typically 15 to 25, depending on the database size.

Learning the tree (Algorithm 2). A separate tree is constructed for every type of LAF construction. Starting with a set S_A of frames of a single type, the set is recursively divided into subsets at non-terminal nodes. Non-terminal nodes are inserted until (a) the cardinality of the particular subset is below a predefined threshold, the 'leaf capacity', or (b) the frames in the subset are indistinguishable. The condition (b) accommodates for the situation where there are multiple images of the same object in the database, or when the objects contain repetitive structures.

Let $r(A)$ denote a random realisation of frame A in a query image. The random function r encapsulates geometric and photometric misalignments between corresponding frames, as well as image noise, blur and other image distortions. Algorithm 2 ensures that A is represented in every leaf where the probability of a query realisation $r(A)$ falling to that leaf is above a threshold Θ_p; Θ_p is a parameter of the method. The probability that a query realisation $r(A)$ of frame A will descend the left $\big(p(d_{\overline{\mathbf{x}},\Theta_{\overline{\mathbf{x}}}}(r(A)) = \mathrm{L})\big)$, and right $\big(p(d_{\overline{\mathbf{x}},\Theta_{\overline{\mathbf{x}}}}(r(A)) = \mathrm{R})\big)$ subtree respectively, given a classifier $d_{\overline{\mathbf{x}},\Theta_{\overline{\mathbf{x}}}}$ and the frame A is analysed below.

Estimation of geometric precision of the MSER-LAF method. Local affine frames were constructed on several image pairs related by known homographies. Corresponding

Algorithm 1: MSER-LAF-TREE: Retrieving stored frames

```
Input:
  $A$: a query local affine frame
Output:
  $S$: a set of candidate matches

function Tree.retrieveFrames $(A) \rightarrow S$
  $S :=$ root.retrieveFrames $(A)$

function Node.retrieveFrames $(A) \rightarrow S$
  if isLeaf then
    $S := \{A_i : A_i \in \text{leafFrames} \wedge \text{similar}(A, A_i)\}$
  else
    if $d_{\bar{\mathbf{x}},\Theta_{\bar{\mathbf{x}}}}(A) = \mathrm{L}$ then
      $S :=$ leftSubtree.retrieveFrames $(A)$
    else /*$d_{\bar{\mathbf{x}},\Theta_{\bar{\mathbf{x}}}}(A) = \mathrm{R}$*/
      $S :=$ rightSubtree.retrieveFrames $(A)$
```

Algorithm 2: MSER-LAF-TREE: Learning

```
Input: $S_A$: Set of LAFs of one type

procedure Tree.build $(S_A)$
  $S := \emptyset$
  for all $A \in S_A$ do
    $S := S \cup \{\{A, 1\}\}$ /*assign unit probability*/
  root.build $(S)$

procedure Node.build $(S)$
  if $|S| \leq$ leaf capacity or indistinguishable $(S)$
  then
    isLeaf := true, leafFrames := $S$
  else
    $d_{\bar{\mathbf{x}},\Theta_{\bar{\mathbf{x}}}} :=$ selectClassifier $(S)$
    $S_L = \emptyset, S_R = \emptyset$
    for all $\{A, p_A\} \in S$ do
      $p_L := p_A \cdot p(d_{\bar{\mathbf{x}},\Theta_{\bar{\mathbf{x}}}}(r(A)) = \mathrm{L})$
      $p_R := p_A \cdot p(d_{\bar{\mathbf{x}},\Theta_{\bar{\mathbf{x}}}}(r(A)) = \mathrm{R})$
      if $p_L \geq \Theta_p$ then
        $S_L := S_L \cup \{\{A, p_L\}\}$
      if $p_R \geq \Theta_p$ then
        $S_R := S_R \cup \{\{A, p_R\}\}$
    if $S_L \neq \emptyset$ then
      leftSubtree.build $(S_L)$
    if $S_R \neq \emptyset$ then
      rightSubtree.build $(S_R)$
```

frames do not align perfectly – a single spot in the scene occurs at slightly different pixel positions. Figure 2 shows covariance matrices of distributions of pixel displacements, estimated on thousands of frames. The distributions represent a localisation uncertainty $l_{\bar{\mathbf{x}}}$ of pixels in query frames. As expected, the farther from the detected frame, the larger is the uncertainty. It is clear that the distributions differ significantly for different types of frame constructions. A separate set of distributions is therefore maintained for each frame type.

Considering the estimated geometric uncertainty. Given a database frame A of a certain type, what is the probability of observing value v at measurement position $\bar{\mathbf{x}}$ in a corresponding query frame $r(A)$? The situation is depicted in Figure 3. Fig. 3(a) illustrates a part of the frame neighbourhood around measurement position $\bar{\mathbf{x}}$, and the corresponding distribution of localisation uncertainty $l_{\bar{\mathbf{x}}}$ for that particular frame type. The probability $p(v)$ of observing a value v in a query frame at position $\bar{\mathbf{x}}$ is given as

$$p_{\bar{\mathbf{x}},A}(v) = \int_{\Omega_{v,A}} l_{\bar{\mathbf{x}}} \, \mathrm{d}\Omega, \tag{1}$$

where $\Omega_{v,A}$ is the area in A covered by pixels of value v. Fig. 3(b) shows the resulting distribution $p_{\bar{\mathbf{x}},A}(v)$ for the example from Fig. 3(a). Narrow distributions of $p_{\bar{\mathbf{x}},A}(v)$, which are benign for unambiguous decisions about query frames, are intuitively obtained either in areas of uniform intensity or where the localisation is precise.

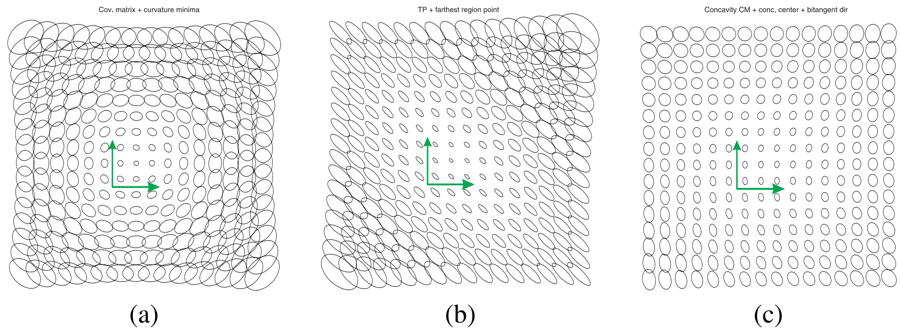

Figure 2: Geometric misalignment of detected frames, experimentally obtained for different types of frame constructions. The images show covariance matrices of distributions of displacements of pixels in normalised neighbourhoods of detected LAFs. (a) LAF construction based on normalisation by region covariance matrix, (b) LAF construction based on a bi-tangent segment, (c) LAF construction based on normalisation by covariance matrix of a concavity [10]

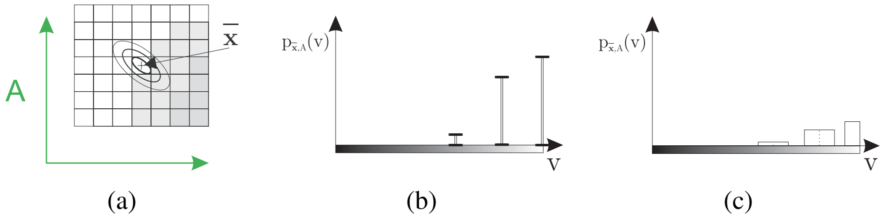

Figure 3: Probability of observing value v at position $\bar{\mathbf{x}}$ in a query realisation of frame A. (a) localisation uncertainty $l_{\bar{\mathbf{x}}}$ for a pixel at position $\bar{\mathbf{x}}$, (b) probability $p_{\bar{\mathbf{x}},A}(v)$ of value v, (c) the probability after considering photometric noise

The framework also consistently handles situations when some of the measurements are undefined, e.g. because not being on the object. Imagine hand-segmented model images where the outline of the object is available (as in Figure 4(a)). Some of the frames will partially cover an area not on the object. In this area, the model cannot predict what value v will occur in a query frame. Without a background model (the probability distribution of intensities in the scene background), all values v are considered equiprobable. That is, if $\bar{\mathbf{x}}$ is known to be outside of the object, $p_{\bar{\mathbf{x}},A}(v)$ has flat distribution over the whole domain of v $\big(p_{\bar{\mathbf{x}},A}(v) = 1/256$ for $v \in \{0,\ldots,255\}\big)$.

Modelling photometric noise. A very simple model of photometric noise is employed – the noise distribution is assumed to be flat in a range of $(-\varepsilon,\varepsilon)$ intensity values. As illustrated in Figure 3(c), the probability of observing value v becomes $p_{\bar{\mathbf{x}},A}(v)/\varepsilon$ over the ε-range. In the experiments, ε is set to 10, independently of v.

Going back to Algorithm 2, the probabilities that a query realisation $r(A)$ will descend into the left and right subtree respectively are expressed as

$$p\big(d_{\bar{\mathbf{x}},\Theta_{\bar{\mathbf{x}}}}(r(A)) = \mathrm{L}\big) = \int_0^{\Theta_{\bar{\mathbf{x}}}} p_{\bar{\mathbf{x}},A}(v)\,\mathrm{d}v, \text{ resp. } p\big(d_{\bar{\mathbf{x}},\Theta_{\bar{\mathbf{x}}}}(r(A)) = \mathrm{R}\big) = \int_{\Theta_{\bar{\mathbf{x}}}}^{255} p_{\bar{\mathbf{x}},A}(v)\,\mathrm{d}v \quad (2)$$

for $v \in \{0\ldots255\}$.

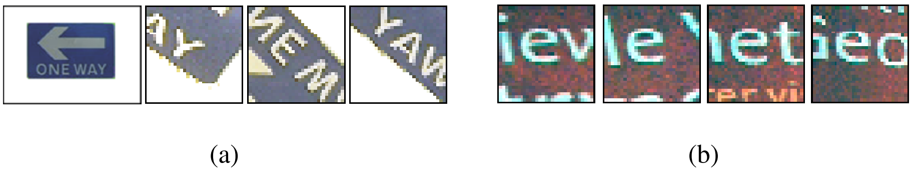

Figure 4: The need for variable-sized measurement regions. (a) An example of a segmented model image and some of its frames. Using a common fixed measurement region where values are defined for all frames would lead to small nondescriminative descriptors. Large regions would include background in test images. (b) Frames detected on multiple instances of the 'e' letter on the 'Multiple view geometry' book title. The instances cannot be distinguished close to the detected frames and a distant measurement (e.g. on a neighbouring letter) is needed to separate them.

The remaining issue in the tree construction algorithm is the choice of weak classifiers for non-terminal nodes. The objective is to minimise the expected recall time for query frames. To select the classifier for a non-terminal node, let us have a set S of frames A, each with assigned probability p_A – the probability that $r(A)$ will descend from root to that node. The task is to select a measurement position $\overline{\mathbf{x}}$ and a threshold $\Theta_{\overline{\mathbf{x}}}$ so that, on average, the queries reach leaf nodes in minimal time, i.e. on minimal tree level. The requirements translate to (a) that the tree is balanced for query frames and (b) the number of ambiguous frames stored in *both* subtrees is minimised. It follows from (a) that for any particular $\overline{\mathbf{x}}$, the threshold $\Theta_{\overline{\mathbf{x}}}$ is set to median value, so that

$$\begin{aligned} \sum_{A\in S} p_A\, p\big(d_{\overline{\mathbf{x}},\Theta_{\overline{\mathbf{x}}}}(r(A)) = \mathrm{L}\big) &= \sum_{A\in S} p_A\, p\big(d_{\overline{\mathbf{x}},\Theta_{\overline{\mathbf{x}}}}(r(A)) = \mathrm{R}\big) \\ \sum_{A\in S} p_A \int_0^{\Theta_{\overline{\mathbf{x}}}} p_{\overline{\mathbf{x}},A}(v)\,\mathrm{d}v &= \sum_{A\in S} p_A \int_{\Theta_{\overline{\mathbf{x}}}}^{255} p_{\overline{\mathbf{x}},A}(v)\,\mathrm{d}v \end{aligned} \tag{3}$$

The measurement position $\overline{\mathbf{x}}$ that best separates (minimises overlap) of the frames in S is selected as

$$\overline{\mathbf{x}} = \operatorname*{argmin}_{\overline{\mathbf{x}}} \sum_{A\in S} \min\Big(p_A\, p\big(d_{\overline{\mathbf{x}},\Theta_{\overline{\mathbf{x}}}}(r(A)) = \mathrm{L}\big), p_A\, p\big(d_{\overline{\mathbf{x}},\Theta_{\overline{\mathbf{x}}}}(r(A)) = \mathrm{R}\big)\Big), \tag{4}$$

with $\Theta_{\overline{\mathbf{x}}}$ given by Eq. 3. Ideally, when a position $\overline{\mathbf{x}}$ (and a corresponding threshold $\Theta_{\overline{\mathbf{x}}}$) is found which perfectly separates the set S, the minimised term evaluates to zero. In the worst case of identical distributions $p\big(d_{\overline{\mathbf{x}}}, \Theta_{\overline{\mathbf{x}}}(r(A))\big)$ for all $A \in S$, the term evaluates to 0.5 (after normalisation by $\frac{1}{|S|}$). Let us consider the example shown in Figure 4 (b). No measurement positions $\overline{\mathbf{x}}$ on the letter 'e' nor the brown background will allow for discrimination of the frames. Due to formula 4, a distant but discriminative measurement is rather selected.

3 Experiments

The performance of the proposed method, both in the recognition rate and execution speed, was evaluated on two datasets. The COIL-100 dataset has been widely used in object recognition literature [14, 10, 3, 1, 15], and the experiment is included to compare the recognition rate with other state-of-the-art methods. ZuBuD, the second dataset, represents a larger, real-world problem, with images taken outdoor, with occluded objects, varying background, and illumination changes.

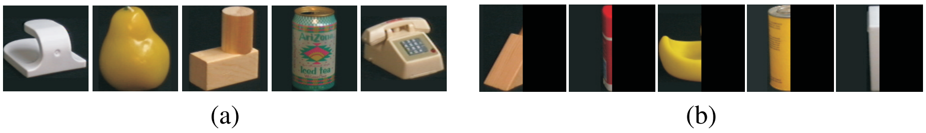

Figure 5: COIL-100: (a) Objects from the database, (b) Query images for the occlusion experiment

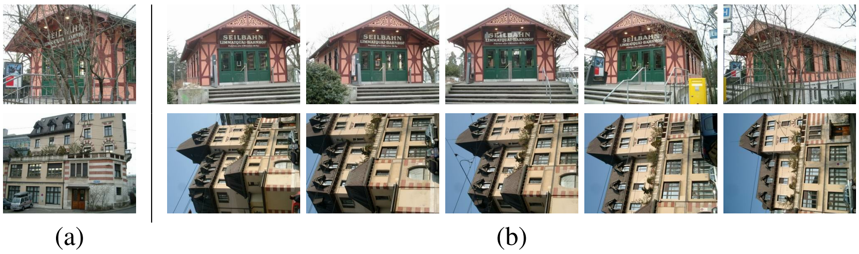

Figure 6: ZuBuD dataset [12]: Examples of (a) query and (b) the corresponding database images.

COIL-100. The Columbia Object Image Library (COIL-100)[2] is a database of colour images of 100 different objects; 72 images of each object placed on a turntable were acquired at pose intervals of 5°. Neither occlusion, background clutter, nor illumination changes are present. Several images from the database are shown in Figure 5(a). Two experiments were performed, differing in the number of images used for training. The achieved recognition rate was 98.2% for 4 training views per object (90° apart, 68 test views per object) and 99.7% for 8 training views (45° apart, 64 test views). Table 1 summarises the results and provides comparison to other published results.

In another experiment, occlusion of the objects was simulated by blanking one half of the test images (see Figure 5 (b)). Four full (unoccluded) training views per object were used in training. The recognition rate was 87%, which is comparable to published results on unoccluded images.

Table 2 provides detailed information about the experiments. Two variants of the recognition system were evaluated, one which recalls the stored frames via the proposed decision tree (with sub-linear recall time), and a second one which sequentially scans through all stored frames (linear recall time). The recall times in the Table 2 show that using the decision tree, matching of approximately 500 query frames against hundreds of thousands of stored frames takes about 2 milliseconds. The total response time of the recognition system is the sum of the time needed to build the query image representation (independent of the number of database objects – 7th row of Table 2) and the recall time (8th or 9th row). Note that doubling the number of training images (columns 2 and 3) did not double the recall time for the tree approach. The required time increased from 1.99 ms to 2.17 ms, i.e. by less than 10%. It confirms the claim that the recall time is sub-linear

[2] http://www.cs.columbia.edu/CAVE

in the number of stored frames. Training of the tree took approximately 30 hours and the tree representation required approximately 1GB of memory.

Method	8 views	4 views	Method	8 views	4 views
MSER+LAF+tree (proposed)	99.8%	98.2%			
MSER+LAF 2002 [10]	99.4%	94.7%	Spectral representation [3]	96.3%	–
Kullback-Leibler SVM [14]	95.2%	84.3%	SNoW / edges [15]	89.2%	88.3%
Spin-Glass MRF [1]	88.2%	69.4%	SNoW / intensity [15]	85.1%	81.5%
Linear SVM [15]	84.8%	78.5%	Nearest Neighbour [15]	79.5%	74.6%

Table 1: COIL-100 experiment: Comparison with published results

	COIL-100			ZuBuD
1. Occluded queries	no	no	yes	n/a
2. Training view dist	90°	45°	90°	n/a
3. Number of DB images	400	800	400	1005
4. Number of DB frames	186346	385197	186346	251633
5. Number of query images	6800	6400	6800	115
6. Avg number of query frames	494	494	269	1594
7. avg time to build representation	520 ms	522 ms	251 ms	1255 ms
8. avg recall time without the tree	493 ms	3471 ms	277 ms	27234 ms
9. avg recall time with the tree	1.99 ms	2.17 ms	1.07 ms	14.3 ms
10. recognition rate	98.24%	99.77%	87.01%	93 %

Table 2: Experimental results on COIL-100 and ZuBuD datasets

ZuBuD dataset. The experiment was conducted on a set of images of 201 buildings in Zurich, Switzerland, which is publicly available [12]. The database consists of five photographs of the 201 buildings. A separate set of 115 query images is provided. For every query image, there are exactly five matching images of the same building in the database. Query and database images differ in viewpoint, variations in the illumination are present, but rare. Examples of corresponding query and database images are shown in Figure 6. Experimental results are summarised in the last column of Table 2. The slower recall times, compared with the COIL-100 dataset, are caused by a higher number of query frames and by the increase of the leaf capacity from 4 to 10 – up to 10 frames were searched exhaustively in the leaf nodes. The leaf capacity represents a trade-off between recall speed and recognition rate. Setting the capacity to 1000, a recognition rate of 98.2% was achieved, but the average recall time dropped to 510 ms. Linear exhaustive scan through all the stored frames (avoiding the tree) achieved recognition rate of 100% [11], but with recall times over 27 seconds per image.

4 Conclusions

An object recognition method capable of sub-linear recall has been proposed. Objects are represented by local affine frames, i.e. as a set of local photometric measurements expressed in object-centred coordinates. The local affine frames are stored in a binary decision-measurement tree organised to minimise average time to decision. A frame is recognised by descending the tree where each decision not only reduces the number of potential corresponding frames, but also defines which measurements are taken next.

We show experimentally that the method supports near real-time recognition of hundreds of real-world objects with state-of-the-art recognition rates. Establishing correspondences between hundreds of query local frames and hundreds of thousands of stored frames takes only a few milliseconds. The proposed LAF-TREE method possesses all the strengths of local region methods – robustness to background clutter, occlusion, and large changes of viewpoints.

References

[1] B. Caputo, J. Hornegger, D. Paulus, and H. Niemann. A spin-glass markov random field for 3-D object recognition. Technical Report LME-TR-2002-01, Lehrstuhl für Mustererkennung, Institut für Informatik, Universität Erlangen-Nürnberg, 2002.

[2] V. Lepetit, P. Lagger, and P. Fua. Randomized trees for real-time keypoint recognition. In *Computer Vision and Pattern Recognition*, June 2005.

[3] X. Liu and A. Srivastava. A spectral representation for appearance-based classification and recognition. In *ICPR (1)*, pages 37–40, 2002.

[4] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *Int. J. Comput. Vision*, 60(2):91–110, 2004.

[5] R. Marée, P. Geurts, J. Piater, and L. Wehenkel. Random subwindows for robust image classification. In *CVPR '05*, 2005.

[6] J. Matas, O. Chum, M. Urban, and T. Pajdla. Robust wide baseline stereo from maximally stable extremal regions. In *BMVC 2002*, volume 1, pages 384–393, London, UK, 2002.

[7] K. Mikolajczyk, T. Tuytelaars, C. Schmid, A. Zisserman, J. Matas, F. Schaffalitzky, T. Kadir, and L. V. Gool. A comparison of affine region detectors. *Accepted to IJCV*, 2005.

[8] S. K. Murthy. Automatic construction of decision trees from data: A multi-disciplinary survey. *Data Mining and Knowledge Discovery*, 2(4):345–389, 1998.

[9] S. A. Nene and S. K. Nayar. A simple algorithm for nearest neighbor search in high dimensions. *IEEE Trans. Pattern Anal. Mach. Intell.*, 19(9):989–1003, 1997.

[10] Š. Obdržálek and J. Matas. Object recognition using local affine frames on distinguished regions. In *The British Machine Vision Conference (BMVC02)*, September 2002.

[11] Š. Obdržálek and J. Matas. Image retrieval using local compact dct-based representation. In *DAGM 2003: Proceedings of the 25th DAGM Symposium*, pages 490–497, 9 2003.

[12] H. Shao, T. Svoboda, and L. Van Gool. ZuBuD — Zurich Buildings Database for Image Based Recognition. Technical Report 260, Computer Vision Laboratory, ETH, Switzerland, 2003. http://www.vision.ee.ethz.ch/showroom/zubud.

[13] J. Šivic and A. Zisserman. Video Google: A text retrieval approach to object matching in videos. In *ICCV03*, pages 1470–1477, 2003.

[14] N. Vasconcelos, P. Ho, and P. J. Moreno. The Kullback-Leibler kernel as a framework for discriminant and localized representations for visual recognition. In *ECCV '04*, 2004.

[15] M. H. Yang, D. Roth, and N. Ahuja. Learning to Recognize 3D Objects with SNoW. In *ECCV 2000*, pages 439–454, 2000.