Real-Time Visual Recognition of Objects and Scenes Using P-Channel Matching^{*}

Michael Felsberg and Johan Hedborg

Computer Vision Laboratory, Linköping University, S-58183 Linköping, Sweden
mfe@isy.liu.se
<http://www.cvl.isy.liu.se/~mfe>

Abstract. In this paper we propose a new approach to real-time view-based object recognition and scene registration. Object recognition is an important sub-task in many applications, as e.g., robotics, retrieval, and surveillance. Scene registration is particularly useful for identifying camera views in databases or video sequences. All of these applications require a fast recognition process and the possibility to extend the database with new material, i.e., to update the recognition system online.

The method that we propose is based on P-channels, a special kind of information representation which combines advantages of histograms and local linear models. Our approach is motivated by its similarity to information representation in biological systems but its main advantage is its robustness against common distortions as clutter and occlusion. The recognition algorithm extracts a number of basic, intensity invariant image features, encodes them into P-channels, and compares the query P-channels to a set of prototype P-channels in a database. The algorithm is applied in a cross-validation experiment on the COIL database, resulting in nearly ideal ROC curves. Furthermore, results from scene registration with a fish-eye camera are presented.

Keywords: object recognition, scene registration, P-channels, real-time processing, view-based computer vision.

1 Introduction

Object and scene recognition is an important application area of methods from image processing, computer vision, and pattern recognition. Most recognition approaches are based on either of the following two paradigms: Model-based recognition or view-based recognition. As we believe that view-based recognition is better motivated from biological vision, we focus on the latter.

Hence, the problem that we consider in this paper is the following: Recognize a previously seen object or scene with a system which has seen many objects or

---

^{*} This work has been supported by EC Grants IST-2003-004176 COSPAL and IST-2002-002013 MATRIS. This paper does not represent the opinion of the European Community, and the European Community is not responsible for any use which may be made of its contents.

scenes from different poses. In case of scene recognition, different scenes rather means the same setting seen from different view angles and the task is to find a view with an approximately correct view angle. We consider both problems in this paper as we believe that there are many similarities in the problems and that our proposed approach solves both issues.

A further side condition of the problems is that they have to be solved in real-time. For object recognition, the real-time requirement is dependent on the task, but for scene recognition, video real-time is required, i.e., recognition in images with PAL resolution at 25 Hz. The real-time requirement rules out many powerful approaches for recognition. The situation becomes even worse if even the *learning has to be done on the fly* in terms of constantly adding new objects or scenes to the database. The latter requirement disqualifies all methods which rely on computationally expensive data analysis during a batch mode learning stage.

In the literature a vast amount of different recognition techniques are proposed, and we do not intend to give an exhaustive overview here. One prominent member of recognition systems is the one developed by Matas et. al., see e.g. [1] for a contribution to the indexing problem in the recognition scheme. The main purpose of the cited work is however to recognize objects as good as possible from a single view, whereas we propose a method which recognizes an object or a scene with an approximately correct pose.

This kind of problem is well-reflected by the COIL database [2], where 72 poses of each of the 100 objects are available. The main drawback of the COIL database is that the recognition task is fairly simple and perfect recognition has been reported for a subset COIL (30 images), with 36 views for learning and 36 views for evaluation [3]. This has been confirmed by later results, e.g., [4]. These methods are however not real-time capable and cannot perform on-the-fly learning. Furthermore, the recognition is very much intensity sensitive, as intensity respectively RGB channels are used for recognition. A more recent work [5] reaches real-time performance, but reported a significant decrease of the ROC (receiver operating characteristic) compared to the previously mentioned methods.

Our proposed method combines the following properties:

- Real-time recognition
- On-the-fly learning is possible
- Intensity invariance (to a certain degree)
- Few training views necessary (experiment: 12)
- State-of-the-art recognition performances (ROC)

This is achieved by a very efficient implementation of a sampled density estimator for the involved features hue, saturation, and orientation. The estimator is based on P-channels, a feature representation that is motivated from observations in biological systems. The density is then compared to reference densities by means of a modified Kullback-Leibler divergence, see e.g. [6]. The method for comparing P-channels is the main contribution of this paper. The resulting recognition method performs comparable to other, computationally much more demanding

methods, which is shown in terms of ROC curves for experiments on the COIL database.

2 Methods for Density Estimation

Density estimation is a very wide field and similar to the field of recognitions methods, we do not intend to give an overview here. The interested reader is referred to standard text books as e.g. [7,8]. In this section we introduce a method for non-parametric density estimation, which is significantly faster than standard kernel density estimators and grid-based methods.

2.1 Channel Representations

The approach for density estimation that we will apply in what follows, is based on the biologically motivated channel representation [9,10]. The latter is based on the idea of placing local functions, the *channels*, pretty arbitrarily in space and to project the data onto the channels - i.e., we have some kind of (fuzzy) voting. The most trivial case are histograms, but their drawback of losing accuracy is compensated in the channel representation by knowledge about the algebraic relation between the channels.

The projections onto the channels result in tuples of numbers which - although often written as vectors (boldface letters) - do not form a vector space. In particular the value zero (in each component) has a special meaning, *no information*, and need not be stored in the memory. Note that channel representations are not just a way to re-represent data, but they allow advanced non-linear processing by means of linear operators, see Sect. 2.2.

Formally, the channel representation is obtained from a finite set of *channel projection operators* F_n . These are applied to the feature vectors $\mathbf{f}$ in a point-wise way to calculate the *channel values* p_n :

$$p_n = F_n(\mathbf{f}) \quad n = 1, \dots, N. \quad (1)$$

Each feature vector $\mathbf{f}$ is mapped to a vector $\mathbf{p} = (p_1, \dots, p_N)$, the *channel vector*. If the considered feature is vector-valued, i.e., we would like to encode K feature vectors $\mathbf{f}_k$, we have to compute the outer product (tensor product) of the respective channel vectors $\mathbf{p}_k$:

$$\mathbf{P} = \bigotimes_{k=1}^K \mathbf{p}_k, \quad (2)$$

which is only feasible for small K , since the number of computations scales with a^K if a is the overlap between the channels.

2.2 Relation to Density Estimation

The projection operators can be of various form, e.g., $\cos^2$ functions, B-splines, or Gaussian functions [11]. The channel representation can be used in

different contexts, but typically it is applied for associative learning [12] or robust smoothing [13].

In context of robust smoothing it has been shown that summing B-spline channel vectors of samples from a stochastic variable ξ results in a sampled kernel density estimate of the underlying distribution $p(\xi)$:

$$E\{\mathbf{p}\} = E\{[F_n(\xi)]\} = (B * p)(n) . \quad (3)$$

The global maximum of p is the most probable value for ξ and for locally symmetric distributions, it is equivalent to the maximum of $B * p$. The latter can be approximately extracted from the channel vector $\mathbf{p}$ using an implicit B-spline interpolation [13] resulting in an efficient semi-analytic method. The extraction of the maximum can therefore be considered as a functional inverse of the projection onto the channels.

In what follows, we name the projection operation also *channel encoding* and the maximum extraction *channel decoding*. In [14], advanced methods for channel decoding have been considered, which even allow the reconstruction of a complete density function. In this paper we concentrate on linear B-splines (B_1 -kernels), such that no prefiltering according to [15] is necessary, and we just apply ordinary linear interpolation.

2.3 P-Channels

The idea of P-channels [16] is borrowed from projective geometry where homogeneous coordinates are used to represent translations as linear mappings and where vectors are invariant under global scalings. The P-channels are obtained by dropping the requirements for real-valued and smooth basis functions for channel representations. Instead, rectangular basis functions, i.e., ordinary histograms, are considered. Since rectangular basis functions do not allow exact reconstruction, a second component which stores the offset from the channel center is added. As a consequence, the channels become vector-valued (boldface letters) and the channel vector becomes a matrix (boldface capital).

A set of 1D values f_j is encoded into P-channels as follows. Without loss of generality the channels are located at integer positions. The values f_j are accounted respectively to the channels with the center $[f_j]$, where $[f_j]$ is the closest integer to f_j :

$$\mathbf{p}_i = \sum_j \delta(i - [f_j]) \begin{pmatrix} f_j - i \\ 1 \end{pmatrix} , \quad (4)$$

where δ denotes the Kronecker delta. Hence, the second component of the channel vector is an ordinary histogram counting the number of occurrences within the channel bounds. The first component of the channel vector contains the cumulated linear offset from the channel center.

Let $\mathbf{f}_j$ be a K -dimensional feature vector. The P-channel representation of a set of vectors $(\mathbf{f}_j)_j$ is defined as

$$\mathbf{p}_i = \sum_j \delta(\mathbf{i} - [\mathbf{f}_j]) \begin{pmatrix} \mathbf{f}_j - \mathbf{i} \\ 1 \end{pmatrix}, \quad (5)$$

where $\mathbf{i}$ is a multi-index, i.e., a vector of indices $(i_1, i_2, \dots)^T$, and $[\mathbf{f}]$ means $([f_1], [f_2], \dots)^T$.

The main advantage of P-channels opposed to overlapping channels is the linear increase of complexity with growing number of dimensions. Whereas each input sample affects a^K channels if the channels overlap with a neighbors, it affects only $K + 1$ P-channels. Hence, P-channel representations have a tendency to be extremely sparse and thus fast to compute.

3 Object Recognition Based on P-Channels Matching

The new contributions of this paper are: the special combination of image features, cf. Sect. 3.1, the conversion of P-channels to B_1 -spline channels, cf. Sect. 3.2, and the matching scheme according to the Kullback-Leibler divergence, cf. Sect. 3.3.

3.1 Feature Space

The feature space is chosen in an ad-hoc manner based on the following requirements:

- Fast implementation
- Include color information
- Intensity invariance for a certain interval
- Inclusion of geometric information
- Stability
- Robustness

The first three requirements motivate the use of hue h and saturation s instead of RGB channels or some advanced color model. The value component v , which is not included in the feature vector, is used to derive the geometric information. Stability requirements suggest a linear operator and robustness (sensible behavior outside the model assumptions) induces a simple geometric descriptor. Therefore, we use an ordinary gradient estimate to determine the local orientation in double-angle representation, cf. [17]:

$$\theta = \arg((\partial_x v + i \partial_y v)^2). \quad (6)$$

The feature vector is complemented with explicit spatial coordinates, such that the final vector to be encoded is five-dimensional: hue, saturation, orientation, x-, and y-coordinate. For each image point, such a vector is encoded into P-channels, where we used up to 8 channels per dimension. Note in this context that periodic entities (e.g. orientation) can be encoded in exactly the same way as linear ones (e.g. x-coordinate). Only for the conversion to overlapping channels (see below), the periodicity has to be taken into account.

3.2 Computing Overlapping Channels from P-Channels

In this section, we describe a way to convert P-channels into B_1 -spline channels with linear complexity in the number of dimensions K . This appears to be surprising at first glance, as the volume of non-zero channels seems to increase by a factor of 2^K . This is surely true for a single channel, but in practice data clusters, and the overlapping channels result in making the clusters grow in diameter. As the cluster volume has a linear upper bound, cf. [16], and since isotropic clusters grow sub-linearly with the diameter, one can easily find an upper linear bound. This is however not true for very flat clusters, i.e., clusters which have nearly zero extent in several dimensions.

The efficient computational scheme to obtain linear B-spline channels is based on the separability of the K -D linear B-spline, i.e., the multi-linear interpolation. We start however with a short consideration of the 1D case: Two neighbored P-channels correspond to the local constant (h_1, h_2) respectively linear (o_1, o_2) kernels in Fig. 1. Combining them in a suitable way results in the linear B-spline:

Fig. 1. Left: basis functions of two neighbored P-channels. Right: B_1 -spline.

$$B_1 = \frac{h_1 + h_2}{2} + o_1 - o_2 . \quad (7)$$

The histogram components are averaged (2-boxfilter) and the offset components are differentiated (difference of neighbors), which implies also a shifting of the grid by $1/2$. Consequently, non-periodic dimensions shrink by one channel. For periodic dimensions, one has to replicate the first channel after the last before computing the B_1 -channels and the number of channels remains the same.

In order to convert multi-dimensional P-channels into multi-linear kernels, the histogram components have to be convolved with a multi-dimensional 2-boxfilter, which can be separated into 1D 2-boxfilters. Each offset component has to be convolved with the corresponding gradient operator and 2-boxfilters for all other dimensions than the one of the offset. This can be considered as a kind of div computation known from vector fields.

The resulting channel representation is identical to the one obtained from directly computing an overlapping channel representation with B_1 -functions, if the underlying distribution of the data is locally independent. By the latter we

mean that for each B_1 -function support, the distribution is separable. In practice this means that the equality only holds approximately, but with the benefit of a much faster implementation. The final result is a sampled kernel density estimate with a multi-linear window function.

3.3 Matching of Densities

There are many possible ways to measure the (dis-) similarity of densities. In the standard linear framework, one could apply an SVD to the matrix of all densities in order to obtain the pseudoinverse. This method has the advantage of being the central step in any type of least-squares estimation method, e.g., if the aim is to extract not only the object type but also a pose interpolation. Unfortunately, we are not aware of any SVD algorithms which exploit and maintain the sparseness of a matrix, i.e., the computation becomes fairly demanding.

Until recently, we were not aware of the method for incremental SVD computation [18], and hence, we based our implementation below on the ordinary SVD. This means that our SVD-based recognition method cannot be used in a scenario with online learning, as e.g., in a cognitive system, but it is well suited for partly controlled environments where pose measurements are required, e.g., scene registration for augmented reality.

Same as the SVD, an ad hoc choice of (dis-) similarity measure as, e.g., the Euclidian distance, does not constrain the comparisons of prototype $\mathbf{p}$ and query $\mathbf{q}$ to be based on non-negative components. Due to the special structure of the problem, namely to compare estimates of densities, we believe that the Kullback-Leibler divergence

$$D(\mathbf{p}, \mathbf{q}) = \sum_j p_j \log \frac{p_j}{q_j} \quad (8)$$

is most appropriate, as it combines maintaining sparseness, incremental updating, and non-negativity. In order to speed up the matching, one can precompute the terms that only depend on $\mathbf{p}$, i.e., the negative entropy of $\mathbf{p}$:

$$-H_{\mathbf{p}} = \sum_j p_j \log p_j \quad , \quad (9)$$

such that the divergence can be computed by a scalar product:

$$D(\mathbf{p}, \mathbf{q}) = -H_{\mathbf{p}} - \langle \mathbf{p} | \log \mathbf{q} \rangle \quad . \quad (10)$$

All involved logarithms are typically regularized by adding an $\varepsilon > 0$ to $\mathbf{p}$ respective $\mathbf{q}$.

4 Recognition Experiments

In this section we present two experiments, one quantitative using the COIL database and reporting the ROC curves, and one qualitative experiment for scene registration.

4.1 Experiment on Object Recognition

The learning set for this experiment consists of 12 poses for each object from the COIL database. The test set consists of the remaining 60 views for each object, 6000 altogether. We evaluated the recognition based on three different feature sets: RGB colors (for comparison with [4]), orientation only (according to (6)), and the complete feature set described in Sect. 3.1. For each of these three recognition problems, we applied two different matching schemes: The one using SVD and the one using (10).

For each of the six combinations, we computed the ROC curves (Fig. 2) and their integrals, cf. Tab. 1.

Table 1. Results for object recognition (COIL database), using Kullback-Leibler divergence (KLB) and SVD. Three different features: orientation θ , RGB, and hue, saturation, orientation ($hs\theta$).

Method	ROC integral
KLD, θ	0.9817
SVD, θ	0.9840
KLD, RGB	0.9983
SVD, RGB	0.9998
KLD, $hs\theta$	0.9939
SVD, $hs\theta$	1.0000

As it can be seen from the ROC curves and the integrals, the SVD performs marginally better than the KLD matching, both though close to the ideal result. However, we have not tried to improve the matching results by tuning the channel resolutions. We have fixed the resolutions by experience and practical constraints before the experiment was started. The number of P-channels is kept constant in all experiments.

4.2 Experiment on Scene Recognition

The task considered in this experiment was to find the view to a scene with the most similar view angle, using a fisheye lens. A first run was made on real data without known pose angles, i.e., the evaluation had to be done by hand. A second run was made on synthetic data with known pose angles. Example views for either experiment can be found in Fig. 3.

In either case, recognition rates were similar to those reported for object recognition above. A true positive was only reported if the recognized view had an angle close to the correct one, either by inspection (real data) or measured against the accompanying XML data (synthetic data). It virtually never happened that two false responses were generated in a row, which means with a one-frame delay, the scene is reliably registered.

Fig. 2. ROC curves (semi logarithmic scale) for the six experiments described in the text. Index 1 refers to orientation only, index 2 to RGB, and index 3 to $hs\theta$.

Fig. 3. Examples for fisheye images (scene registration). Left: real fisheye image. Right: synthetic image.

5 Conclusion

We have presented a novel approach to object recognition and scene registration suitable for real-time applications. The method is based on a sampled kernel density estimate, computed from the P-channel representation of the considered image features. The density estimates from the test set are classified according to the SVD of the training set and according to the Kullback-Leibler divergence. Both methods result in nearly perfect ROC curves for the COIL database, where

the SVD approach performs slightly better. The divergence-based method is however suitable for online learning, i.e., adding new views and / or objects on the fly.

References

1. Obdržálek, Š., Matas, J.: Sub-linear indexing for large scale object recognition. In: Clocksin, W.F., Fitzgibbon, A.W., Torr, P.H.S., (eds.): *BMVC 2005: Proceedings of the 16th British Machine Vision Conference*. London, UK, BMVA Vol. 1, pp. 1–10 (September 2005)
2. Nene, S.A., Nayar, S.K., Murase, H.: Columbia object image library (coil-100). Technical Report CUCS-006-96 (1996)
3. Pontil, M., Verri, A.: Support vector machines for 3d object recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 20(6), 637–646 (1998)
4. Roobaert, D., Zillich, M., Eklundh, J.O.: A pure learning approach to background-invariant object recognition using pedagogical support vector learning. In: *IEEE Computer Vision and Pattern Recognition*. vol. 2, pp. 351–357 (2001)
5. Murphy-Chutorian, E., Aboutalib, S., Triesch, J.: Analysis of a biologically-inspired system for real-time object recognition. *Cognitive Science Online* 3, 1–14 (2005)
6. Cover, T.M., Thomas, J.A.: *Elements of Information Theory*. Wiley, New York (1991)
7. Bishop, C.M.: *Neural Networks for Pattern Recognition*. Oxford University Press, New York (1995)
8. Bishop, C.M.: *Pattern Recognition and Machine Learning*. Springer, Heidelberg (2006)
9. Granlund, G.H.: An Associative Perception-Action Structure Using a Localized Space Variant Information Representation. In: *Proceedings of Algebraic Frames for the Perception-Action Cycle (AFPAC)*, Kiel, Germany (September 2000)
10. Snippe, H.P., Koenderink, J.J.: Discrimination thresholds for channel-coded systems. *Biological Cybernetics* 66, 543–551 (1992)
11. Forssén, P.E.: *Low and Medium Level Vision using Channel Representations*. PhD thesis, Linköping University, Sweden (2004)
12. Johansson, B., Elfving, T., Kozlov, V., Censor, Y., Forssén, P.E., Granlund, G.: The application of an oblique-projected landweber method to a model of supervised learning. *Mathematical and Computer Modelling* 43, 892–909 (2006)
13. Felsberg, M., Forssén, P.E., Schar, H.: Channel smoothing: Efficient robust smoothing of low-level signal features. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 28(2), 209–222 (2006)
14. Jonsson, E., Felsberg, M.: Reconstruction of probability density functions from channel representations. In: *Proc. 14th Scandinavian Conference on Image Analysis* (2005)
15. Unser, M.: Splines – a perfect fit for signal and image processing. *IEEE Signal Processing Magazine* 16, 22–38 (1999)
16. Felsberg, M., Granlund, G.: P-channels: Robust multivariate m-estimation of large datasets. In: *International Conference on Pattern Recognition*, Hong Kong (August 2006)
17. Granlund, G.H., Knutsson, H.: *Signal Processing for Computer Vision*. Kluwer Academic Publishers, Dordrecht (1995)
18. Brand, M.: Incremental singular value decomposition of uncertain data with missing values. Technical Report TR-2002-24, Mitsubishi Electric Research Laboratory (2002)