Machine Vision and Applications (2004) 16: 59–63
Digital Object Identifier (DOI) 10.1007/s00138-004-0143-8

Machine Vision and Applications

Efficient pose estimation using view-based object representations

Gabriele Peters

Universität Dortmund, Informatik VII, Otto-Hahn-Str. 16, 44227 Dortmund, Germany
(e-mail: peters@ls7.cs.uni-dortmund.de
homepage: http://ls7-www.cs.uni-dortmund.de/∼peters/)

Published online: 13 July 2004 – © Springer-Verlag 2004

Abstract. We present an efficient method for estimating the pose of a three-dimensional object. Its implementation is embedded in a computer vision system which is motivated by and based on cognitive principles concerning the visual perception of three-dimensional objects. Viewpoint-invariant object recognition has been subject to controversial discussions for a long time. An important point of discussion is the nature of internal object representations. Behavioral studies with primates, which are summarized in this article, support the model of *view-based* object representations. We designed our computer vision system according to these findings and demonstrate that very precise estimations of the poses of real-world objects are possible even if only a small number of sample views of an object is available. The system can be used for a variety of applications.

Keywords: Pose estimation – 3d object recognition – tracking – cognitive modeling

1 Implications from cognition

Each object in our environment can cause considerably different patterns of excitation in our retinae depending on the observed viewpoint of the object. Despite this we are able to perceive that the changing signals are produced by the same object. It is a function of our brain to provide this constant recognition from such inconstant input signals by establishing an internal representation of the object.

There are uncountable behavioral studies with primates that support the model of a view-based description of three-dimensional objects by our visual system. If a set of unfamiliar object views is presented to humans, their response time and error rates during recognition increase with increasing angular distance between the learned (i.e., stored) and the unfamiliar view [5]. This angle effect declines if intermediate views are experienced and stored [20]. The performance is not linearly dependent on the shortest angular distance in three dimensions to the best-recognized view, but it correlates with an "image-plane feature-by-feature deformation distance" between the test view and the best-recognized view [2]. Thus, measurement of image-plane similarity to a few feature patterns seems to be an appropriate model for human three-dimensional object recognition.

Experiments with monkeys show that familiarization with a "limited number" of views of a novel object can provide viewpoint-independent recognition [13].

In a psychophysical experiment subjects were instructed to perform mental rotation, but they switched spontaneously to "landmark-based strategies", which turned out to be more efficient [22].

Numerous physiological studies also give evidence for a view-based processing of the brain during object recognition. Results of recordings of single neurons in the inferior temporal cortex (IT) of monkeys, which is known to be concerned with object recognition, resemble those obtained by the behavioral studies. Populations of IT neurons have been found that respond selectively to only some views of an object and their response declines as the object is rotated away from the preferred view [14].

The capabilities of technical solutions for three-dimensional object recognition continue to lag far behind the efficiency of biological systems. Summarizing, one can say that, for biological systems, object representations in the form of single, but connected, views seem to be sufficient for a huge variety of situations and perception tasks.

2 Description of the vision system

In this section we introduce our approach to learning an object representation, which takes these results about primate brain functions into account.

We automatically generate sparse representations for real-world objects, which satisfy the following conditions:

a1 They are constituted from *two-dimensional* views.

a2 They are *sparse*, i.e., they consist of *as few views as possible*.

a3 They are capable of *performing perception tasks*, especially pose estimation.

Our system consists of a *view representation builder* and an *object representation builder*. They are shown, together with

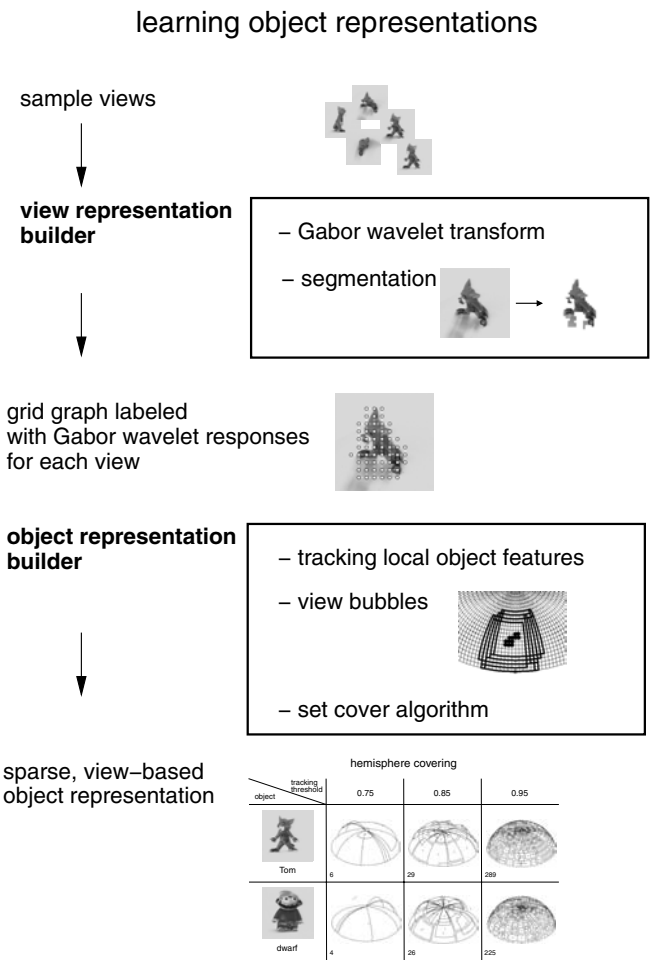


Fig. 1. The system for learning sparse object representations consists of a view and an object representation builder. The resulting object representation consists of single but connected views. The numbers next to the resulting partitionings of the viewing hemisphere are the numbers of view bubbles that constitute the representation

their input and output data, in the diagram in Fig. 1, which depicts a one-directional flow of information.

Of course, feedback from higher levels of processing to lower ones would allow for, e.g., unsupervised system tuning or an improved segmentation, but this is not the subject of this contribution. We start with the recording of a densely sampled set of views of the upper half of the viewing sphere of a test object. In the following disussion we aim at choosing only such views for a representation that are representative for an area of viewpoints as large as possible.

2.1 View representation builder

Each of the recorded views is preprocessed by a *Gabor wavelet transform*, which is biologically inspired because Gabor wavelets approximate response patterns of neurons in the visual cortex of mammals [1,9]. A simple *segmentation* based on [6] utilizing graylevel values follows. It separates the object from the background (in some instances only coarsely, but segmentation is not the main subject here). A regular grid graph is placed on the object segment and the vertices of the graph are labeled with the corresponding Gabor wavelet responses. This results in a representation of each view in the form of a *labeled grid graph*. Each vertex label is a feature vector that describes the local surroundings of the vertex. Such a feature vector is called *jet* [12]. It consists of the amplitude and phase components of the convolution of the image with the Gabor filter bank at the vertex position. A filter bank with wavelets of 8 orientations and 4 frequencies is used.

It has been shown in many studies, e.g., in [12,23], that a representation in the form of a graph labeled with Gabor wavelet responses can be used for a robust object recognition.

2.2 Object representation builder

To facilitate an advantageous selection of views for the object representation, a surrounding area of similar views is determined for each view. This area is called a *view bubble*. For a selected view it is defined as the largest possible surrounding area on the viewing hemisphere for which two conditions hold:

b1 The views constituting the view bubble are *similar* to the view in question.

b2 *Corresponding object points* are known or can be inferred for each view of the view bubble.

The similarity mentioned in **b1** is specified below. Condition **b2** is important for a reconstruction of novel views as, e.g., needed by our pose estimation algorithm. A view bubble may have an irregular shape. To simplify its determination we approximate it by a rectangle with the selected view in its center, which is determined in the following way.

The object representation builder starts by *tracking local object features*. Jets can be tracked from a selected view to neighboring views by utilizing the phase components of jets in successive views [16]. A similarity function $S(\mathcal{G}, \mathcal{G}')$ is defined between a selected view and a neighboring view, where $\mathcal{G}$ is the graph that represents the selected view and $\mathcal{G}'$ is a tracked graph that represents the neighboring view. This similarity function takes the amplitude components of corresponding jets of $\mathcal{G}$ and $\mathcal{G}'$ into account. Utilizing this similarity function we determine a *view bubble* for a selected view by tracking its graph $\mathcal{G}$ from view to view in both directions on the line of latitude until the similarity between the selected view and the tested view either to the west or to the east drops below a threshold τ, i.e., until either $S(\mathcal{G}, \mathcal{G}^w) < \tau$ or $S(\mathcal{G}, \mathcal{G}^e) < \tau$. The same procedure is performed for the neighboring views on the line of longitude, resulting in a rectangular area with the selected view in its center. The representation of a view bubble consists of the graphs of the center and four border views:

$$\mathcal{B} := \langle \mathcal{G}, \mathcal{G}^w, \mathcal{G}^e, \mathcal{G}^s, \mathcal{G}^n \rangle \,, \tag{1}$$

with *w, e, s,* and *n* standing for, respectively, *west, east, south,* and *north*. This algorithm can be extended to obtain view bubbles of a more general shape, e.g., by tracking graphs not only on the lines of latitude and longitude, but also in other directions until the view similarity drops below the threshold. This has not been performed here, but rectangular shapes of view bubbles have proved to be sufficient for the purpose of pose estimation.

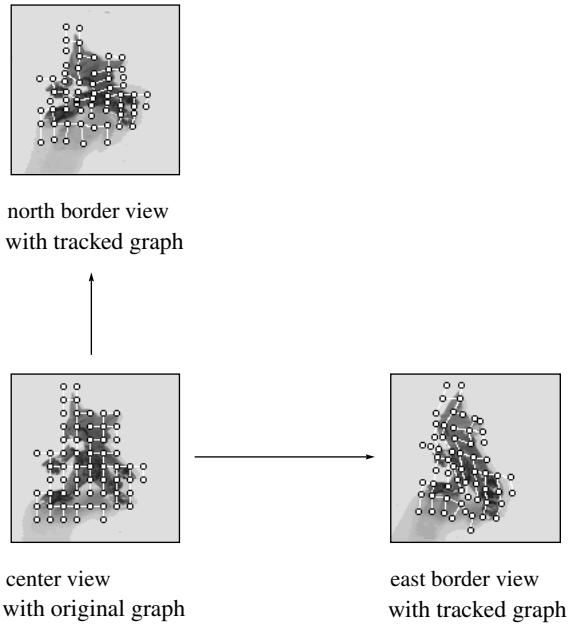


Fig. 2. Graph of the center view of a view bubble tracked to its east and north border views

As this procedure is performed for each of the recorded views, it results in view bubbles overlapping on a large scale on the viewing hemisphere (Figs. 1 and 2).

To meet the first condition **a1** of a sparse object representation, we try to choose single views (in the form of labeled graphs) to constitute it. To meet the second condition **a2**, the large number of overlapping view bubbles must be reduced and as few of them chosen as possible that nevertheless cover the whole hemisphere. For the selection of the view bubbles we use the *greedy set cover algorithm* [3], which successively selects view bubbles that cover the largest number of views that have not been covered previously by other view bubbles. This process is repeated until all views of the hemisphere are covered, and thus a set of view bubbles is provided that covers the whole viewing hemisphere. We define the *sparse, view-based object representation* by

$$\mathcal{R} := \{\langle \mathcal{G}_i, \mathcal{G}_i^w, \mathcal{G}_i^e, \mathcal{G}_i^s, \mathcal{G}_i^n \rangle\}_{i \in R} , \qquad (2)$$

where R is a cover of the hemisphere. Neighboring views of the representation are "connected" by known corresponding object points (the correspondences between center and border views), which have been provided by the tracking procedure. Figure 1 shows different covers of the hemisphere for two test objects.

The definition of the sparse object representation implies that it consists of a larger number of view bubbles for more complex objects, which is also suggested in Fig. 1.

The center views the graphs $\mathcal{G}_i$ are extracted from can be regarded as canonical views [2]. This concept and the relationship between view bubbles and aspect graphs [10] and other view-based object representations are discussed in [17].

3 Pose estimation

Given the sparse representation of the object in question and given a test view of the object, the aim is the determination of the object's pose displayed in the test view, i.e., the assignment of the test view to its correct position on the viewing hemisphere. In this section a solution to this problem is proposed (Sect. 3.1), and the results of simulations with a series of test views are reported (Sect. 3.2) and discussed (Sect. 3.3).

Many approaches to pose estimation have been proposed, starting with closed-form solutions for no more than four non-collinear points [7,4,8] up to iterative nonlinear optimization algorithms, which have to rely on a good initial guess to converge to a reasonable solution [15,24]. More recent approaches to pose estimation also utilize Gabor wavelet representations, e.g., in combination with artificial neutral networks [11].

Here we propose a model-based pose estimation algorithm. In the first step it determines the rough position of the given pose on the viewing hemisphere as an initial guess. Then this estimate is refined in a second step. It requires the generation of *virtual views*, i.e., artificially generated images of unfamiliar views, which are not represented in the object representation. For this purpose we

(1) Calculate linear combinations of corresponding vertex positions in the center and border graphs of view bubbles and

(2) Interpolate the corresponding jets attached to these vertices.

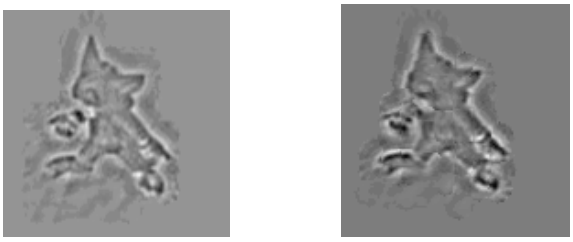
The new positions and jets define a representation graph of the virtual view. From this graph the virtual view can be generated by reconstructing the information contained in Gabor wavelet responses [19]. To interpolate between jets we calculate the weighted sum of corresponding jets in the sample views. The weights are chosen according to the relative position of the unfamiliar view with respect to the sample views. Our method for deriving vertex positions in unfamiliar views follows Ullman and Basri's [21] purely two-dimensional approach of generating unfamiliar views by linear combination of sample views. Detailed formulas are given in [18].

3.1 Methods

Let T be the test view, the pose of which should be estimated, and $\mathcal{G}_T$ its representing graph, which is extracted from the original image of view T after the test view has been divided into object and background segments as described in Sect. 2.1. This means that no a priori knowledge about the object is provided. A view is determined by its position on the viewing hemisphere.

Let $I_i, i \in R$, be the center images of the view bubbles that the graphs $\mathcal{G}_i$ of the object representation $\mathcal{R}$ are extracted from. The *pose estimation algorithm* for estimating the pose of a single test view T proceeds in two steps:

1. Match $\mathcal{G}_T$ to each image $I_i, i \in R$ using a graph matching algorithm [12]. As a *rough estimate* of the object's pose, choose that view bubble $\widehat{B}$ whose center image I_i provides the largest similarity to $\mathcal{G}_T$.
2. Generate the representation $\widehat{\mathcal{G}}$ for discrete, unfamiliar views that are included inside the area defined by $\widehat{B}$ but not represented explicitly. Generate $\widehat{\mathcal{G}}$ by (1) a linear combination of corresponding vertex positions in the center



a b

Fig. 3. a Virtual view $\widehat{V}$ reconstructed from interpolated graph $\widehat{\mathcal{G}}$. **b** Virtual test view $\widehat{V}_T$ reconstructed from its original graph $\mathcal{G}_T$

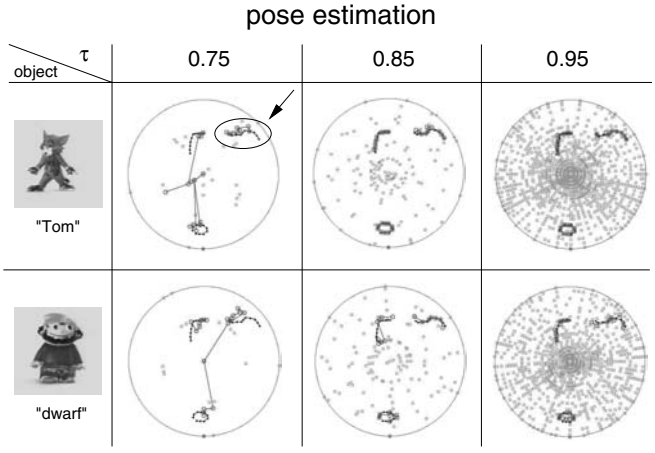


Fig. 4. Results of pose estimations for three different partitionings of the viewing hemisphere and two different objects are depicted. The tracking threshold τ influences the resulting number of views in the final representations. Because the graphs of the center and four border views are stored for each view bubble of the final representation, the border views of neighboring view bubbles lie close together. This is obvious especially for $\tau = 0.75$

and one border graph of $\widehat{B}$ and (2) an interpolation of the corresponding jets as described in Sect. 3. (We choose the graph of that border view that lies closest to the discrete, unfamiliar view. The number of calculated graphs $\widehat{\mathcal{G}}$ depends on the size of $\widehat{B}$.) From each of the calculated graphs $\widehat{\mathcal{G}}$ reconstruct a corresponding virtual view $\widehat{V}$ using an algorithm that reconstructs the information contained in Gabor wavelet responses [19]. Accordingly, reconstruct a virtual test view $\widehat{V}_T$ from $\mathcal{G}_T$ (Fig. 3). Compare each of the virtual views $\widehat{V}$ with the virtual view $\widehat{V}_T$ using an error function $\epsilon(\widehat{V}, \widehat{V}_T)$ that performs a pixelwise comparison between $\widehat{V}_T$ and each $\widehat{V}$. The estimated pose $\widehat{T}$ of the test view T is the position on the viewing hemisphere of that virtual view $\widehat{V}$ that provides the smallest error ϵ.

The *estimation error* between T and $\widehat{T}$ can be determined by the Euclidean distance: $\epsilon_{esti}(T, \widehat{T}) = d(T, \widehat{T})$.

3.2 Results

For the evaluation of the algorithm, 30 test views have been chosen. Their positions on the viewing hemisphere are displayed in Fig. 4. For two different toy objects and for three different partitionings of the viewing hemisphere, which have been derived by applying different tracking thresholds τ, the poses of these 30 test views have been estimated. The light gray squares indicate the views represented in the object representation $\mathcal{R}$, black dots mark the positions of the test images, and the estimated positions are tagged by dark gray circles. The arrow points at the test images and their estimations, which are displayed in Fig. 5.

Table 1. Mean pose estimation errors. For example, for object "Tom" and the partitioning of $\tau = 0.75$ the average estimation deviation of the estimated pose $\widehat{T}$ to the true pose T computed from 30 test views is 36.51°

τ	0.75	0.8	0.85	0.9	0.95
Object "Tom"	36.51°	3.63°	0.77°	3.35°	0.36°
Object "dwarf"	20.54°	19.47°	4.2°	2.65°	1.71°

The illustrations in Fig. 4 indicate that pose estimation becomes more precise with an increasing number of sample views in the object representation. This result was expected and is confirmed by an inspection of the mean estimation errors taken over the 30 test views for each object and each partitioning of the hemisphere separately. They are summarized in Table 1. With one exception for the "object" Tom, the

Fig. 5. Test images and their estimations, which are marked in Fig. 4. For this example the representation of the object "Tom" for $\tau = 0.75$ has been chosen. It consists of only 30 views. In the first row the true poses of the object, which should be estimated, are displayed. The second row shows the poses that have been estimated by treating each view of the sequence independently. The estimation error for this sequence averages 5.78°

mean errors are decreasing with an increasing value of τ, i.e., with an increasing number of views in $\mathcal{R}$.

3.3 Discussion

The results of the pose estimation experiments are quite satisfactory. This is particularly obvious for the example displayed in Fig. 5, taking into account that the sparse representation of the object "Tom" contains only the representations of 30 views. These were the test images for which the best result for $\tau = 0.75$ was obtained, but also for a reasonable partitioning of the viewing hemisphere ($\tau = 0.85$) the mean estimation errors were smaller than 5° for both objects, which can be regarded as a good result, taking into account that humans are hardly able to recognize a difference of 5° between two object poses.

As experiments reported in [17] have shown, the method proposed in Sect. 3.1 cannot be improved very much by a more elaborate determination of the initial guess, e.g., by testing more neighboring candidates. Better results can be expected by applying a more efficient segmentation. In addition, the proposed methods will be applied to more complex objects in the future.

4 Conclusion

We proposed a computer vision system based on cognitive principles that is able to estimate the pose of a three-dimensional object from an unobstructed view in an efficient manner. The pose estimation results support a good quality of our sparse object representation and allow the conclusion that a view-based approach to object perception with object representations that consist of only single, connected views is suitable for performing perception tasks, as is advocated by brain researchers. Besides the biological relevance of our approach, there are a variety of possible applications, such as object recognition, view morphing, or data compression.

References

1. Burr DC, Morrone MC, Spinelli D (1989) Evidence for edge and bar detectors in human vision. Vision Res 29(4):419–431
2. Cutzu F, Edelman S (1994) Canonical views in object representation and recognition. Vision Res 34:3037–3056
3. Chvatal V (1979) A greedy heuristic for the set-covering problem. Math Oper Res 4(3):233–235
4. Dhome M, Richetin M, Lapreste J, Rives G (1989) Determination of the attitude of 3-D objects from a single perspective view. IEEE Trans Patt Anal Mach Intell 11(12):1265–1278
5. Edelman S, Bülthoff HH (1992) Orientation dependence in the recognition of familiar and novel views of three-dimensional objects. Vision Res 32(12):2385–2400
6. Eckes C, Vorbrüggen JC (1996) Combining data-driven and model-based cues for segmentation of video sequences. In: Proc. WCNN96, pp 868–875
7. Horaud R, Conio B, Leboulleux O, Lacolle B (1989) An analytic solution for the perspective 4-point problem. Comput Vision Graph Image Process 47:33–44
8. Haralick RM, Lee C, Ottenberg K, Nölle M (1991) Analysis and solutions of the three point perspective pose estimation problem. In: Proc. IEEE conference on computer vision and pattern recognition, pp 592–598
9. Jones JP, Palmer LA (1987) An evaluation of the two-dimensional gabor filter model of simple receptive fields in cat striate cortex. J Neurophysiol 58(6):1233–1258
10. Koenderink JJ, van Doorn AJ (1976) The singularities of the visual mapping. Biol Cybern 24:51–59
11. Krüger V, Sommer G (2002) Gabor wavelet networks for efficient head pose estimation. Image Vision Comput 20(9–10):665–672
12. Lades M, Vorbrüggen JC, Buhmann J, Lange J, von der Malsburg C, Würtz RP, Konen W (1993) Distortion invariant object recognition in the dynamic link architecture. IEEE Trans Comput 42:300–311
13. Logothetis NK, Pauls J, Bülthoff HH, Poggio T (1994) View-dependent object recognition by monkeys. Curr Biol 4:401–414
14. Logothetis NK, Pauls J, Poggio T (1995) Shape representation in the inferior temporal cortex of monkeys. Curr Biol 5(5):552–563
15. Lowe DG (1987) Three-dimensional object recognition from single two-dimensional images. Artif Intell 31:355–395
16. Maurer T, von der Malsburg C (1996) Tracking and learning graphs and pose on image sequences of faces. In: Proc. international conference on automatic face- and gesture- recognition, pp 176–181
17. Peters G (2002) A view-based approach to three-dimensional object perception. Ph.D. Thesis, Shaker Verlag, Aachen, Germany
18. Peters G, von der Malsburg C (2001) View reconstruction by linear combination of sample views. In: Proc. BMVC 2001, pp 223–232
19. Pötzsch M (1994) Die Behandlung der Wavelet-Transformation von Bildern in der Nähe von Objektkanten. Technical Report IRINI 94-04, Institut für Neuroinformatik, Ruhr-Universität Bochum, Germany
20. Tarr MJ (1993) Orientation dependence in three-dimensional object recognition. Ph.D. Thesis, MIT, Cambridge, MA
21. Ullman S, Basri R (1990) Recognition by linear combinations of models. IEEE Trans Patt Anal Mach Intell 13(10):992–1006
22. Wexler M, Kosslyn SM, Berthoz A (1998) Motor processes in mental rotation. Cognition 68:77–94
23. Wiskott L, Fellous J-M, Krüger N, von der Malsburg C (1997) Face recognition by elastic bunch graph matching. IEEE Trans Patt Anal Mach Intell 19(7):775–779
24. Yuan J (1989) A general photogrammetric method for determining object position and orientation. IEEE J Robot Automat 5(2):129–142