Probabilistic object models for pose estimation in 2D images

Damien Teney[1] and Justus Piater[2]

[1] University of Liège, Belgium
Damien.Teney@ULg.ac.at
[2] University of Innsbruck, Austria
Justus.Piater@UIBK.ac.at

Abstract. We present a novel way of performing pose estimation of known objects in 2D images. We follow a probabilistic approach for modeling objects and representing the observations. These object models are suited to various types of observable visual features, and are demonstrated here with edge segments. Even imperfect models, learned from single stereo views of objects, can be used to infer the maximum-likelihood pose of the object in a novel scene, using a Metropolis-Hastings MCMC algorithm, given a single, calibrated 2D view of the scene. The probabilistic approach does not require explicit model-to-scene correspondences, allowing the system to handle objects without individually-identifiable features. We demonstrate the suitability of these object models to pose estimation in 2D images through qualitative and quantitative evaluations, as we show that the pose of textureless objects can be recovered in scenes with clutter and occlusion.

1 Introduction

Estimating the 3D pose of a known object in a scene has many applications in different domains, such as robotic interaction and grasping [1,6,13], augmented reality [7,9,19] and the tracking of objects [11]. The observations of such a scene can sometimes be provided as a 3D reconstruction of the scene [4], e.g. through stereo vision [5]. However, in many scenarios, stereo reconstructions are unavailable or unreliable, due to resource limitations or to imaging conditions such as a lack of scene texture.

This paper addresses the use of a single, monocular image as the source of scene observations. Some methods in this context were proposed to make use of the appearance of the object as a whole [6,13,15]. These so-called *appearance-based* methods however suffer from the need of a large number of training views. The state-of-the-art methods in the domain rather rely on matching characteristic, local features between the observations of the scene and a stored, 3D model of the object [1,7,17]. This approach, although efficient with textured objects or otherwise matchable features, would fail when considering non-textured objects, or visual features that cannot be as precisely located as the texture patches or geometric features used in the classical methods. Hsiao et al.'s method [8] seeks to better handle multiple possible correspondences between the model and scene features, but still requires a large fraction of exact matches to work efficiently.

The proposed method follows a similar approach to the aforementioned references for modeling the object as a 3D set of observable features, but it is different in the sense that few assumptions are made about the type of features used, and in that it does not rely on establishing specific matches between features of the model and features of the observed scene. For this purpose, we represent both the object model and the 2D observations of a scene as probabilistic distributions of visual features. The model is built from 3D observations that can be provided by any external, independent system. One of the main interests of the proposed method, in addition to the genericity of the underlying principles, is its ability to effectively handle non-textured objects. The general method itself does not make particular assumptions about the type of features used, except that they must have a given, although not necessarily exact, position in space, and they must be potentially observable in a 2D view of the object.

In order to demonstrate the capabilities of the proposed method at handling textureless objects, we apply it to the use of local edge segments as observations. Practically, such features cannot be precisely and reliably observed in 2D images, e.g., due the ambiguity arising from multiple close edges, 3D geometry such as rounded edges, or depth discontinuities that change with the point of view. Such problems motivate the probabilistic approach used to represent the scene observations.

The 3D observations used to build the model are provided by an external system that performs stereopsis on a single pair of images. Such a model can thus be quickly and automatically learned, at the expense of imprecision and imperfections in the model. This again motivates the use of a probabilistic distribution of features as the object model. Other *model-based* methods proposed in the literature have used rigid learned [7,17] or preprogrammed (CAD) models [9,19], but such CAD models are, in general, not available. Our approach for object modeling is more similar to the work of Detry et al. [5], where an object is modeled as a set of parts, themselves defined as probability distribution of smaller visual features. The main contribution of this paper is the extension of those principles to the use of 2D observations.

The representations of the object model and of the scene observations that we just introduced can then be used to perform pose estimation in monocular images, using an inference mechanism. Algorithms such as belief propagation [5] and Metropolis-Hastings MCMC methods [4] were proposed in the literature to solve similar problems, and we adapt the algorithm presented in that last reference to our specific type of model and observations.

Finally, our method provides a rigorous framework for integrating evidence from multiple views, yielding increased accuracy with only a linear increase of computation time with respect to the number of views. Using several views of a scene is implicitly accomplished when using a stereo pair of images, together with a method operating on 3D observations [5]. However, our approach does not seek matches between the two images, as stereopsis does, and can thus handle arbitrarily wide baselines. Other methods for handling multiple views with a 2D method have been proposed [2,14]. In these methods however, the underlying process relies on the matching of characteristic features.

2 Object Model

Our object model is an extension of earlier work [4]. For completeness and clarity, the upcoming sections include essential background following this source.

2.1 General form

We use a 3D model that allows us to represent a probabilistic distribution of 3D features that compose the model. These features must be characterized by a localization in the 3D space, and can further be characterized by other observable characteristics, such as an orientation or an appearance descriptor. The model of an object is built using a set

$$M = \left\{ \left(\lambda^{\ell}, \alpha^{\ell} \right) \right\}_{\ell \in [1,n]} \qquad (1)$$

of features, where $\lambda^{\ell} \in \mathbb{R}^3$ represents the location of a feature, and $\alpha^{\ell} \in \mathcal{A}$ is a (possibly zero-element) vector of its other characteristics from a predefined appearance space $\mathcal{A}$. When learning an object model, the set of features M is decomposed into q distinct subsets M_i, with $i \in [1, q]$, which correspond ideally to the different parts of the object. This step allows the pose estimation algorithm presented below to give equal importance to each of the parts, therefore avoiding distinctive but small parts being overwhelmed by larger sections of the object. The procedure used to identify such parts is detailed in [4].

Our method relies on a continuous probability distribution of 3D features to represent the model. Such a distribution can be built using Kernel Density Estimation (KDE), directly using the features of M_i as supporting particles [5,18]. To each feature of M_i is assigned a kernel function, the normalized sum of which yields a probability density function $\psi_i(x)$ defined on $\mathbb{R}^3 \times \mathcal{A}$. The kernels assigned to the features of M_i will depend on the type of these features.

Reusing the distribution of 3D features of part i, ψ_i, and considering an intrinsically calibrated camera, we now define $\psi'_{i,w}$ as the 2D projection onto the image plane of that distribution set into pose w, with $w \in SE(3)$, the group of 3D poses. Such a distribution is defined on the 2D appearance space, which corresponds to $\mathbb{R}^2 \times \mathcal{B}$, where $\mathcal{B}$ is the projected equivalent of $\mathcal{A}$. For example, if $\mathcal{A}$ is the space of 3D orientations, $\mathcal{B}$ would be the space of 2D orientations observable on an image. Similarly, if $\mathcal{A}$ is a projection-independent appearance space of 3D features, $\mathcal{B}$ would be the simple appearance space of direct 2D observations of such features.

Practically, $\psi'_{i,w}$ can be obtained by setting the features of M_i into pose w, and projecting them onto the image plane (Fig. 1c). The resulting 2D features $\in \mathbb{R}^2 \times \mathcal{B}$ can, similarly to the 3D points, be used as particles to support a KDE on that space, using an equivalent projection of the kernels used in 3D.

2.2 Use of edge segments

This paper presents the particular application of the object model presented above to the use of local edge segments as visual features. Those features basically correspond to 3D oriented points, which are characterized, in addition to their localization in 3D, by an orientation along a line in 3D. Therefore, reusing the notations introduced above, the space $\mathcal{A}$, on which the elements α^ℓ are defined, corresponds to the half 2-sphere S^2_+, i.e. half of the space of 3D unit vectors. The kernels used to compose a 3D probability distribution ψ_i can then be decomposed into a position and an orientation part [5,18]. The first is chosen to be a Gaussian trivariate isotropic distribution, and the latter a von Mises-Fisher distribution on S^2_+. The bandwidth of the position kernel is then set to a fraction of the size of the object, whereas the bandwidth of the orientation kernel is set to a constant. The 2D equivalent of those distributions are obtained using classical projection equations. Fig. 2 depicts the correspondence between the 2D and 3D forms of a particle corresponding to an edge segment and its associated kernel.

The visual features used in our implementation are provided by the external Early Cognitive Vision (ECV) system of Krüger et al. [12,16]. This system extracts, from a given image, oriented edge features in 2D, but can also process a stereo pair of images to give 3D oriented edge features we use to build object models (Fig. 1b).

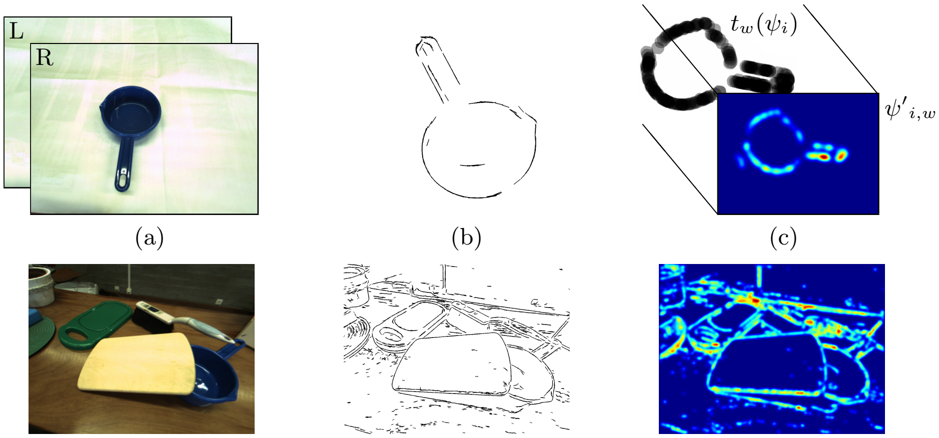
3 Scene observations

The observations we can make of a scene are modeled as a probability distribution in a similar way to the model. The observations are given as a set

$$O = \left\{ \left(\delta^\ell, \beta^\ell \right) \right\}_{\ell \in [1,m]} \tag{2}$$

of features, where $\delta^\ell \in \mathbb{R}^2$ is the position of the feature on the image plane, and $\beta^\ell \in \mathcal{B}$ are its observable characteristics. These characteristics must obviously be a projected equivalent to those composing the object model. Here again, the features contained in O can directly be used as particles to support a continuous probability density, using KDE.

In the particular case of edge segments, the observations correspond to 2D oriented points (Fig. 1e). They are thus defined on $\mathbb{R}^2 \times \mathcal{B}$ with $\mathcal{B} = [0, \pi[$. As mentioned before, the uncertainty on the position and orientation of visual features like edge segments can arise from different sources, and no particular assumptions can thus be made on the shape of their probability distribution. The kernels used here are thus simple bivariate isotropic Gaussians for the position part, and a mixture of two antipodal von Mises distributions for the orientation part. The sum of those kernels, associated with each point of O, then yields a continuous probability density function $\phi(x)$ defined on $\mathbb{R}^2 \times [0, \pi[$ (Fig. 1f).



(a) (b) (c)

(d) (e) (f)

Fig. 1: Proposed method applied to edge segments (orientation of segments not represented). (a) Stereo images used to build object model; (b) 3D edge segments that compose the model; (c) probabilistic model (ψ_i) in pose w, spheres representing the position kernel (their size is set to one standard deviation), and its simulated projection in 2D ($\psi'_{i,w}$; blue and red represent resp. lowest and highest probability densities); (d) image of a scene; (e) 2D edge segments used as observations; (f) probabilistic representation of observations (ϕ).

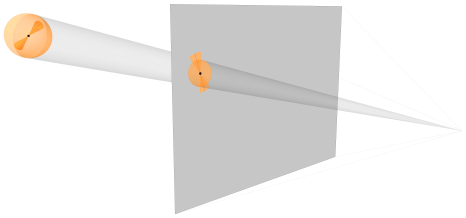
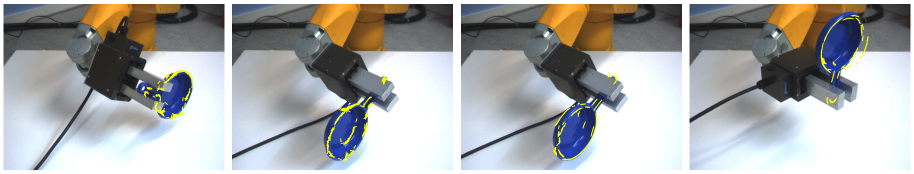


Fig. 2: Correspondence of 3D edge segment and associated kernel, with their 2D projection on image plane. Orange boundaries represent one standard deviation.



(a) (b) (c) (d)

Fig. 3: Results of pose estimation; model features reprojected on input image. (a) Good result (close to ground truth); (b) good result; (c) same frame as (b) with incorrect result, orientation error of about 80°, even though the reprojection matches observations slightly better than (b); (d) incorrect result, insufficient observations extracted from pan bottom, and orientation error of about 180°.

4 Pose estimation

The object and observation models presented above allow us to estimate the pose of a known object in a cluttered scene. This process relies on the idea that the 2D, projected probability distribution of the 3D model defined above can be used as a "template" over the observations, so that one can easily measure the likelihood of a given pose.

Let us consider a known object, for which we have a model composed of q parts M_i ($i \in [1, q]$), which in turn define ψ_i and $\psi'_{i,w}$. On the other hand, we have a scene, defined by a set of observations O, leading to a probabilistic representation ϕ of that scene. We model the pose of the object in the scene with a random variable $W \in SE(3)$. The distribution of object poses in the scene is then given by

$$p(w) \propto \prod_{i=1}^{q} m_i(w) \; , \tag{3}$$

with $m_i(w)$ being the cross-correlation of the scene observations $\phi(x)$ with the projection $\psi'_{i,w}$ of the ith part of the model transformed into pose w, that is,

$$m_i(w) = \int_{\mathbb{R}^2 \times \mathcal{B}} \psi'_{i,w}(x) \, \phi(x) \, \mathrm{d}x \; . \tag{4}$$

Computing the maximum-likelihood object pose $\arg\max_w p(w)$, although analytically intractable, can be approximated using Monte Carlo methods. We extend the method proposed in [4], which computes the pose via simulated annealing on a Markov chain. The chain is defined with a mixture of local- and global-proposal Metropolis Hastings transition kernels. Simulated annealing does not guarantee convergence to the global maximum of $p(w)$, and we thus run several chains in parallel, and eventually select the best estimate. In practice, a strong prior is usually available concerning the distance between the camera and the object, e.g., as information on the scale at which the object can appear in an image. The global transition kernel can benefit from this prior to favor more likely proposals, and therefore drive the inference process more quickly towards the global optimum.

As mentioned above, the proposed method naturally extends to observations from v multiple views. We define $m_{i,j}(w)$ similarly to Eq. 4 but relative to specific views j, $j = 1, \ldots, v$. Accounting for observations from all available views, Eq. 3 then becomes

$$p(w) \propto \prod_{j=1}^{v} \prod_{i=1}^{q} m_{i,j}(w) \; , \tag{5}$$

which is handled by the inference process similarly to the single-view case.

5 Evaluation

This sections presents the applicability of the proposed method for estimating the pose of objects on two publicly available datasets [3,10].

5.1 Experimental setup

In this work, each model is built from one manually segmented stereo view of the object (such as Fig. 1a). The models used here are typically composed of between 1 and 4 parts, containing around 300 to 500 observations in total. Pose estimation is performed on single 1280 × 960 images taken with an intrinsically calibrated camera. The number of parallel inference processes (see Section 4) is set to 16. On a typical 8-core desktop computer, the pose estimation process on a single view typically takes about 20 to 30 seconds. Also, as proposed in Section 4 and detailed below, a crude estimate of the distance between the camera and the object is given as an input to the system.

The ECV observations we use (see Section 2.2) can be characterized with an appearance descriptor composed of the two colors found on the sides of the edge. This appearance information does not enter into the inference procedure. However, in the following experiments we use it to discard those scene observations whose colors do not match any of the model features. This step, although not mandatory, helps the pose estimation process to converge more quickly to the globally best result by limiting the number of local optima.

5.2 Rotating object

We first evaluated our method on a sequence showing a plastic pan undergoing a rotation of 360° in the gripper of a robotic arm [10]. The ground truth motion of the object in the 36 frames of the sequence is thus known. The estimate of the distance to the object, given as input to the system, is the same for the whole sequence, and is a rough estimate of the distance between the gripper and the camera (about 700 mm). Let us note that, for some images of the sequence, this estimate is actually quite different from the exact object-camera distance, since the object is not rotating exactly around its center.

This publicly available dataset is composed of stereo images, and we used the frame corresponding to a rotation of 50° to learn the model, as it gives a good overall view of the object. Four types of experiments were then performed (Fig. 4). First, the pose of the object was estimated in each frame of the sequence, using one single view. One can observe that correct pose estimates can mostly be made close to the viewpoint used for learning the model (Fig. 4). A number of results have an orientation error of almost 180°, which correspond to a special case (Fig. 3d) that can be explained by the flat and almost symmetrical object we consider. Indeed, if very few observations are extracted from the bottom of the pan, only the handle and the top rim of the object can be matched to the image. Another large number of incorrect pose estimates have orientation errors of 70–110°; most of them correspond to ambiguities inherent to a 2D projection, as illustrated on Fig. 3b–c. Similarly, most of the translation errors occur along the camera-object axis, as an inherent limitation of 2D observations. The percentage of correct pose estimates, defined by orientation and translation errors of less than 10° and 30 mm resp., and evaluated over the whole sequence, is only 20%. Second, the same experiment is performed using two views. Some of the ambiguities can then be resolved, and this percentage rises to 60%. This result can be compared to the evaluation of Detry et al. [5] on a similar sequence, which achieved a score of only 40–50%. We stress that the latter method relied on 3D observations computed from stereo, whereas our method uses one or more 2D images directly, and is not limited to short-baseline stereo pairs.

Finally, we used our framework to track the pose of the object over the whole sequence, using one and two views, respectively. The pose is initialized with ground truth information for the first frame, and is then tracked from one frame to the next, using the same process as outlined in Section 4, but without the use of global proposals in the chain, and thus limiting the inference process to a local search. These experiments yield very good results (see Fig. 4), the remaining error being mostly due to the limitations of the model, learned from a single view of the object.

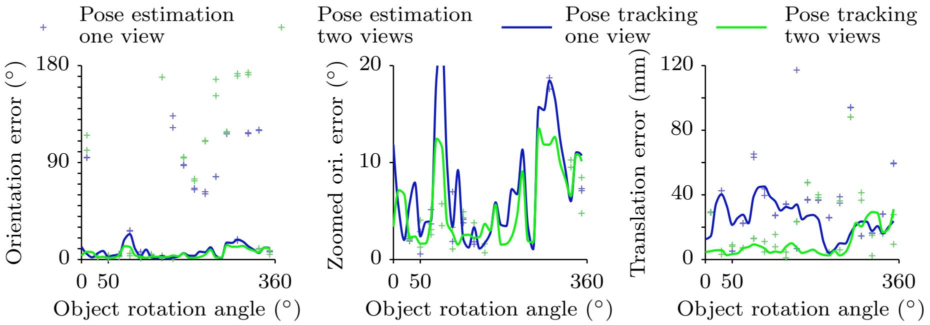


Fig. 4: Results of the "rotating object" sequence. For pose estimation, one marker represents one run of the algorithm (the same number of runs are executed for each frame). For pose tracking, the lines represent means over multiple runs.

5.3 Cluttered scenes

We evaluated the robustness of our method to clutter and occlusions by computing the pose of various objects in several cluttered scenes [3], using a single input image. The estimate of the distance to the objects, used as input, is the same for all scenes and objects, and roughly corresponds to the distance between the camera and the table on which the objects are placed (about 370 mm). Here again, this is an only crude estimate, as the actual distance to the objects varies from 200 to 600 mm.

Several of these scenes are presented in Fig. 5, with object models superimposed in the estimated pose. Sometimes, insufficient observations are extracted from the image, and the pose cannot be recovered (e.g. second row, last image). However, the reprojection error achieved by our algorithm is clearly low in most cases; the models generally appear in close-to-correct poses. A perfect match between the reprojected model and the observations is not always possible, which is a limitation of the sparse observations and object models we use. Small differences in the reprojection on the image plane may then correspond to large errors in the actual 3D pose recovered. Most of these errors can be greatly reduced by using additional views of the scene, which is easily done with our method.

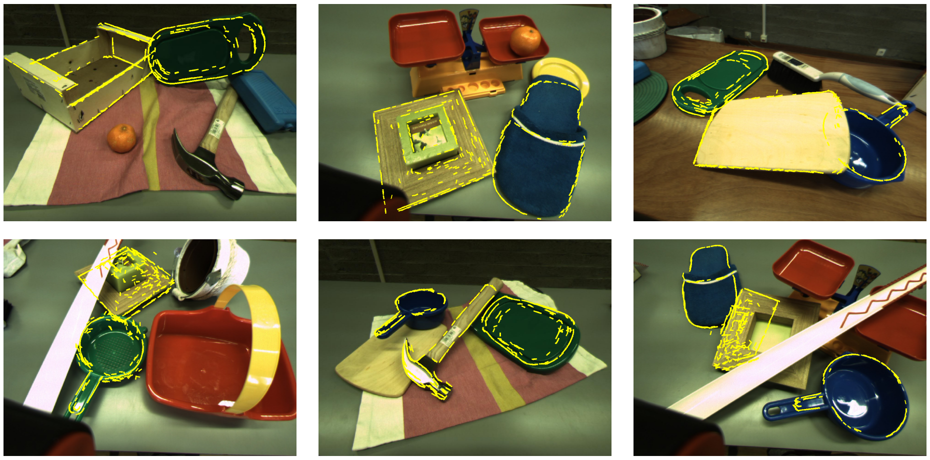


Fig. 5: Results of pose estimation (using a single view), with model features reprojected onto the input image. Most remaining errors are a limitation of the simple object models used, each learned from a single stereo pair.

6 Conclusions

We presented a generic method for 3D pose estimation of objects in 2D images, using a probabilistic scheme for representing object models and observations. This allows the method to handle various types of observations, including features that cannot be matched individually; here we use local edge segments. Using these principles, we showed how to use Metropolis-Hastings MCMC to infer the maximum-likelihood pose of a known object in a novel scene, using a single 2D view of that scene. The probabilistic approach makes the pose estimation process possible without establishing explicit model-to-scene correspondences, as opposed to existing state-of-the-art methods. Together with the use of edge segments as observations, the method allows us to effectively handle non-textured objects. Further, the method extends to the use of multiple views, providing a rigorous framework for integrating evidence from multiple viewpoints of a scene, yielding increased accuracy with only a linear increase of computation time with respect to the number of views. We validated the proposed approach on two publicly-available datasets. One dataset allowed quantitative evaluation; the result of an experiment was compared to the results of an existing method, and showed an advantage in performance for our method. The pose estimation process was also evaluated with success on scenes with clutter and occlusion. Future work will extend the current implementation to the use of other visual features, thereby extending the types of objects that can be handled.

Acknowledgments

The research leading to these results has received funding from the European Community's Seventh Framework Programme FP7/2007-2013 (Specific Programme Cooperation, Theme 3, Information and Communication Technologies) under grant agreement no. 270273, Xperience. Damien Teney is supported by a research fellowship of the Belgian National Fund for Scientific Research.

References

1. Collet, A., Berenson, D., Srinivasa, S., Ferguson, D.: Object recognition and full pose registration from a single image for robotic manipulation. In: ICRA (2009)
2. Collet, A., Srinivasa, S.S.: Efficient multi-view object recognition and full pose estimation. In: ICRA. pp. 2050–2055 (2010)
3. Detry, R.: A probabilistic framework for 3D visual object representation: Experimental data (2009), `http://intelsig.org/publications/Detry-2009-PAMI/`
4. Detry, R., Piater, J.: Continuous surface-point distributions for 3D object pose estimation and recognition. In: ACCV (2010)
5. Detry, R., Pugeault, N., Piater, J.: A probabilistic framework for 3D visual object representation. IEEE Trans. PAMI 31(10), 1790–1803 (2009)
6. Ekvall, S., Hoffmann, F., Kragic, D.: Object recognition and pose estimation for robotic manipulation using color cooccurrence histograms. In: IROS (2003)
7. Gordon, I., Lowe, D.G.: What and where: 3D object recognition with accurate pose. In: Toward Category-Level Object Recognition. pp. 67–82 (2006)
8. Hsiao, E., Collet, A., Hebert, M.: Making specific features less discriminative to improve point-based 3D object recognition. In: CVPR. pp. 2653–2660 (2010)
9. Klein, G., Drummond, T.: Robust visual tracking for non-instrumented augmented reality. In: ISMAR. pp. 113–122. Tokyo (October 2003)
10. Kraft, D., Krüger, N.: Object sequences (2009), `http://www.mip.sdu.dk/covig/sequences.html`
11. Kragic, D., Miller, A.T., Allen, P.K.: Real-time tracking meets online grasp planning. In: ICRA. pp. 2460–2465 (2001)
12. Krüger, N., Wörgötter, F.: Multi-modal primitives as functional models of hypercolumns and their use for contextual integration. In: Gregorio, M.D., Maio, V.D., Frucci, M., Musio, C. (eds.) BVAI. Lecture Notes in Computer Science, vol. 3704, pp. 157–166. Springer (2005)
13. Mittrapiyanuruk, P., DeSouza, G.N., Kak, A.C.: Calculating the 3D pose of rigid objects using active appearance models. In: ICRA. pp. 5147–5152 (2004)
14. Pless, R.: Using many cameras as one. In: CVPR (2). pp. 587–593 (2003)
15. Pope, A.R., Lowe, D.G.: Probabilistic models of appearance for 3D object recognition (2000)
16. Pugeault, N.: Early Cognitive Vision: Feedback Mechanisms for the Disambiguation of Early Visual Representation. Vdm Verlag Dr. Müller (2008)
17. Rothganger, F., Lazebnik, S., Schmid, C., Ponce, J.: 3D object modeling and recognition using local affine-invariant image descriptors and multi-view spatial constraints. Int. J. Comput. Vision 66(3), 231–259 (2006)
18. Sudderth, E.B.: Graphical models for visual object recognition and tracking. Ph.D. thesis, Massachusetts Institute of Technology, Cambridge, MA, USA (2006)
19. Vacchetti, L., Lepetit, V., Fua, P.: Stable real-time 3D tracking using online and offline information. IEEE Trans. PAMI 26(10), 1385–1391 (2004)