Hierarchical Integration of Local 3D Features for Probabilistic Pose Recovery

Renaud Detry
Montefiore Institute, University of Liège, Belgium
Email: Renaud.Detry@ULg.ac.be

Justus Piater
Montefiore Institute, University of Liège, Belgium
Email: Justus.Piater@ULg.ac.be

***Abstract*— This paper presents a 3D object representation framework. We develop a hierarchical model based on probabilistic correspondences and probabilistic relations between 3D visual features. Features at the bottom of the hierarchy are bound to local observations. Pairs of features that present strong geometric correlation are iteratively grouped into higher-level meta-features that encode probabilistic relative spatial relationships between their children. The model is instantiated by propagating evidence up and down the hierarchy using a Belief Propagation algorithm, which infers the pose of high-level features from local evidence and reinforces local evidence from globally consistent knowledge. We demonstrate how to use our framework to estimate the pose of a known object in an unknown scene, and provide a quantitative performance evaluation on synthetic data.**

I. Introduction

Objects can be characterized by configurations of parts. This insight is reflected in computer vision by the increasing popularity of representations that combine local appearance with spatial relationships [1, 2, 12]. Such methods are richer and more easily constructed than purely geometric models, more expressive than methods purely based on local appearance such as bag-of-features methods [10, 3] and more robust and more easily handled in the presence of clutter and occlusions than methods based on global appearance. Moreover, they not only allow bottom-up inference of object parameters based on features detected in images, but also top-down inference of image-space appearance based on object parameters.

We have recently presented a framework for unsupervised learning of hierarchical representations that combine local appearance and probabilistic spatial relationships [13, 14]. By analyzing a set of training images, our method creates a codebook of features and observes recurring spatial relationships between them. Pairs of features that are often observed in particular mutual configurations are combined into a meta-feature. This procedure is iterated, leading to a hierarchical representation in the form of a graphical model with primitive, local features at the bottom, and increasingly expressive meta-features at higher levels. Depending on the training data, this leads to rich representations useful for tasks such as object detection and recognition from 2D images.

We are currently developing an extension of this method to 3D, *multi-modal* features. We intend to integrate multiple perceptual aspects of an object in one coherent model, by combining visual descriptors with haptic and proprioceptive information. This will be directly applicable to robotic tasks such as grasping and object manipulation. Correlated percepts of different natures will induce cross-modal associations; a grasp strategy may be linked directly to visual features that predict its applicability.

In this paper, we focus on hierarchical models for visual object representation. Here, an *observation* is an oriented patch in 3–space, annotated by various visual appearance characteristics. To infer the presence of an object in a scene, evidence from local features is integrated through bottom-up inference within the hierarchical model. Intuitively, each feature probabilistically votes for all possible object configurations consistent with its pose. During inference, a consensus emerges among the available evidence, leading to one or more consistent scene interpretations. The system never commits to specific feature correspondences, and is robust to substantial clutter and occlusions.

We illustrate our method on the application of object pose estimation. Object models are learned within a given world reference frame, within which the object is placed in a reference pose. Comparing an instance of the model in an unknown scene with an instance in the learned scene allows us to deduce the object pose parameters in the unknown scene.

II. Hierarchical Model

Our object model consists of a set of generic *features* organized in a hierarchy. Features that form the bottom level of the hierarchy, referred to as *primitive features*, are bound to visual observations. The rest of the features are *meta-features* which embody spatial configurations of more elementary features, either meta or primitive. Thus, a meta-feature incarnates the relative configuration of two features from a lower level of the hierarchy.

A feature can intuitively be associated to a "part" of an object, i.e. a generic component instantiated once or several times during a "mental reconstruction" of the object. At the bottom of the hierarchy, primitive features correspond to local parts that each may have many *instances* in the object. Climbing up the hierarchy, meta-features correspond to increasingly complex parts defined in terms of constellations of lower parts. Eventually, parts become complex enough to satisfactorily represent the whole object. Figure 1 shows a didactic example of a hierarchy for a bike. The bike is the composition of *frame* and *wheel* features. A wheel is composed of pieces of tire and spokes. The generic piece of tire at the

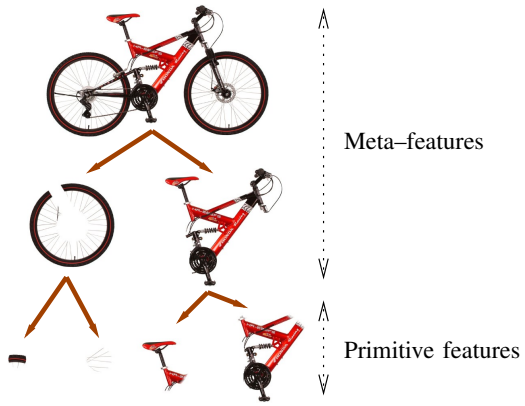


Fig. 1. A didactic example of a hierarchical model of a bike.

Fig. 2. Instances of the generic piece-of-tire primitive feature in the bike scene.

bottom of the hierarchy is a primitive feature; the pieces of tire squared in green in the scene (Figure 2) are instances of that primitive feature.

At the bottom of the hierarchy, primitive features are tagged with an appearance descriptor called a *codebook vector*. The set of all codebook vectors forms a *codebook* that binds the object model to the feature observations, by associating observations to primitive features.

In summary, information about an object is stored within the model in the three following forms:

i. the topology of the hierarchy,
ii. the relationships between related features,
iii. the codebook vectors annotating bottom-level features.

A. Parametrization

Formally, the hierarchy is implemented using a Pairwise Markov Random Field (see Figure 3). Features are associated to hidden nodes (white in Figure 3), and the structure of the hierarchy is reflected by the edge pattern between them. Each meta-feature is thus linked to its two child features. Observed variables y_i of the random field stand for observations.

When a model is associated to a particular scene (during construction or instantiation), features are associated to corresponding instances in that scene. The correspondence between a feature i and its instances is represented by a probability density over the pose space $SE(3) = \mathbb{R}^3 \times SO(3)$ represented by a random variable x_i.

As noted above, a meta-feature encodes the relationship between its two children. However, the graph records this information in a slightly different but equivalent way: instead of recording the relationship between the two child features,

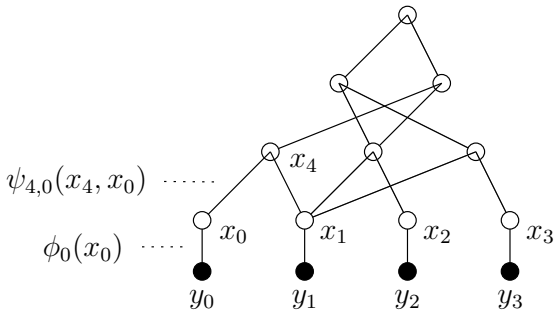


Fig. 3. A Pairwise Markov Random Field representing a feature hierarchy. Features correspond to hidden variables (white). Observed variables (black) correspond to observations, bound to bottom-level primitive features.

the graph records the two relationships between the meta-feature and each of its children. The relationship between a meta-feature i and one of its children j is parametrized by a *compatibility potential function* $\psi_{ij}(x_i, x_j)$ associated to the edge e_{ij}. A compatibility potential specifies, for any given pair of poses of the features it links, the probability of finding that particular configuration for these two features. We only consider rigid-body relationships. Moreover, relationships are *relative* spatial configurations. Compatibility potentials can thus be represented by a probability density over the feature–to–feature transformation space $SE(3)$.

Compatibility potentials allow relationship distributions to have multiple modes. In the bike model, let us consider the meta-feature that represents a generic wheel. There are two wheels in the picture; two instances of the wheel feature will be used in a mental reconstruction of the bike. Hence, the compatibility potential between the *wheel* feature and the *bike* feature will be dense around two modes, one corresponding to the transformation between the bike and the front wheel ("the front wheel is on the right side of the bike"), the other between the bike and the rear wheel ("the rear wheel is on the left side of the bike").

Finally, the statistical dependency between a hidden variable x_i and its observed variable y_i is parametrized by an *observation potential* $\phi_i(x_i)$, also referred to as *evidence* for x_i, which corresponds to the spatial distribution of x_i's observations.

The term *primitive feature instance* formally refers to a random draw from a primitive feature distribution. While a primitive feature instance often corresponds to an observation, observations enter into the graphical model merely as prior knowledge. Primitive feature instances result from inference; they depend on observations *and* on all features of the hierarchy. Owing to inference mechanisms presented in the next paragraph, if an observation is discarded (e.g. occluded), a primitive feature instance may nevertheless appear at its place.

B. Model Instantiation

Model instantiation is the process of detecting instances of an object model in a scene. It provides pose densities for all features of the model, indicating where the learned object is likely to be present. Instantiating a model in a

scene amounts to inferring posterior marginal densities for all features of the hierarchy. Thus, once priors (observation potentials, evidence) have been defined, instantiation can be achieved by any applicable inference algorithms. We currently use a Belief Propagation algorithm described in Section III-A.

For primitive features, evidence is estimated from feature observations. Observations are classified according to the primitive feature codebook; for each primitive feature i, its observation potential $\phi_i(x_i)$ is estimated from observations that are associated to the i^{th} codebook vector. For meta-features, evidence is uniform.

C. Model Construction

The construction procedure starts by clustering feature observations in the appearance space to build a codebook of observations. The number of classes is a parameter of the system. These classes are then used to initialize the first level of the graph:

1) A primitive feature is created for each class;
2) Each primitive feature is tagged with the codebook vector (cluster center) of its corresponding class.

The spatial probabilistic density of each primitive feature is then computed from the spatial distribution of corresponding observations. We use nonparametric representations (see section III-B); the set of observations bound to each primitive feature can thus be directly used as a density representation.

After primitive features have been computed, the graph is built incrementally, in an iterative manner. The construction algorithm works by extracting feature co-occurrence statistics. Features that tend to occur at non-accidental relative positions are repeatedly grouped into a higher-level meta-feature. At each step, the top level of the graph is searched for strongly correlated pairs of features. The k most strongly correlated pairs are selected to form the k meta-features of the next level. The number of meta-features created at each step is a parameter, which we usually keep equal to the initial number of classes. The search for strong feature combinations is the operation responsible for the *topology* of the graph.

The k new meta-features are then provided with a spatial probability distribution, generated from a combination of the children's densities. The meta-feature is placed in the middle of its children, location- and orientation-wise (thus, the meta-feature distribution will be dense between dense regions of the children's distributions). Finally, spatial relations between each meta-feature and its children are extracted, which defines the compatibility potentials. This is achieved by repeatedly taking a pair of samples, one from the parent distribution and one from a child's distribution. The spatial relationships between a large number of these pairs form the relationship distribution between the parent and that child. While the search for strong combinations was responsible for the topology of the graph, the extraction of spatial relations is responsible for the *parametrization* of the graph through the definition of compatibility potentials associated with edges between adjacent features. This parametrization constitutes the principal outcome of the learning algorithm. Relationship extraction is the last operation of a level-construction iteration.

Incremental construction of the graph can, in principle, continue indefinitely, growing an ever-richer representation of the observed scene. The number of levels is a parameter that is chosen to reach a desired level of abstraction; its effect will be discussed in Section V.

III. Implementation

A. Inference

Graphical models are a convenient substrate of sophisticated *inference algorithms*, i.e. algorithms for efficient computation of statistical quantities. An efficient inference algorithm is essential to the hierarchical model, for it provides the mechanism that will let features communicate and propagate information.

Our inference algorithm of choice is currently the Belief Propagation algorithm (BP) [11, 16, 6]. Belief Propagation is based on incremental updates of marginal probability estimates, referred to as *beliefs*. The belief at feature i is denoted

$$b(x_i) \approx \mathbf{p}(x_i|y) = \int ... \int \mathbf{p}(x_1, ..., x_N|y) \, dx_1 ... dx_{i-1} dx_{i+1} ... dx_N$$

where y stands for the set of observations. During the execution of the algorithm, *messages* are exchanged between neighboring features (hidden nodes). A message that feature i sends to feature j is denoted $m_{ij}(x_j)$, and contains feature i's belief about the state of feature j. In other words, $m_{ij}(x_j)$ is a real positive function proportional to feature i's belief about the plausibility of finding feature j in pose x_j. Messages are exchanged until all beliefs converge, i.e. until all messages that a node receives predict a similar state.

At any time during the execution of the algorithm, the current pose belief (or marginal probability estimate) for feature i is the normalized product of the local evidence and all incoming messages, as

$$b_i(x_i) = \frac{1}{Z} \phi_i(x_i) \prod_{j \in \text{neighbors}(i)} m_{ji}(x_i). \quad (1)$$

where Z is a normalizing constant. To prepare a message for feature j, feature i starts by computing a local "pose belief estimation", as the product of the local evidence and all incoming messages *but* the one that comes from j. This product is then multiplied with the compatibility potential of i and j, and marginalized over x_i. The complete message expression is

$$m_{ij}(x_j) = \int \psi_{ij}(x_i, x_j) \phi_i(x_i) \prod_{k \in \text{neighbors}(i) \setminus j} m_{ki}(x_i) dx_i. \quad (2)$$

As we see, the computation of a message doesn't directly involve the complete local belief (1). In general, the explicit belief for each node is computed only once, after all desirable messages have been exchanged.

When BP is finished, collected evidence has been propagated from primitive features to the top of the hierarchy, permitting inference of marginal pose densities at top-level features. Furthermore, regardless of the propagation scheme (message update order), the iterative aspect of the message passing algorithm ensures that global belief about the object pose – concentrated at the top nodes – has at some point been propagated back down the hierarchy, reinforcing globally consistent evidence and permitting the inference of occluded features. While there is no theoretical proof of BP convergence for loopy graphs, empirical success has been demonstrated in many situations.

B. Nonparametric Representation

We opted for a nonparametric approach to probability density representation. A density is simply represented by a set of particles; the local density of these particles in space is proportional to the actual probabilistic density in that region. Compared to usual parametric approaches that involve a limited number of parametrized kernels, problems like fitting of mixtures or the choice of a number of components can be avoided. Also, no assumption concerning the shape of the density has to be made.

Particles live in the Special Euclidean Space $SE(3)$. The location/translation component is parametrized by a 3–vector. For the orientation/rotation component it was decided to prefer quaternions over rotation matrices, for they provide a well-suited formalism for the manipulation of rotations such as composition or metric definition [9, 7].

For inference, we use a variant of BP, Nonparametric Belief Propagation, which essentially develops an algorithm for BP message update (2) in the particular case of continuous, non-Gaussian potentials [15]. The underlying method is an extension of particle filtering; the representational approach is thus nonparametric and fits our model very well.

IV. Object Pose Estimation

Since features at the top of an object model represent the whole object, they will present relatively concentrated densities that are unimodal if exactly one instance of this object is present in the scene. These densities can be used to estimate the object pose. Let us consider a model for a given object, and a pair of scenes where the object appears. In the first scene, the object is in a reference pose. In the second scene, the pose of the object is unknown. The application our method to estimate the pose of the object in the second scene goes as follows:

1) Instantiate the object model in the reference scene. For every top-level feature i of the instantiated graph, compute a *reference aggregate feature pose* π_1^i from its unimodal density.
   Instantiating the model in a reference scene is necessary because even though the top-level features all represent the whole object, they come from different recursive combinations of features of various poses.
2) Instantiate the object model in the unknown scene. For every top feature of that graph, compute an *aggregate feature pose* π_2^i.
3) For all top level features i, the transformations from π_1^i to π_2^i should be very similar; let us denote the mean transformation t. This transformation corresponds to the rigid body motion between the pose of the object in the first scene and its pose in the second scene. Since the first scene is a reference pose, t is the pose of the object in the second scene.

A prominent aspect of this procedure is its ability to recover an object pose without explicit point-to-point correspondences. The estimated pose emerges from a negotiation involving all available data.

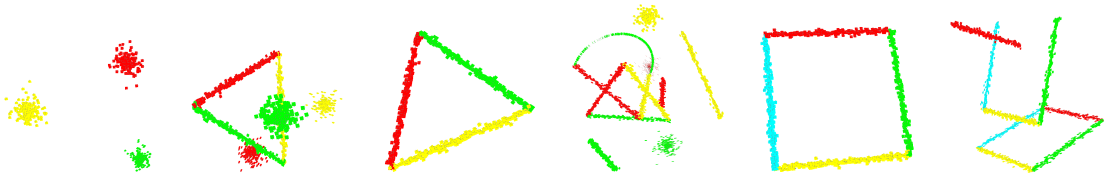
V. Experiments

We ran pose estimation experiments on a series of artificial "objects" presented in Figure 4. In these experiments, we bypass the clustering step and directly generate evidence for primitive features. Since we use nonparametric density representations, we generate observations that directly become evidence for primitive features. Primitive features may have distributions in the shape of blobs, lines, and curves (see Figure 4). For a blob, location components of observations are drawn from a Gaussian distribution around a random 3D point; orientation components are drawn from a Von Mises-Fisher distribution [5, 4] centered at a random 3D orientation. For a line, locations are drawn from a Gaussian distribution around a line segment; orientations are drawn from a Von Mises-Fisher distribution centered at a 3D orientation such that its main direction is along the line and its second direction is in a fixed plane. Figure 5 illustrates orientations.

In the next paragraphs, we go through the procedure of a pose estimation experiment. First, a model is learned from one set of observations of an object of interest (the reference scene). A hierarchy is built up to n levels, we instantiate the model in the reference scene, and compute a reference aggregate feature pose π_1^i for every top feature i of the model.

We are then ready to estimate the pose of our object in a novel, noisy scene. We initialize primitive-feature evidence of the model on a fresh draw of observations of the object of interest in a random pose *plus* observations of a foreign object (see Figures 4(b), 4(d), 4(f)). Evidence is propagated through the hierarchy, and we can eventually estimate the top-feature poses. Since the object of interest is present only once in the noisy scene, top level features should, after instantiation, present unimodal densities; we can safely compute a mean pose π_2^i for each of them.

Finally, we compute the transformation t_i between π_1^i and π_2^i for every top feature i. As noted in Section IV, all t_i are very similar. Let us denote the mean transformation t, which corresponds to the *estimated* rigid body motion between the pose of the object in the reference scene, and its pose in the noisy scene. Let us also denote δt the standard deviation of individual t_i's around t.



(a) **blobs** (object) (b) **blobs** (noisy scene) (c) **triangle** (object) (d) **triangle** (noisy scene) (e) **square** (object) (f) **square** (noisy scene)

Fig. 4. Synthetic object observations in Figures (a), (c), (e); noisy scene for each object in Figures (b), (d), (f). Each figure shows primitive-feature densities; color indicates the different primitive feature classes. For instance, Figure (a) shows a simple object consisting of three blobs. The bottom level of the hierarchy corresponding to this object will be composed of three primitive features. For each blob, all observations are associated to one and the same primitive feature.

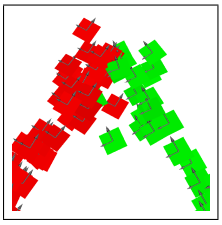
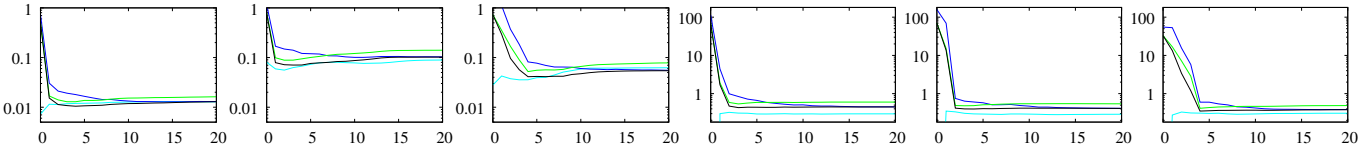


Fig. 5. Artificially generated observations and their poses.



(a) Translation error (blobs) (b) Translation error (triangle) (c) Translation error (square) (d) Rotation error (blobs) (e) Rotation error (triangle) (f) Rotation error (square)

Fig. 6. Error of the translation (relative to object size) and rotation estimates (in degrees) as a function of the number of levels. The cyan line indicates the error when the pose is estimated on a background-noise–free scene, i.e. for an experiment similar to that described in the text, except that we do not add observations from a foreign object before pose estimation. This error is already low at level 0, since the mean of each primitive feature observations for model instantiation is very similar to that used for model learning. The black line indicates the mean error for noisy scenes – i.e. scenes including foreign objects. The green and blue lines indicate the variance across runs and across top-level nodes. See the text for details.

To evaluate the quality of our estimation, we compare t to the *ground truth* rigid body motion T of the object of interest between the reference scene and the noisy scene. Comparison relies on the distances between translations and distances between rotations. The distance between two rotations θ and θ' is defined as the angle (in degrees) of the 3D rotation that moves from θ to θ'. It can be computed using the quaternion representations of θ and θ' as [9]:

$$\mathbf{d}(q, q') = 2 \arccos(|q \cdot q'|).$$

For each object, this experiment is repeated with different hierarchy heights, from 0 to 20, and for different random seeds. Results are presented in Figure 6. Let us denote $(\lambda_t^s, \theta_t^s)$ and $(\lambda_T^s, \theta_T^s)$ the translational and rotational parts of transformations t and T for a random seed s. Figures 6(a), 6(b) and 6(c) show the mean error of translation estimates as a function of the number of levels. They present on a logarithmic scale the mean distance between λ_t^s and λ_T^s for all s divided by the global size of the object. The global size of the object is defined as the standard deviation of its observations from its center of gravity. Figures 6(d), 6(e) and 6(f) show, on a logarithmic scale, the mean error in degrees of rotation estimates as a function of the number of levels.

The mean error is always large for shallow hierarchies, but decreases rapidly for taller hierarchies until it eventually reaches a stable value. For objects of increasing complexity, this happens at increasingly higher levels. In particular, the noisy scene for the square contains the square itself, plus a second shape that corresponds to a square with one displaced edge. It is only after level 4 that the wrong shape is discarded, and a correct pose of the square is successfully estimated. The triangle has to be detected in a very noisy scene. This leads to a larger translational error that does not get smaller than 0.1 – about 5% of the edge length of the triangle.

In Figure 6, green lines give an idea of the variance between runs under different random seeds. They show the mean error plus three standard deviations. This variance is relatively large since the random variations affect both the synthetic scenes and the models constructed. Lines in blue show the mean error plus three times the mean (over individual runs) of inter-feature standard deviations δt; they give an idea of the variance between top-level features of the same graph during a given run. This variance is large for shallow hierarchies, but converges to 0 for higher levels, which means that top-level features of a model tend to agree more and more as we use taller hierarchies.

The accuracy of pose estimation is further illustrated in Figure 7 that shows the noisy triangle scene (green) and the estimated triangle pose (red).

In the above experiments, feature observations are generated synthetically. Thereby, we avoid the problem of extracting 3D features from sets of images. By manually associating observations to primitive features, we have control over the clustering step. Since the features are synthesized in 3D, there are no viewpoint issues. Despite their simplicity, these experiments demonstrate the feasibility of our sophisticated method.

One way to obtain 3D feature observations from real objects is the early-cognitive-vision system MoInS [8], which extracts 3D primitives from stereo views of a scene (see Figure 8).

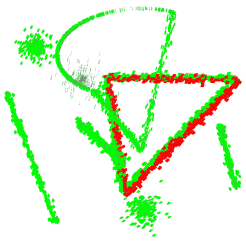
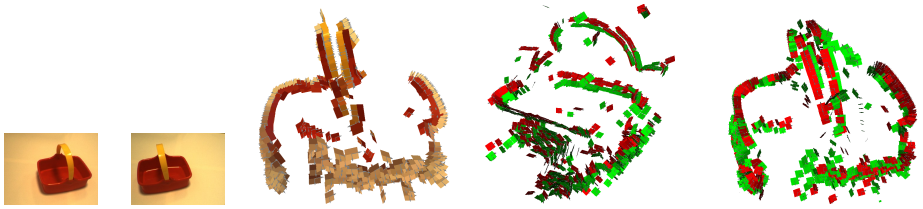


Fig. 7. Accuracy of pose estimation. The noisy triangle scene is green, and the red triangle indicates where the system estimates its position.



(a) Left (b) Right (c) 3D primitives (d) Pose estimation after a 1cm translation. (e) Pose estimation after a 5cm translation.

Fig. 8. Using MoInS 3D primitives as observations. Figures (a) and (b) show a stereo view of a basket. Figure (c) shows MoInS observations. Figures (d) and (e) show the result of two pose estimation experiments; in Figure (e), visualization is rendered from the camera viewpoint whereas in Figure (d) it is rendered from a different viewpoint.

Figures 8(d) and 8(e) show preliminary results with MoInS features. A model for the basket is learned from one stereo pair (see Figure 8(c)). The model is then instantiated in a scene shot 1cm closer to the basket (Figure 8(d)) and in another scene shot 5cm closer to the basket (Figure 8(e)). The 5cm result happens to look better because it is rendered from a viewpoint similar to the stereo camera, and – as is typical for stereo reconstruction – MoInS 3D primitives are localized much more accurately in a direction perpendicular to the optical axis of the camera than in depth.

As noted above, this experiment is preliminary. For technical reasons, we were limited to translational motions along the optical axis. We plan to work on sequences involving rotations and multiple objects in the near future. The system already proved some robustness against clutter in the artificial experiments, and viewpoint-related issues will be eased by the MoInS system.

VI. Conclusion

We presented a probabilistic framework for hierarchical object representation. Hierarchies are implemented with Pairwise Markov Random Fields in which hidden nodes represent generic features, and edges model the abstraction of highly correlated features into a higher-level meta-feature. Once PMRF evidence is extracted from observations, posterior marginal pose densities for all features of the graph are inferred by the Belief Propagation algorithm.

Posterior pose densities can be used to compute a pose for a known object in an unknown scene, which we demonstrated through a series of experiments to estimate rigid body motion. We are thus able to achieve pose recovery without prior object models, and without explicit point correspondences.

Our framework is not specific to visual features and allows the natural integration of non-visual features such as haptic and proprioceptive parameters. This will potentially lead to cross-modal representations useful for robotic grasping and exploratory learning of object manipulation, which we will explore in future work.

Acknowledgment

This work was supported by the Belgian National Fund for Scientific Research (FNRS) and the EU Cognitive Systems project PACO-PLUS (IST-FP6-IP-027657).

The authors thank Norbert Krüger and Nicolas Pugeault for providing MoInS data and for fruitful discussions.

References

[1] M. Burl, T. Leung, and P. Perona. Face localization via shape statistics. In *Intl. W. on Automatic Face and Gesture Recognition*, 1995.

[2] Michael C. Burl, Markus Weber, and Pietro Perona. A probabilistic approach to object recognition using local photometry and global geometry. In *ECCV '98: Proceedings of the 5th European Conference on Computer Vision-Volume II*, pages 628–641, London, UK, 1998. Springer-Verlag.

[3] Chris Dance, Jutta Willamowski, Lixin Fan, Cedric Bray, and Gabriela Csurka. Visual categorization with bags of keypoints. In *ECCV International Workshop on Statistical Learning in Computer Vision*, 2004.

[4] I. Dhillon and S. Sra. Modeling data using directional distributions. Technical report, University of Texas, Austin, 2003.

[5] RA Fisher. Dispersion on a sphere. In *Proc. Roy. Soc. London Ser. A.*, 1953.

[6] Michael I. Jordan and Yair Weiss. Graphical models: Probabilistic inference. In M. Arbib, editor, *The Handbook of Brain Theory and Neural Networks, 2nd edition*. MIT Press, 2002.

[7] Charles F. F. Karney. Quaternions in molecular modeling. *J. Mol. Graph. Mod.*, 25, 2006.

[8] Norbert Krüger and Florentin Wörgötter. Multi-modal primitives as functional models of hyper-columns and their use for contextual integration. In Massimo De Gregorio, Vito Di Maio, Maria Frucci, and Carlo Musio, editors, *BVAI*, volume 3704 of *Lecture Notes in Computer Science*, pages 157–166. Springer, 2005.

[9] James Kuffner. Effective sampling and distance metrics for 3d rigid body path planning. In *Proc. 2004 IEEE Int'l Conf. on Robotics and Automation (ICRA 2004)*. IEEE, May 2004.

[10] Thomas K. Leung and Jitendra Malik. Representing and recognizing the visual appearance of materials using three-dimensional textons. *International Journal of Computer Vision*, 43(1):29–44, 2001.

[11] Judea Pearl. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. Morgan Kaufmann, 1988.

[12] Justus H. Piater and Roderic A. Grupen. Toward learning visual discrimination strategies. In *CVPR*, pages 1410–1415. IEEE Computer Society, 1999.

[13] Fabien Scalzo and Justus H. Piater. Statistical learning of visual feature hierarchies. In *CVPR '05: Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05) - Workshops*, page 44, Washington, DC, USA, 2005. IEEE Computer Society.

[14] Fabien Scalzo and Justus H. Piater. Unsupervised learning of dense hierarchical appearance representations. In *ICPR '06: Proceedings of the 18th International Conference on Pattern Recognition (ICPR'06)*, volume 2, pages 395–398, Washington, DC, USA, August 2006. IEEE Computer Society.

[15] Erik B. Sudderth, Alexander T. Ihler, William T. Freeman, and Alan S. Willsky. Nonparametric belief propagation. *cvpr*, 01:605, 2003.

[16] Jonathan S. Yedidia, William T. Freeman, and Yair Weiss. Understanding belief propagation and its generalizations. Technical report, Mitsubishi Electric Research Laboratories, 2002.