Recursive Tower of Knowledge for Learning to Interpret Scenes

Mai Xu and Maria Petrou
Electrical and Electronic Engineering Department
Imperial College London
Exhibition Road, London, SW7 2AZ, UK
{Mai.Xu06, Maria.Petrou}@imperial.ac.uk

Abstract

The Tower of Knowledge architecture integrates probability theory and logic for making decisions. The scheme models the causal dependencies between the functionalities of objects and their descriptions, and then employs the maximum expected utility principle, which combines probability theory and logic, to select the most appropriate label for the object. Since most existing scene interpretation methods rely heavily on training data, we develop in this paper a recursive version of ToK to avoid such dependency. Recursive ToK learns the prior distributions iteratively from the decisions of labelling components made in the last iteration, partly by functionalities of components, and partly by the already learnt prior distributions in previous iterations. To validate our method in the domain of 3D outdoor scene interpretation, we compare ToK against a state-of-the-art method, Expandable Bayesian Networks (EBN), for labelling components of buildings. Experimental results then show that the labelling accuracy of ToK is superior to that of EBN. Also, these results reveal that recursive ToK improves the accuracy of ToK for labelling 3D components in the worst case when lacking any training data.

1 Introduction

For several years, probabilistic and logic based approaches were used in dichotomy. Recently, it has been recognised that a combination of these approaches may prove very useful in computer vision [19]. It is also emerging that statistical (and by extension probabilistic) reasoning on objects may best be inferred via semantic relationships between the objects, and that dynamic scenes with observed relations and actions in temporal sequences may help in cognitive tasks [4, 15].

Scene interpretation is a fundamental problem in computer vision with aim to recognise objects by relating a set of primitives to a collection of labels or semantic representations. Based on probabilistic graphical models, a system called Description Logic (DL) was proposed for scene interpretation by Neumann et al [14]. Another algorithm is the informative local features approach based on decision-trees [6]. Hudelot et al [10] introduced a support vector machine as another method for learning in scene interpretation. A large body of research focuses on neural networks or advanced neural networks for scene interpretation [17]. A representative work using neural networks is evolutionary

BMVC 2008 doi:10.5244/C.22.63

optimization (ENN) [18]. Recently, a growing trend of scene interpretation has focused on some graphical probabilistic models such as Bayesian networks algorithms [2, 3, 12] and Markov random fields (MRF) [9, 13]. Kim and Nevatia [12] investigated expandable Bayesian networks (EBN) as a method of interpreting 3D objects. EBN is introduced as a reasoning tool of interpretation to solve the problem that evaluation of hypotheses based on evidence is uncertain because the number of images being used is not fixed and some modalities may not be always available. However, the accuracy of all the above methods relies heavily on the availability of enough training examples to populate adequately the feature space. The main motivation of our paper is therefore in solving this problem.

Most recently, an architecture, namely the Tower of Knowledge (ToK), that combines logic and probabilities was proposed for scene interpretation [15, 16]. In this paper, we extend this scheme into an iterative form to improve its performance and enforce its self-learning capabilities.

2 Brief Overview of the ToK Scheme

Figure 1 shows schematically the ToK architecture. This architecture is designed to label a scene on the basis of answering the question "what" through answering the questions "why" and "how". In trying to label a component being a balcony, the following sequence of logic processing may be envisaged:
"What is this?"– "It is a balcony."
"Why?"– "Because it is attached to a building and people can stand in it."
"How?"– "By offering enough space for a person to stand in and by being attached on a wall with an opening area to allow people to enter it from the building. "
"Is it really like that? Let me check."
Given the above reasoning sequence, the tower of knowledge consists of four levels: image level, semantic level, functionality level and description level. The image level belongs to low-level vision. Features extracted from images are the units of this level and the input to the tower of knowledge. Image processing modules working at this level process the components that are input to the next level for labelling. The other three levels belong to the high level vision. The nouns of the semantic level are the names of the objects, i.e. labels (e.g. "balcony","window"). The remaining two levels are those of the functionalities and the descriptors, which may be seen as the implicit logic representation of object models. The verbs of the functionality level are functionalities of the objects such as "to stand in" and "to look out". A functionality may be fulfilled, if the object has certain characteristics. These are captured at the description level. Examples of these units are "having enough space for a person" and "there is an opening on the wall". The units in the description level can interrogate the sensors and the images to verify that a required descriptor applies to an object. This way, the vertical connections of the scheme encode implicitly the generic models of objects, seen not in isolation or as static entities, but in the context of what they are used for and what their generic characteristics are. These generic models effectively encode the logic rules of meta-knowledge that have been learnt from spatio-temporal activities involving the objects we wish to label. According to this scheme, an object is assigned a label as follows.

Let us assume that we use maximum a posteriori (MAP) estimation to assign labels to a scene. In the conventional way of doing so, object a_i ($\in \{a_1, a_2, \ldots, a_n\}$) will be

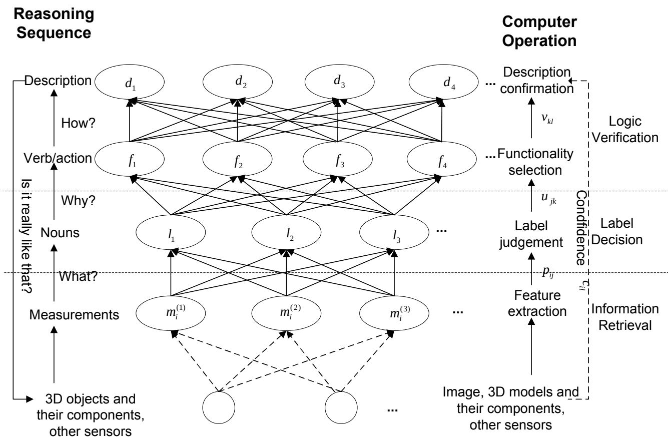


Figure 1: The tower of knowledge for labelling object a_i. The units in this figure stand for the measurements $\mathbf{M_i}$, labels $\mathbf{L}$, functionalities $\mathbf{F}$ and descriptors $\mathbf{D}$.

assigned label l_j ($\in \mathbf{L} = \{l_1, l_2, \ldots, l_M\}$) with probability p_{ij}, given by:

$$p_{ij} = p(l_j|\mathbf{M_i})p(\mathbf{M_i}) = p(\mathbf{M_i}|l_j)p(l_j) = \prod_{t=1}^{s} p(m_i^{(t)}|l_j)p(l_j) \tag{1}$$

where $\mathbf{M_i}$ ($\mathbf{M_i} = \{m_i^{(1)}, m_i^{(2)}, \ldots, m_i^{(s)}\}$) represents all the measurements we have made on object a_i, and $p(\mathbf{M_i})$ and $p(l_j)$ are the prior probability mass functions (pmfs) of measurements and labels, respectively. Let us identify the units in the functionality level of Figure 1 by f_k ($\in \{f_1, f_2, \ldots, f_p\}$), and the units at the description level of Figure 1 by d_l ($\in \{d_1, d_2, \ldots, d_q\}$). Here, we use utility of utility theory to represent the logic consequences inferred by the functionalities of objects and their descriptions. We may choose label l_{j_i} for object a_i according to the *maximum expected utility principle* as follows:

$$l_{j_i} = \underset{l_j \in \mathbf{L}}{\operatorname{argmax}} \sum_{k=1}^{p} u_{jk} \sum_{l=1}^{q} v_{kl} c_{il} p_{ij} \tag{2}$$

where u_{jk} indicates how important is for an object with label l_j to fulfil functionality f_k; v_{kl} indicates how important characteristic d_l is for an object to be able to fulfil functionality f_k, and c_{il} is the confidence we have that descriptor d_l applies to object a_i.

3 A recursive version of ToK

One practical difficulty in applying Equation (2) is that it requires the initial knowledge of $p(m_i^{(t)}|l_j)$ and $p(l_i)$. If there are not enough training data, we may apply a recursive version of ToK to deal with the unavailability of enough training data. First of all, let us assume that the tth observed values of each component $m^{(t)}$ of the measurements vector $\mathbf{M}$ are represented by a histogram $h^{(t)}$, with a finite number of fixed width bins. Recursive ToK then provides a simple yet effective way to sequentially update $p(m^{(t)}|l_j)$ and $p(l_j)$, leading to optimal decisions. Based on Equation (2), at each step r of the recursive ToK, the label can be assigned as,

$$l_{j_i}^{(r)} = \arg\max_{l_j \in \mathbf{L}} \sum_{k=1}^{p} u_{jk} \sum_{l=1}^{q} v_{kl} c_{il} \prod_{t=1}^{s} p^{(r)}(m_i^{(t)}|l_j) p^{(r)}(l_j) \tag{3}$$

Now consider the way of computing $p(m^{(t)}|l_j)$ and $p(l_j)$ for each step r, denoted by $p^{(r)}(m^{(t)}|l_j)$ and $p^{(r)}(l_j)$. Here, we consider the worst case of lacking training examples, which means that there are no training data in the database. Therefore, at the initial step, we assume that every label is equally probable a priori ($p^{(0)}(l_j) = p^{(0)}(l_k)$) for all l_i and l_k in $\mathbf{L}$). Similarly, for all l_i and l_k in $\mathbf{L}$, we have equal conditional probabilities $p^{(0)}(m^{(t)}|l_j)$.

A term called "innovation" is introduced related to a new global distribution of each label in the scene at each step. It is common [8] to represent the innovation of label l_j for each step r as a simple function:

$$In(l_j, r) = (1-\lambda) \frac{\sum_{i=1}^{n} \delta(l_{j_i}^{(r-1)} = l_j)}{n} \tag{4}$$

where $\delta(l_{j_i}^{(r-1)} = l_j)$ will be 1 if $l_{j_i}^{(r-1)} = l_j$, and else it will be 0. Parameter n is the total number of components, and λ ($\in [0,1]$) is a memory factor, which is used to control the impact of pmfs of the previous steps $(1, 2, \ldots, r-1)$ to the current step r. Also, the innovation of value $m^{(t)}$ conditioned on l_j can be represented as,

$$In(m^{(t)}|l_j, r) = (1-\lambda) \frac{\sum_{k=1}^{n} \delta(m^{(t)}, l_{j_k}^{(r-1)} = l_j)}{\sum_{k=1}^{n} \delta(l_{j_k}^{(r-1)} = l_j)} \tag{5}$$

where $\delta(m^{(t)}, l_{j_k}^{(r-1)} = l_j)$ will be 1 if $l_{j_k}^{(r-1)} = l_j$ for the specified value of the t-th measurement, and else it will be 0.

At each step r, the innovations can be used in the following equations,

$$p^{(r)}(l_j) = \lambda p^{(r-1)}(l_j) + (1-\lambda) \frac{\sum_{i=1}^{n} \delta(l_{j_i}^{(r-1)} = l_j)}{n} \tag{6}$$

$$p^{(r)}(m^{(t)}|l_j) = \lambda p^{(r-1)}(m^{(t)}|l_j) + (1-\lambda) \frac{\sum_{k=1}^{n} \delta(m^{(t)}, l_{j_k}^{(r-1)} = l_j)}{\sum_{k=1}^{n} \delta(l_{j_k}^{(r-1)} = l_j)} \tag{7}$$

In Equations (6) and (7), the first term on the right-hand side is used to avoid the sudden change of value and thus convergence to a wrong solution. The second term, innovation, adapts the new knowledge learnt in the previous steps to this specific example (scene).

Finally, we have $l_{j_i}^{(R)}$ after R iterations as the output of the recursive ToK. Such a recursive version of the ToK learns prior knowledge adaptively from the meta-knowledge it already has, and can endure the situation of lacking training data. The overall algorithm is summarised in Table 1.

4 Application to labelling 3D models of buildings

In order to verify the effectiveness and robustness of ToK and its recursive version for scene interpretation, we exemplify our ideas in the context of labelling the components

Table 1: Summary of the recursive tower of knowledge

```
1.  (a) Create the histograms of $m^{(t)}$, one per possible label.
    (b) Initialize the pmfs $p^{(0)}(l_j)$ and $p^{(0)}(m^{(t)}|l_j)$ as uniform pmfs, $j = 0, 1, \dots, M$.
2. Repeat for $r = 1, 2, \dots, R$
    (a) Repeat for $i = 1, 2, \dots, n$
        i. Assign each component $a_i$ label $l_{j_i}^{(r-1)}$ by using Equation (3).
    (b) Repeat for $j = 1, 2, \dots, M$
        i. Update the prior pmfs of each component by using Equation (6).
        ii. For $t = 1, 2, \dots, s$ and for each measurement value $m^{(t)}$ of all components,
            update the prior conditional pmfs using Equation (7).
3. By using Equation (3), assign to component $a_i$ label $l_{j_i}^{(R)}$ as the output of recursive
   ToK, $i \in \{1, 2, \dots, n\}$.
```

of 3D models of buildings. The prior probabilities for each type of component and the conditional probabilities of (2) ($p(l_j)$ and $p(m_i^{(t)}|l_j)$) have been learnt using the e-TRIMS database [11] and the histograms of the distributions of measurements from manually annotated training images. We dicuss next how we define the values of u_{jk}, v_{kl} and c_{il}.

4.1 Meta-knowledge of 3D scene interpretation

The specific meaning of the meta-knowledge codes used in Section 2 is given in Figure 2. Assumimg that all effects from a certain cause are equally likely, we may express u_{jk} and v_{kl} as

$$u_{jk} = \frac{\delta(l_j \to f_k)}{\sum_{m=1}^{p} \delta(l_j \to f_m)} \tag{8}$$

$$v_{kl} = \frac{\delta(f_k \to d_l)}{\sum_{n=1}^{q} \delta(f_k \to d_n)} \tag{9}$$

where $\delta(l_j \to f_k)$ takes value 1 if label l_j implies functionality f_k; else it takes value 0. Similarly, $\delta(f_k \to d_l)$ takes value 1 if functionality f_k can be fulfilled by description d_l; else it takes value 0. Arranging the values of u_{jk} and v_{kl} in the form of matrices, we may write:

$$\mathbf{U} = \begin{bmatrix} 0 & 1/2 & 0 & 1/2 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \quad \mathbf{V} = \begin{bmatrix} 1/2 & 0 & 1/2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1/2 & 1/2 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 1/2 & 1/2 & 0 & 0 & 0 & 0 \end{bmatrix}$$

Finally, we must have a method that will allow the ToK to work out the values of c_{il} of equation (2) by interrogating the measurements made on the scene. In particular, c_{il} can be calculated using the method described in [16].

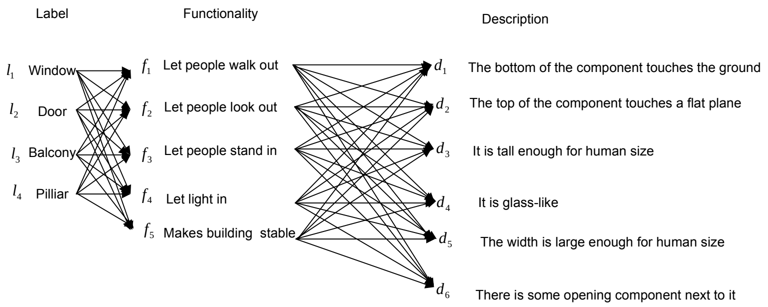

Label
Functionality
Description
l_1 Window
l_2 Door
l_3 Balcony
l_4 Pilliar
f_1 Let people walk out
f_2 Let people look out
f_3 Let people stand in
f_4 Let light in
f_5 Makes building stable
d_1 The bottom of the component touches the ground
d_2 The top of the component touches a flat plane
d_3 It is tall enough for human size
d_4 It is glass-like
d_5 The width is large enough for human size
d_6 There is some opening component next to it

Figure 2: The detailed meta-knowledge of the logic verification sub-system in the ToK. The single-headed arrow lines display the relationship from effect to cause.

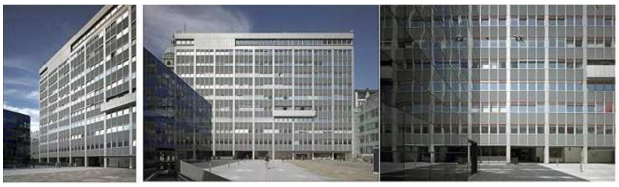


Figure 3: Some original images of a building. There are 5 different view images for reconstructing this building.

4.2 Experimental Results

To validate our method, this section describes our experimental results for labelling components of 3D models with the ToK, its recursive version and another state-of-the-art version of scene interpretation algorithm, namely expandable Bayesian networks [12]. In [12], Kim and Nevatia applied EBN to recognise roofs, walls etc. leading to the detection of buildings, but we apply here EBN to recognise building components such as doors, windows etc. Learning of $p(m_i^{(t)}|l_j)$ and $p(l_j)$ is done by using the eTRIMS database [11] (which consists of more than 200 buildings with over 5000 manually identified components).

Problems of 3D reconstruction do not concern us here since the ToK scheme is independent of the way the 3D model is reconstructed. We thus assume that the 3D model of the building has already been reconstructed by epipolar geometry estimation [7] to obtain 3D points of the building. Plane estimation [5] was used to establish the walls of the building from the 3D points. In addition, we segmented the 3D components of the objects manually. Here, 500 components of 5 reconstructed buildings (3 types: modern, classical and traditional) are tested, yet the labelling results of only one building will be discussed in detail, and the overall results for all buildings will be presented in general.

Figure 3 shows three selected images of a building we wish to reconstruct and label. Its 3D model is displayed in Figure 4.

We first tested the ToK and EBN methods for labelling components of the building in the case that training data are available. We obtain the results of labelling its components by EBN and ToK as shown in Figures 5. Notice that both EBN and ToK made mistakes of labelling two windows in the ground floor as doors because even human beings may make such a mistake without previously knowing that they can not be opened. Also, notice

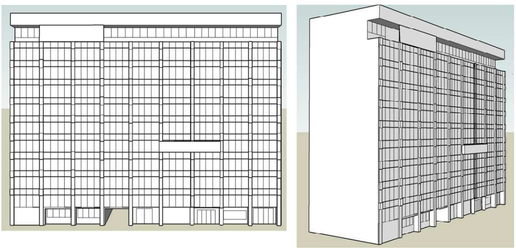
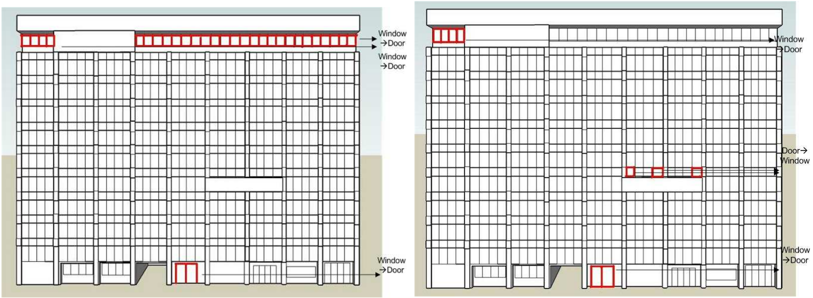


Figure 4: The 3D model of Figure 3



(a)

(b)

Figure 5: The experimental results of labelling the components of the 3D model of Figure 4 using EBN (Figure (a)) and ToK (Figure (b)). The red pane is the component with the incorrect label. Window → door means that the window is mislabelled as door.

that, the ToK scheme mislabelled three doors above the balcony, yet EBN lebelled them correctly. This is mainly due to the glass-like appearance of these doors, also necessary for the functionalities of a window to let light in and let people look through it.

Then, in order to demonstrate the robustness of recursive ToK in the case of lacking training data, we used both ToK and recursive ToK for labelling components in the worst case of totally lacking any training data, which leads to $p(m_i^{(t)}|l_j)$ and $p(l_j)$ of equation (2) to be uniform distributions. Here, iteration number R of recursive ToK was set to 50. Figure 6 characterises the performance of recursive ToK for 3D scene interpretation along with the increased number of iterations when no training examples are provided. Note that, in this figure, the accuracy of the first iteration corresponds to the labelling results of the original ToK without any training data. This figure reveals the convergence of the accuracy to a better percentage after a few iterations and the speed of such convergence with respect to memory factor λ when the recursive version of ToK is applied.

Figures 7 and 8 show four more buildings we used in our experiments for evaluation. The accuracy of labelling the components of all buildings is reported in Table 2, and in more detail in Table 3. The memory factor λ and the iteration number in all experiments were set to 0.9 and 50, respectively. Notice that without any training data, EBN can not work at all since the prior probabilities are unknown and its results are thereby wholly random, so we do not present the results of EBN without any training data. For all these buildings, the results then clearly demonstrate that ToK works better than EBN, and that recursive ToK can greatly improve the accuracy of ToK for 3D scene interpretation while the prior training data are not available.

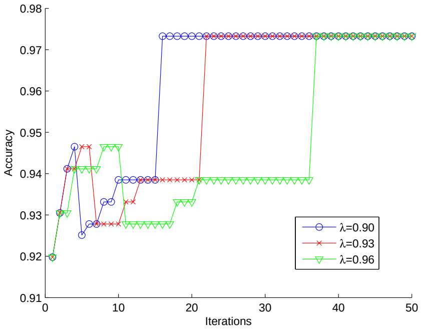
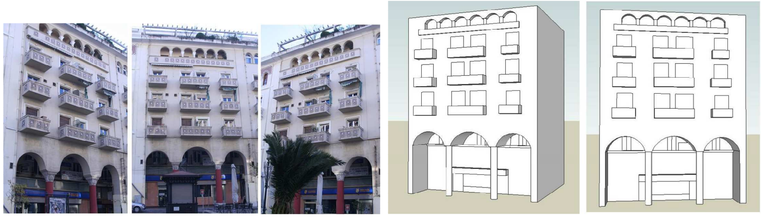


Figure 6: Performance of recursive ToK, as a function of iterations.



(a) (b)

Figure 7: (a) The selected original three-view images of a building chosen from the eTRIMS database [11]. There are 9 different view images for reconstructing the building. (b) Its 3D reconstructed model.

	EBN with training data	ToK with training data	ToK without training data	Recursive ToK without training data
Building 1	92%	97.6%	91.2%	97.3%
Building 2	91.4%	100%	77.1%	91.4%
Building 3	96.8%	96.8%	87.1%	96.8%
Building 4	100%	100%	95.0%	95.0%
Building 5	100%	100%	100%	100%
All buildings	93.4%	98.0%	90.8%	97.0%

Table 2: Summary of percentage accuracy of the EBN algorithm with training data, the ToK scheme with and without training data, and recursive ToK scheme without training data for 3D scene interpretation. There are in total 500 components of those 5 buildings for being labelled.

	Window(Results)	Door(Results)	Balcony(Results)	Pillar(Results)
Window (Ground Truth)	**375-380-398-394**	30-22-6-10	0-1-1-1	0-2-0-0
Door (Ground Truth)	0-21-3-4	**42-24-42-41**	0-0-0-0	3-0-0-0
Balcony (Ground Truth)	0-0-0-0	0-0-0-0	**19-19-19-19**	0-0-0-0
Pillar (Ground Truth)	0-0-0-0	0-0-0-0	0-0-0-0	**31-31-31-31**

Table 3: The overall results of 3D scene interpretation for all buildings using EBN - ToK without prior training data - ToK - Recursive ToK without prior training data.

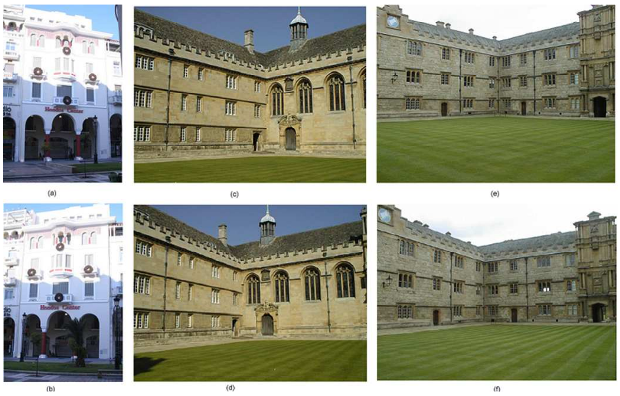


Figure 8: The selected two-view images of another 3 buildings chosen from the eTRIMS database [11] and the IMPACT database [1].

5 Conclusions

In this paper, we proposed, implemented and tested a recursive version of ToK for scene interpretation. This recursive ToK theoretically draws on the ideas of ToK and adaptive algorithms by using the results of each iteration, decided by meta-knowledge (the logic inserted into the computer by humans) and training results of previous iterations, to learn and update the distributions of measurement values for the various classes and the prior probability of the various labels for the next iteration. One significant advantage of recursive ToK over most existing methods, such as EBN, is that it does not lie heavily on the availability of enough training data to populate the whole feature space. It is also attractive for its self-learning capability by iteratively integrating probabilistic theory and logic. Experimental results on several 3D building models show the superiority of recursive ToK for solving the problem of lacking training data for 3D scene interpretation.

Acknowledgements: This work was supported by the EU project 027113, eTRIMS.

References

[1] The image processing for automatic cartographic tools project. http://www.robots.ox.ac.uk/ impact.

[2] T. O. Binford and T. S. Levitt. Evidential reasoning for object recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(7):837–851, 2003.

[3] L. Cheng, T. Caelli, and A. Sanchez-Azofeifa. Component optimization for image understanding: a bayesian approach. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(5):684–693, 2006.

[4] D. Damen. Constraint-based scene interpretation. In *Dagstuhl Logic and Probability for Scene Interpretation Workshop*, page http://kathrin.dagstuhl.de/08091/Materials2/, 2008.

[5] A. R. Dick, P. H. S. Torr, and R. Cipolla. Modelling and interpretation of architecture from several images. *Journal of Computer Vision*, 60(2):111–134, 2004.

[6] G. Fritz, C. Seifert, and L.Paletta. Urban object recognition from informative local features. In *Proceedings of the 2005 IEEE International Conference on Robotics and Automation*, pages 131–137, 2005.

[7] R. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, Cambridge, 1st edition, 2003.

[8] S. Haykin. *Adaptive Filter Theory*. Princeton University Press, 1st edition, 1986.

[9] D. Heesch and M. Petrou. Non-Gibbsian Markov random fields for object recognition. In *Proceedings of BMVC*, 2007.

[10] C. Hudelot, N. Maillot, and M. Thonnat. Symbol grounding for semantic image interpretation: From image data to semantics. In *Proceedings of ICCV 05*, pages 1875–1883, 2005.

[11] IST06. E-training for interpreting images of man-made scenes. http://www.ipb.uni-bonn.de/projects/etrims/.

[12] Z.W. Kim and R. Nevatia. Expandable Bayesian networks for 3D object description from multiple views and multiple mode inputs. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(6):769–774, 2003.

[13] N. Komodakis, G. Tziritas, and N. Paragios. Fast approximately optimal solutions for single and dynamic MRFs. In *Proceedings of CVPR*, pages 1–8, 2007.

[14] B. Neumann and T. Weiss. Navigating through logic-based scene models for high-level scene interpretations. In *Proceedings of Third International Conference on Computer Vision Systems*, pages 212–222, 2003.

[15] M. Petrou. Learning in computer vision: some thoughts. In *Proceeding of CIARP*, Santiago, Cile, 2007.

[16] M. Petrou and M. Xu. The tower of knowledge scheme for learning in computer vision. In *Proceedings of DICTA 07*, 2007.

[17] B. D. Ripley and N. L. Hjort. *Pattern Recognition and Neural Networks*. Cambridge University Press, 1st edition, 1995.

[18] G. Schneider, H. Wersing, B. Sendhoff, and E. Korner. Evolutionary optimization of a hierarchical object recognition model. *IEEE Transactions on system, man, and cybernetics - part B: cybernetics*, 35(3):426–437, 2005.

[19] Seminar. Logic and probability for scene interpretation. In *Dagstuhl Workshop*, page http://www.dagstuhl.de/en/program/calendar/semhp/?semnr=08091, 2008.