SEMANTIC OBJECT SEGMENTATION BY DYNAMIC LEARNING FROM MULTIPLE EXAMPLES

Yaowu Xu[1,2], Eli Saber[1,3], A Murat Tekalp[1,4]

1. Dept of Electrical and Computer Engineering, Univ. of Rochester, Rochester, NY 14627, USA
2. On2 Technologies, Inc., 21 Corporate Drive, Clifton Park, NY 12065, USA
3. Xerox Corporation, 800 Phillips Road, Webster, NY 14580, USA
4. College of Engineering, KOC University, Sariyer, Istanbul, TURKEY

ABSTRACT

We present a novel "dynamic learning" approach for an intelligent image database system to automatically improve object segmentation and labeling without user intervention, as new examples become available, for object-based indexing. The proposed approach is an extension of our earlier work on "learning by example," which addressed labeling of similar objects in a set of database images based on a single example [1]. It utilizes multiple example object templates to improve the accuracy of existing object segmentations and labels. We also propose to use Normalized Area of Symmetric Differences (NASD) as the similarity metric in "dynamic learning", due to its robustness to boundary noise that results from automatic image segmentation. The performance of the dynamic learning concept is demonstrated by experimental results.

1. INTRODUCTION

Humans navigate through and retrieve samples from large image/video databases by means of semantic concepts, such as, objects, people, etc. However, most current multimedia systems can only process low level visual features, such as color, texture, shape, etc [2, 3] in an automatic fashion. "Learning" approaches have been proposed to automatically compute high-level semantic concepts from low-level visual features. These approaches can be classified as: 1) Learning from interactive user feedback, i.e. relevance feedback, and 2) Learning from examples without run-time user interaction.

Relevance feedback [4-7], requires user responses to indicate relevant or irrelevant items in a search to: 1) establish either positive or negative links between retrieved images and query objects, 2) update the weights of various feature dimensions in a given vector space; or 3) enhance the probability distribution of a proposed Bayes model for the images in the database. Its potential drawbacks are slow convergence, sensitivity to user subjectivity, and inability to propagate the "knowledge" cultivated during the current query session to later queries. "Learning from example," on the other hand, attempts to create semantic abstractions for images containing regions determined to match user provided examples based on similarity of low-level visual features [1]. An attractive benefit of this scheme is automatic abstraction of semantic objects without user intervention.

This paper extends our prior methods [1] to "dynamic learning from multiple sequential examples." This extension poses new challenges in resolving potential conflicts between various semantic abstractions resulting from different example templates and rank ordering of similarities. The proposed "*dynamic learning*" scheme refines the segmentation mask of the object and updates the semantic abstractions when a new example provides a better match than previously existing one(s). Its main advantages are: 1) its self-correction capability, 2) its automatic nature requiring no user intervention, and 3) its flexibility for offline computation

The remainder of this paper is organized as follows. Section 2 presents a new shape similarity metric, which has such desirable properties. Section 3 introduces the proposed data representation and query strategies for the proposed dynamic learning system. Section 4 presents the concept and procedure for "*dynamic learning.*" Experimental results are presented in Section 5. Conclusions are drawn in Section 6.

2. A NEW SHAPE SIMILARITY MEASURE

The proposed dynamic learning procedure requires ranking of similarities among combinations of image regions and multiple objects templates, i.e., we need to find not only the best match to a given object template among many image regions, but also the best match between a given image region and many object templates. Therefore, it is desirable that the similarity measure be: 1) metric, i.e., it should not only be good for a threshold test, but also for ranking; and 2) symmetric, i.e., similarity

measure should be invariant whether it is computed in the image or template domain.

We propose a new shape similarity metric, called Normalized Area of Symmetric Differences, which is normalized to remove the effect of size of the candidate region or template on the similarity measure, that satisfy the above requirements. It is given by:

$$d(A,B) = \frac{(A-B)+(B-A)}{(A+B)}$$

where A and B represent two shapes, *(A+B)* is the area of the union of two regions, *(A-B)* denotes the area over by A but not by B, *(B-A)* is vice versa. The properties of the Normalized Area of Symmetric Differences measure include: 1) it is a metric, 2) it is robust against small changes in the shapes A and B, and 3) it is invariant to rotation, translation, and scaling.

Before computing the NASD, the contours of the shapes A and B are approximated by B-splines, and then registered in either the image or template domain. It is well known that B-spline representation and modal matching can suppress contour noise due to low-level segmentation errors. Hence, we adopted the modal matching approach[1] to establish feature correspondences between the template and candidate region. These correspondences are then used to estimate the affine transform parameters between the two shapes. Upon computation of the affine transform parameters, the image region and template are registered, and the NASD is calculated.

3. DATA STRUCTURE AND QUERYING

We start with a short review of the image representation and data structure used in our original learning method (referred to as static learning here). We represent images by a "scene graph" which consists of a tree that indicates the parent-child relationships between high-level objects and low-level (elementary) image regions, and an adjacency matrix that captures the spatial relationships between these elementary regions [1]. The root node of the tree corresponds to the whole image. Each leaf node (also called elementary node) represents a homogeneous image region with uniform color or texture. "Learning from example" refers to storing those combinations of regions that are similar to the example objects, in the form of composite nodes. The implementation of the learning process requires searching all valid combinations of elementary regions (as determined by the adjacency matrix) in an image for shape and/or color similarity to a user provided example template. A match is established when the similarity measure between a particular combination of elementary nodes and the example template is less than a pre-determined threshold yielding a composite node containing the matching combination of elementary nodes. The composite node provides a level of semantic knowledge over and above the original scene graph containing only low-level nodes. As a result, subsequent searches using the same example template would immediately identify the composite node as a match without processing its lower level.

For dynamic learning from multiple examples, we introduce a new data structure for each database image, called Object vs. Template Similarity Table (OTST) (see Table 1), where each row and column corresponds to a potential object and an example template, respectively. OTST provides a means for ranking the similarity of each composite node vs. each example template.

Table 1 Object-Template Similarity Table for Image *I*

i\j	Template1	Template2	…
C1	D (C1, T1)	d(C2, T2)	…
C2	D (C2, T1)	d(C2, T2)	…
…	…	…	…

Before any learning can take place, the database images have only low-level regions; hence the OTST is a null table. For each image *I,* the OTST is populated, as learning takes place with the introduction of each example template T_j. Matches found during similarity search between T_j and combinations of elementary nodes lead to creation of new composite nodes. These are entered in the OTST as new rows. Their similarity to other example templates is also computed and logged into the OTST. A reject threshold, which is introduced only to keep the size of the OTST reasonable, can be set rather loosely, and it is the only threshold involved in the proposed dynamic learning procedure

The query engine supports three types of queries using the OTST and returns the best *N* matching database images to the query (example) template. These are: 1) Query by known example: the query template is already in the OTST, only ranking of values needs to be done at the time of query, 2) Query by new example: the system first updates the OTST as described above; the updated OTST is then used to generate the rankings, and 3) Query by keyword: there is no specific labeling information stored in the OTST, Links between object templates and keywords may be established at the query user interface, then system performs query-by-template.

4. DYNAMIC LEARNING

In our original static learning method [1], composite nodes (the grouping of the elementary nodes) stay permanent once they are created. Depending on which example template is presented first, it is possible that a composite node is formed by a non-optimal grouping of elementary nodes. In static learning, there is no possibility of updating a composite node with new examples. The "*dynamic learning*" concept rests on the assumption that a portion of a "real" life object is less similar to a given template than a complete object is to its own template. To this effect,

composite nodes that are already established can be later updated when new example templates become available (self-correction). Hence, the grouping of elementary nodes is dynamic and the learning is ongoing as new example template become available.

4.1 Dynamic Learning Concept

The strategy of updating existing composite nodes can be summarized as follows: When an existing composite node is found to also match a new (later) example template, the low-level regions making up the composite node and all neighboring regions are re-searched to find if a better match to the new example template exists. When such search yields a different more optimum grouping of elementary nodes that is better matched to the object template than the existing composite node matches to any of the existing templates, the existing composite node is destroyed and a new composite node is created

The main concepts of "static vs. dynamic learning" are illustrated by the example in Figure 1. In general, when a new object template is introduced in the database, all the images are searched for the object as part of the "learning" process. The search can be performed either online, directly as the user is retrieving images through "query by new example", or offline by employing the user search profile. Either way, each image in the system is searched for the new object template by applying the hierarchical content matching strategy described above.

4.2 Guided Search Procedure for Dynamic Learning

Guided search refers to finding the best matching composite node C* to a new example template T in the neighborhood of an existing composite node C, taking advantage of the established match between T and C. The first step is to set up the search scope based on information provided by the existing match. As presented in [1], correspondences between the image region and the template have been established in the matching process. These correspondences are employed to estimate the affine transform that maps the object template into the image domain. This mapping is then utilized to classify all elementary nodes in three categories: 1) elementary node is fully covered by the template{F}, 2) elementary node is partially covered by the template{P}, and 3) elementary node doesn't intersect with the template {N}. We limit the scope of the search to {F}+{P}, thereby significantly reducing the computational complexity of the search.

The second step is to find the best match to the template in the search scope {F}+{P}. To this effect, all nodes in {F} are pre-determined to be part of any potential match thereby taking full advantage of the previously known best match. Hence, the procedure reduces to determining whether each of the elementary nodes in {P} should be incorporated into the existing composite node to form a more suited match. This is accomplished by computing a corresponding matching score using the techniques discussed in Section 2. The "closest" combination to the object template is compared against the similarity of the existing composite node. The composite node is rebuilt with a new grouping of elementary nodes if it yields the best similarity measurement.

There are a couple of advantages in dynamic learning with the described search strategy: 1) It automatically corrects inaccurate groupings of elementary nodes stored in existing hierarchical content descriptions yielding a better and more accurate semantic abstraction for the images, and 2) It significantly reduces the computational cost of finding the best match to the new example template by taking advantage of the previously established match.

5. EXPERIMENTAL RESULTS

5.1 Shape Matching Similarity Measure Comparison

As we discussed in Section 2, the NASD similarity measurement has several distinct advantages over the previously utilized Hausdorff distance [1]. Here, in Figure 2, we present a direct comparison between these two metrics. Table 2 provides the results for the manual versus automatic segmentations using the Hausdorff and NASD metrics. From the table, it can be easily seen that both measures increased (depicting lesser similarity) for the automatic segmentation (Figure 2c) when compared with the "ground truth" (Figure 2d); a reasonable expectation since most automatic segmentations are much less accurate than their human prepared counterparts. However, the Hausdorff distance increased much more drastically compared to the NASD as demonstrated by the percentage difference in Table 2. Hence, the NASD is much less sensitive to boundary noise and segmentation errors.

Table 2 Comparison of Similarity Measurements

	Hausdorff Distance	Normalized Area of Symmetric Differences
Semi-manual Segmentation	8.663	0.276
Automatic Segmentation	14.623	0.300
Difference in %	68.8%	8.7%

5.2 Dynamic Learning Experiments

To test the performance of the dynamic learning scheme for object based image labeling, experiments were performed on a large dataset using multiple object templates. Figure 3 shows representative images. As can be seen from Figure 3c, the search of "Sedans" template yields "SUV" type objects since they have a close similarity. Subsequently, a composite node, capturing the initial regions in the segmentation map that correspond to the object, is introduced into the content hierarchy for each image. However, a close examination of the first two images reveals that their corresponding composite node has succeeded in capturing the regions of the "SUV" that

match that of a "Sedan" and did not incorporate the "hatchback" portion of the "SUV" since it does not match well to a "Sedan". In the "static learning" scheme, the composite nodes generated after searching for Template 1 are "permanent" without any opportunity for updates.

Table 3 Results for Static vs. Dynamic Learning

OTST by Static learning			OTST After dynamic learning		
Comp. Nodes	T. 1	T. 2	Comp. Nodes	T. 1	T. 2
Img1/C1	0.202	0.408	Img1/C1*	**0.211**	**0.123**
Img2/C1	0.181	0.293	Img2/C1*	**0.234**	**0.176**
Img3/C1	0.095	0.230	Img3/C1	0.095	0.230
Img3/C1	0.208	0.314	Img3/C1	0.208	0.314

To improve this scenario, we utilize the proposed concept of dynamic learning. Figure 3e, 3f and 3g show the result of dynamic learning using Template 2. The process did not affect the last two images since they are not "SUVs". For the first two images, the search confirms that incorporating the "hatchback" regions (the previously missing regions) yields a better similarity to Template 2 than the previously formed node (see Figure 3d) to Template 1. At this point, the composite node hierarchy is reconstructed to reflect the above. Furthermore, the OTST is updated accordingly as shown in Table 3.

6. CONCLUSION

This paper presents a novel approach to dynamically improve object segmentations and labeling without user intervention for object-based indexing. "D*ynamic learning*" process updates the links between composite nodes and groupings of elementary nodes as new examples are introduced. Its main advantage is that inaccurate semantic abstractions of images are automatically corrected in the process of learning from new examples; In addition, "*dynamic learning"* does not require any user intervention; and can be performed offline.

REFERENCES

[1] Y. Xu, E. Saber, A. M. Tekalp, "Object Segmentation and Labeling by Learning from Examples", *IEEE Trans. on Image Processing*, 12(6)(2003) 627-638.

[2] Y. Rui, T. S. Huang, S.-F. Chang, "Image retrieval: current techniques, promising directions and open issues", Journal of Vis. Comm. & Image Rep., 10(4)(1999) 39-62

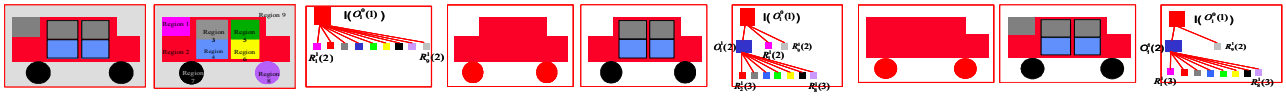
[3] A.W.M. Smeulders, M. Worring, S. Santini, A. Gupta, R. Jain, "Content-based Image Retrievals at the End of the Early Years," IEEE Trans. on PAMI, 22(12)(2000) 1349-1320

[4] Y. Wu, A. Zhang, "A Feature Re-weighting approach for Relevance Feedback in Image Retrieval", ICIP'02, Rochester, New York, USA, 2002

[5] Y. Rui, T.S. Huang, M. Ortega, S. Mehrotra, "Relevance Feedback: A Power Tool for Interactive Content-Based Image Retrieval", IEEE Trans. On CSVT, 8(5)(1998) 644-655

[6] I. J. Cox, M.L. Miller, T.P. Minka, P. N. Yianilos, "An Optimized Interaction Strategy for Bayesian Relevance Feedback" CVPR'98, Santa Barbara, CA, USA, 1998

[7] X. He, O. King, W.-Y. Ma, M. Li H.-J. Zhang, "Learning a Semantic Space From User's Relevance Feedback for Image Retrieval", IEEE Trans. on CSVT, 13(1)(2003) 39-48

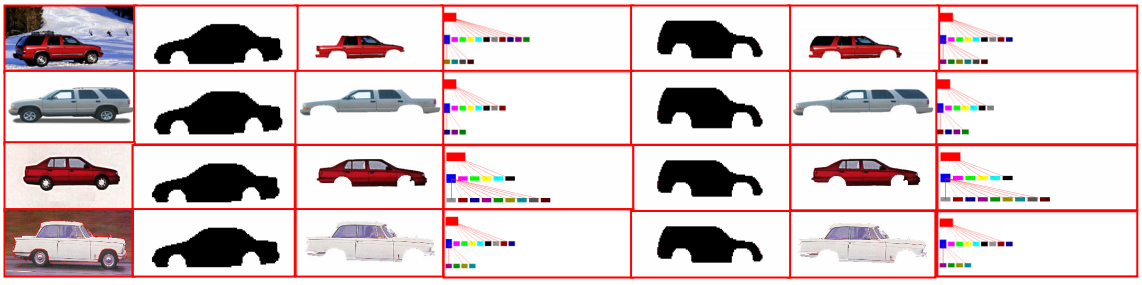


(a) (b) (c) (d) (e) (f) (g) (h) (i)

Figure 1: Dynamic Learning Concept: a) Original image; b) Low-level segmentation; c) Initial Content Hierarchy; (d) Object Template;(e) Matching Result; (f) Content Hierarchy after Initial Learning; (g) New Object Template; (f) Matching Result to New Template; (g) Content Hierarchy After Dynamic Learning

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

Figure 2: Noise Sensitivity of NASD:(a) Template (b) Input Image (b) Automatic Segmentation (c) Manual Segmentation



(a) (b) (c) (d) (e) (f) (g)

Figure 3: Experiments for Dynamic Learning: a) Original image; (b) Template 1;(c) Matching Result; (d) Content Hierarchy after Initial Learning; (e) Template 2; (f) Matching Result to New Template; (g) Content Hierarchy After Dynamic Learning *(Note the changes in hierarchy for the first two "SUV" images while the last two remained unchanged)*