On prototyping Human Image Annotation

Terry Caelli
Department of Computing Science
Universit yof Alberta, Canada, T6G 2H1
tcaelli@ualberta.ca

Abstract

This paper inv estigates a different perspective to Image Understanding: the design of image processing, learning and matching algorithms for tracking and predicting how humans annotate images. Of specific interest is the class of prototyping models and relational learners that generate Horn-clause type annotation rules for binding the properties of image features and their attributes with domain models of scenes.

1 Introduction and Philosophy

T ypically image understanding and annotation systems hav e three basic components. One, a representation or model for the domain which is being sensed. Two, the selection and tuning of feature extractors that can index domain knowledge structures. Three, binding what is sensed with what is kno wnand the process of updating kno wledgeand what to sense (what features to extract) from new observations. In recen t years our approach has been to pose issues related to the *selection and tuning* of algorithms for these three components in terms of Machine Learning[1]. In this paper w eillustrate this approach via one of our systems, CITE. CITE is designed to "prototype" [1] how humans annotate images by creating symbolic descriptions of domain models and image feature states which are most compatible with the dynamics of image annotation *as it is p erformed by the annotator.*

For this reason w e hav e adopted an incremental Explanation-Based Learning (EBL) approach to perceptual learning and adaptation[1]. The end result of such an approach is to con vertimage understanding into a decision-support HCI technology to be incorporated within strategic application areas like Cartography. This view is different from past systems like, for example, SCHEMA[3, 2] and SIGMA[4] where implementations of the three basic components discussed above, although adaptive, are not learned from h uman behaviour.

More formally, then, the task of the learning procedures is to generate annotation rules which are consistent with human behaviour in terms of propositional and Horn clauses which encapulate the three components described above. First, domain knowledge models are defined by a set of clauses which encapulate the scene hierarchical structures like:

```
Building(i) <- Roof(i,Y)^Above(i,Y,X)
                  ^Wall(X)..etc.
Suburb(Z) <-Building(i)^Building (j)
                  ^NextTo(i,j)..etc.
```

Image features used to evidence specific terms of the domain model are defined by rules such as:

```
Wall( ) <- Size(x(1),,x(n))^
           Colour(y(1),,y(m))..e tc.
NextTo(i,j) <- distance(i,j)
           ^overlap(i,j)..etc.
```

The core of the EBL approach is essen tially a generalization of the well-known E-M (Expectation-Maximization) algorithm since, at any giv en time

1. CITE performs an annotation based upon the curren t model and feature extraction states. This is executed by a form of constraint satisfaction using hierarc hical relaxation labeling of the form:

   B Let **B** be a set of objects $\{b_1, ..., b_n\}$, and $\mathbf{\Lambda}$ be a set of labels $\{1, ..., m\}$. F or each object b_i w ecan obtain an initial label probability v ector $\mathbf{P}_i^0 = (p_{i1}^0, ..., p_{im}^0)$ where $0 \leq p_{ij}^0 \leq 1$, for $i = 1...n$ and $j = 1...m$, and $\sum_j p_{ij}^0 = 1$, for $i = 1...n$. Each initial probability v ector is interpreted as the prior probability distribution of labels for that object, and can be computed from the initial unary and binary matching. The compatability constraints can be described as an $n \times n$ block matrix $\mathbf{R}$, where each R_{ij} is an $m \times m$ matrix of non-negative real-v alued compatibility coefficients, denoted $r_{ij}(1..m, 1..m)$. The coefficient

[1] In this con text "Protot yping" refers to the process of generating procedural descriptions for how a specific task is performed by an expert

$r_{ij}(\lambda, \mu)$ is a measure of the compatibility betw een object b_i being labeled λ and object b_j being labeled μ. The relaxation labeling algorithm iterativ ely updates the probability vectors $\mathbf{P}$ using a normalized weigh ted sum equation:

$$p_{i\lambda}^{t+1} = \frac{p_{i\lambda}^{t} q_{i\lambda}^{t}}{\sum_{\mu=1}^{m} p_{i\mu}^{t} q_{i\mu}^{t}} \quad (1)$$

where the denominator is the normalization factor and:

$$q_{i\lambda}^{t} = \sum_{j=1}^{n} \sum_{\mu=1}^{m} r_{ij}(\lambda, \mu) p_{j\mu}^{t} \quad (2)$$

The objective is to end up with a unique label for each object. Depending on the compatibility coefficients $r_{ij}(\lambda, \mu)$ it is not possible to guarantee convergence. In this form of relaxation labeling, the n umber of objects is constant and the number of iterations, t, is the same for each object. However, CITE con tains hierarchical knowledge and can generate and remove image regions dynamically.

2. The expert then updates the domain model by revision or addition of new terms if CITE cannot interpret features (cannot resolv e uncertainty of matching models to data) or produces incorrect feature annotations according to the expert.

3. Induction then occurs ov er the (labeled) feature attributes which define the range of numerical values for each term (feature and model components) which are consistent with the current interpretation (see Figure 2).

4. processes 1 to 3 iterate ov er the course of an annotation session until the end domain model, feature extraction parameter states satisfy the consistency criteria operating in the relaxation procedure or expert evaluation of the output.

Figure 1 show more details of CITE's learning environment. An initial segmentation (1) of the image causes the unary (part) feature processor to compute features for each of the low lev el regions (2). These features are then matched with the knowledge base (3) to provide initial labeling hypotheses which are represented in the indexing structure called the *scene interpr etation*. Clique resolving and hierarchical binary matching then occur on these initial hypotheses (5) using binary (relational) features calculated from hierarchical segmentation (4). The higher level scene hypotheses are added

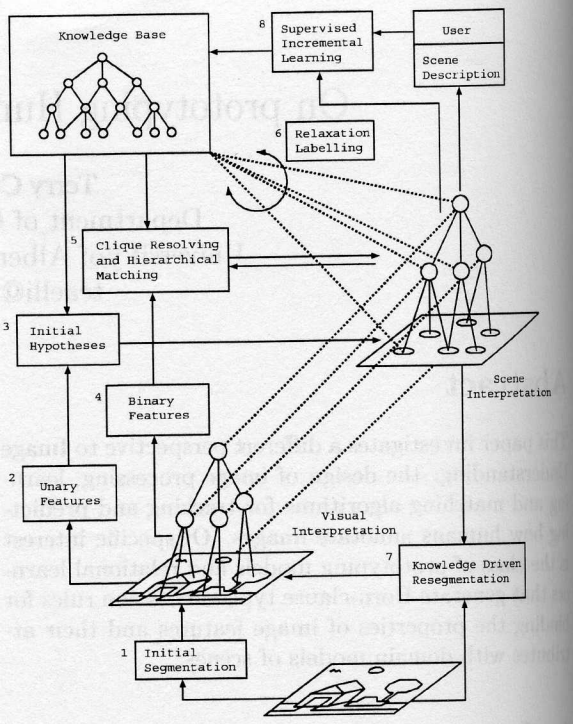


Figure 1: Ov erview of CITE's Architecture.

into the scene interpretation structure, and hierarchical relaxation labelling (parallel constraint satisfaction: see belo w) begins to resolve the multiple ambiguous labels for each object (6).

As the labels begin to resolv e, nodes are individually resegmented (7) using parameters stored in the knowledge base. These resegmentations replace the initial segmentations in the visual interpretation structure, resulting in a repeat of the unary and binary feature extraction and matching (stages (2) through (6)). This cycle continues a number of times until the in terpretation becomes stable for current and past examples.

Critical to CITE is the role of the human operator. If, after the annotation is complete (the system has conv erged on the most parsimoneous hierarchical annotation - at this time) and CITE's final interpretation is incorrect, the user may choose to edit the annotation by selecting the incorrectly labeled features, nodes and the corresponding knowledge base node (8). The updated knowledge base is then av ailable as the next scene is viewed - and the process continues. This form of incremental supervised learning provides CITE with the abilit y to increase the descriptive pow er and accuracy of its analyses, as well as add new world knowledge as it becomes av ailable, rather than requiring the full set of scene objects to be present during an initial learning

phase.

In the following some details of CITE's processes are briefly discussed.

2 The CITESystem

2.1 Knowledge Representation (KR)

CITE represents world knowledge as a (directed) lattice in which each node represents an object or visual concept which is either a **part-of**, a **view-of** or a **type-of** its parent or parents. Within each node information is stored about the (historically) best segmentation(for example, K-Means and Nearest Neighbor clustering-based segmentation algorithms are available to choose from), feature extraction and matching algorithms (for example, pixel correlations, similarity of region statistics, localized graph similarity measures - depending on the type of node) that are used to recognize a given object in an image. CITE represents taxonomies using the type-of node (see Figure 3).

2.2 Interpretation Modules

CITE contains hierarchically segmented image data, again, in the form of a lattice called the *visual interpretation*. Multiple labeling hypotheses can be generated for each node in the visual interpretation and these are stored in an intermediate structure called the *scene interpretation*

2.3 Visual Interpretation

CITE uses hierarchicalsegmentation in which regions are groupedto form larger regions which are grouped to form even larger regions - and so on. There is only one type of node in the visual interpretation, called the VI node, which may contain multiple parents and multiple children. Multiple children represent the grouping of smaller regions with common properties or labels into a larger region with a single label. Multiple parents represent the notion of ambiguous set membership, which is important in solving the clique (grouping) problem.

As already described, CITE also uses a process of parallel hypothesis generation to generate multiple likely solutions (local constraints) and then relaxation labeling to propagate these hypotheses to maximize global consistency. Each VI node also contains a list of pixels that the node represents in the original input image. There is no ordering or adjacency limitation placed on these pixels, which means that a VI node can be used to represent multiple non-connected regions. The final important component of the VI node is that image features are stored when they have been calculated. Unary (part or region) features are stored in the VI node and binary features are stored in the common parent of the multiple VI nodes between which they are calculated.

2.4 Scene Interpretation

The scene interpretation (SI) structure is an indexing structure which connects the visual interpretation to the knowledge base and represents the current scene states.

3 Learning Domain Knowledge

In CITE knowledge is built using incremental forms of supervised learning which track the user's annotation preferences - involving attribute generalizations over the vertices and edges of labeled, attributed and directed graphs which encode the inherent lattice-like relational structures of the knowledge. Learning of both object parts and part relations involves incrementally updating their attributes (part: unary, relation: binary) via the changes to bounding hyperrectangles which enclose attribute values which maximally evidence each model or object and is consistent with the constraints on the interpretation as currently represented in the KR lattice. This constrained iterative attribute splitting procedure is termed "Explanatory Least Generalization" (ELG) as, unlike standard least generalization techniques such as Conceptual Clustering[5] there are constraints on the selection of candidate samples via prior knowledge and vice-versa. The algorithm is illustrated in Figure 2.

4 Hypothesis Generation and Evaluation

CITE generates classification labels and grouping hypotheses in the visual interpretation and scene interpretation structures based on segmentation results, unary and binary feature matching and examination of knowledge base data structures. Again, hypotheses exist at many different levels within CITE and are resolved by a process of hierarchicalrelaxation labeling and constraint propagation.

4.1 Resolving Ambiguities

There are typically multiple labeling and grouping hypotheses generated by CITE for any image region or set of image regions. These multiple hypotheses are resolved by a process of parallel constraint propagation,

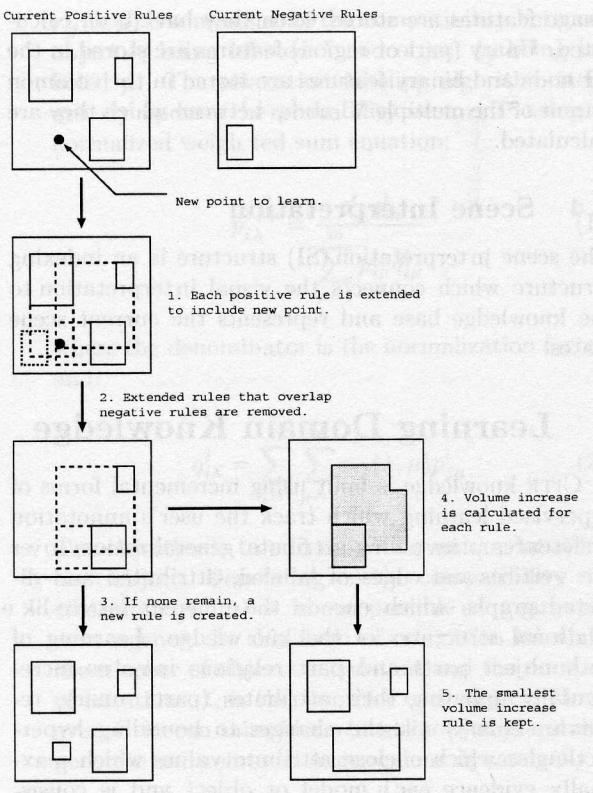


Figure 2: Explanatory Least Generalization used in CITE. Here, learning involves "covering" observed positive example feature attribute states which are consistent with the current knowledge representation and avoiding negative examples. Here rules are defined by hyper-rectangles in both Unary and Binary feature attribute spaces.

in this case, relaxation labeling[1]. This is a method of resolving multiple hypothesis labeling with respect to a compatibility function describing the compatibility between pairwise labels as illustrated in Figure 1.

4.2 Knowledge Driven Segmentation and Resegmentation

In CITE, segmentation is a controlled process in which the current expectation of scene content is used to determine which segmentation algorithm and parameterization is to be used. This is achieved by storing in each knowledge base node a list of those segmentation procedures and parameterizations which have been the most effective at segmenting the object or scene element represented by that node. Here we have used three different data-driven segmentation procedures: clustering, region growing, and edge extraction and edge merging.

4.3 FeatureExtraction

Features calculated on single regions (*unary* features) typically include colour, texture, size and shape statistics (means, variances) of regions. Relational (*binary*) features are calculated on pairs of regions and typically concentrate on geometric properties such as the distance between the regions and the lengths of common borders. CITE uses unary and binary features, which can be computed across the visual interpretation hierarchy.

5 System Performance and Results

We have examined four different scene types of offices, streets, outdoor and airports. In total, 253 scene elements were detected from 38 images of the four scenarios. The knowledge bases for the scenarios were constructed from a collection of 85 images. The knowledge base contained a total of 155 scene elements at different levels of abstraction. Of the 253 detected scene elements, an average classification success of 96% was achieved. In the more complex street scene example shown in Figure 3, a total of 24 scene elements have been detected. Of these, one was misclassified and three were the result of spurious segmentations. One of the spurious segmentations, the shadow under the fuel-truck, generates an incorrect "trunk" object. The other two were under and over segmentations, but are correctly labeled as "grass'.

Figure 3 shows the resultant symbolic interpretation of an outdoor scene image involving a domain knowledge base consisting of houses, fueltrucks, trees, roads and related basic objects; higher-order objects such as diary ,ground, sky - all being composed of more basic parts and resulting in the interpretation shown in the bottom of the figure. Here the numbers associated with each label define the certainty (0-1) of the labeling.

6 Conclusions

This paper presents the view that there is good reason to revise interest in image annotation technologies when they are seen as an important class of decision-support procedures when can be used by professionals as part of more advanced HCI tasks as, for example, occur in Cartography. In that case experts are required to provide a "symbolic" layer of maps which must retain consistencies between feature labels within the context of on-going map revision.

For these reasons this approach does not propose that the ultimate aim is to create fully automated an-

notations but rather to provide experts with advanced annotation tools which have be learned from previous performance of experts in similar domain models and image types. Consequently the real issues for such adaptive and trainable annotation systems are those related to the development of appropriate learning and matching algorithms such as those used in this example - CITE.

Feature-indexed learning is presented as a general representation which appears to function beyond the initial encoding level and we show how CITE uses domain knowledge to constrain rule generation and, in turn, update knowledge. This approach is quite different from current learning models which do not constrain the rule generation process via feature relations and hierarchical scene models. Of particular importance is the use of domain knowledge to resegment what is sensed, the use of incremental supervised learning methods, and parallel constraint satisfaction (hierarchical relaxation labeling) to derive the consistency of new data with current knowledge.

Such a general view of Computer Vision may bring more robustness and purpose to the area while also retaining the investigation of fundamental issues in perceptual learning and adaptation.

References

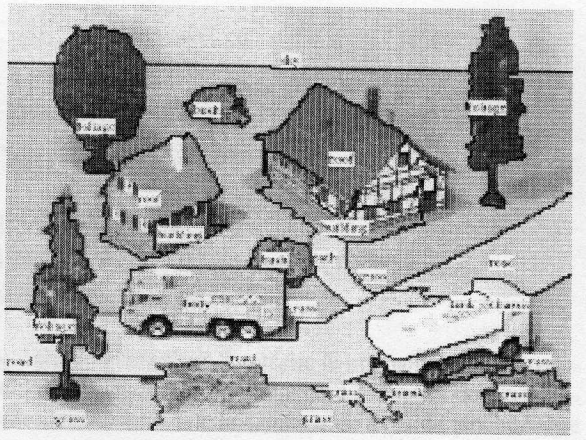
[1] T. Caelli and W. Bisc hof. *Machine Learning and Image Interpretation.* Plenum, 1997.

[2] B. Draper, A. Hanson, and E. Riseman. Knowledge-directed vision: Control, learning, and integration. *Proceedings of the IEEE*11(84):1625–1637, 1996.

[3] B. A. Draper, R. T. Collins, J. Brolio, A. R. Hanson, and E. M. Riseman. The schema system. *International Journal of Computer Vision*, 2:209–250, 1989.

[4] T. Matsuyama and V. Hwang. *SIGMA - A Knowledge-based Aerial Image Understanding System.* Plenum Press, 1990.

[5] T. Mitchel. *Machine Learning* McGraw-Hill, New York, 1997.



```
World[0] (1.000) Consisting of:
    ├─ sky[25] (0.876)
    ├─ ground[26] (1.000) Of Type:
    │   └─ grass[1] (0.825)
    ├─ pencilpine[27] (1.000) Constructed from:
    │   ├─ foliage[9] (0.932)
    │   └─ trunk[5] (0.878)
    ├─ pencilpine[29] (1.000) Constructed from:
    │   └─ foliage[22] (0.922)
    ├─ pencilpine[31] (1.000) Constructed from:
    │   └─ foliage[23] (0.939)
    ├─ house[36] (0.666) Constructed from:
    │   ├─ building[19] (1.000)
    │   └─ roof[18] (1.000)
    ├─ fueltruck[44] (1.000) Constructed from:
    │   ├─ chassis[11] (1.000)
    │   └─ tank[10] (0.920)
    ├─ dairy[45] (1.000) Constructed from:
    │   ├─ roof[21] (0.934)
    │   └─ building[20] (1.000)
    ├─ ground[46] (1.000) Of Type:
    │   └─ grass[2] (0.868)
    ├─ ground[47] (1.000) Of Type:
    │   └─ grass[3] (0.848)
    ├─ ground[48] (1.000) Of Type:
    │   └─ grass[4] (0.563)
    ├─ ground[49] (1.000) Of Type:
    │   └─ road[6] (1.000)
    ├─ ground[50] (1.000) Of Type:
    │   └─ grass[7] (0.687)
    ├─ ground[51] (1.000) Of Type:
    │   └─ road[8] (0.830)
    ├─ firetruck[52] (1.000) Constructed from:
    │   └─ body[12] (0.842)
    ├─ ground[53] (1.000) Of Type:
    │   └─ grass[13] (1.000)
    ├─ ground[54] (1.000) Of Type:
    │   └─ grass[14] (0.716)
    ├─ ground[55] (1.000) Of Type:
    │   └─ path[15] (1.000)
    ├─ ground[56] (1.000) Of Type:
    │   └─ road[16] (1.000)
```

Figure 3: Top: Labeled scene. Bottom: Text Description of Street Scene. Here labels in the instantiated knowledge base (bottom) refer to part labels in the knowledge base and numerical values correspond to certainty of the labeling (0:guess - 1:most certain).