1 Introduction

Wherever you happen to be looking, some things are simply more interesting than others. That cute research student is far more interesting than this dull poster, for example. It is natural for us to take in more information about the interesting object(s) than the uninteresting ones. In computer vision, image segmentation and partitioning is a major research area. However, much of the work in the field ignores this simple fact, building a segmentation without thought as to how interesting different objects are.

The goal of my research is to generate an image partitioning that also functions as an interestingness measure, detailing interesting areas with a fine partitioning, while abstracting uninteresting areas with a coarser partitioning, as shown in Figure 3.

2 How can we do this?

We start with a fine partitioning of our image. Then, we acquire user input in the form of pairs of partitions which should remain separate, either because they’re interesting (like a tree) or different, and which should be stuck together because they are dull. We then want to learn a model which can infer these labels, merge or nonmerge, for some unseen set of partitions.

2.1 A New Graph Representation

Classical graphical models for images have difficulty handling the user data we are working with. As such, we don’t use the classical model. Rather, we work with a graph derived from the edges of the traditional model, as shown in Figure 1.

To infer the label that should apply to each of the nodes in $\mathcal{G}^*$, we train an SVM classifier to produce a probability that the partitions should merge using a combination of the data contained in the pair of partitions associated with it. We use this classification result, together with a simple smoothness model, to infer the most likely label for each node using a Graph Cuts algorithm.

2.2 Translating from the edge graph to labels

Unfortunately for us, just knowing merge and nonmerge instructions is not enough to actually produce an informative image partitioning. Once we have our set of labels on $\mathcal{G}^*$, we have to translate this into a set of partition labels over the image. Even more unfortunately, this translation does not necessarily have a unique solution, and there are numerous ways of resolving this, as shown in Figure 2.

The way we handle this ambiguity depends on the application. If we want large areas to be smoother, we use (a), while if we want to ensure different things are never merged we use something like (b) or (c). You can see the difference between these solutions in Figure 3(c) and (d).

3 Conclusions

Using this merging approach we produce an image partitioning which has a strong relationship to how interesting a user finds parts of the image, while still separating objects which are different from one another. This type of partitioning can then be fed into other algorithms for classification, compression or other tasks.

We are presently investigating numerous methods for improving our technique, including more robust features, improved models for classification on $\mathcal{G}^*$ and a more intelligent loop condition resolution. We are also looking at methods for encapsulating more than two interestingness levels.