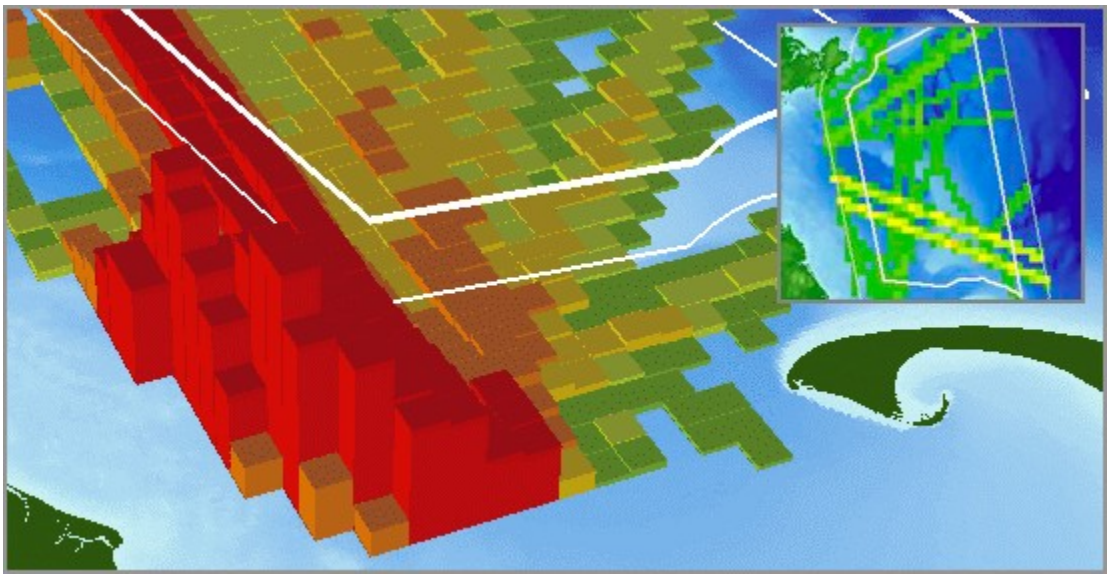
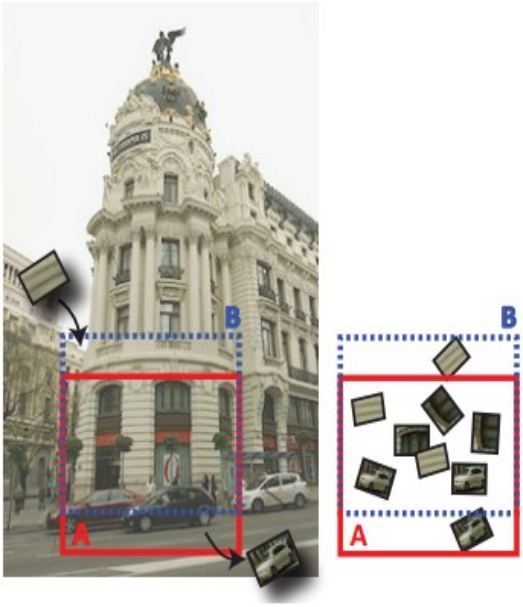
Image analysis by counting on a grid



CMC MSU, October 2012

Alexander Chistyakov

Introduction



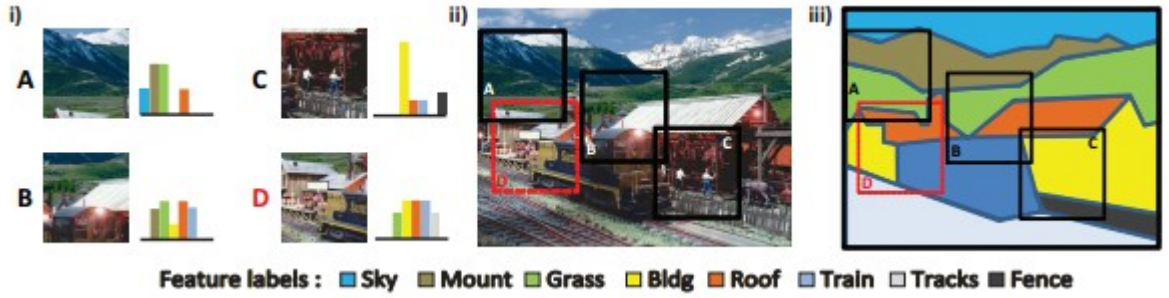
In recent scene recognition research images or large image regions are often represented as disorganized ”bags” of image features. For example, as a camera pans upwards from a building entrance over its first few floors and then above the penthouse to the backdrop formed by the mountains, and then further up into the sky, some feature counts in the image drop while others rise – only to drop again giving way to features found more often at higher elevations. The space of all possible feature count combinations is constrained by the properties of the larger scene as well as the size and the location of the window into it. Accordingly, our model is based on a grid of feature counts, considerably larger than any of the modeled images, and considerably smaller than the real estate needed to tile the images next to each other tightly. Each modeled image is assumed to have a representative window in the grid in which the sum of feature counts mimics the distribution in the image. The method described in this article provides learning procedures that jointly map all images in the training set to the **counting grid** and estimate the appropriate local counts in it. Experimentally, we demonstrate that the resulting representation captures the space of feature count combinations more accurately than the traditional models, such as latent Dirichlet allocation, even when modeling images of different scenes from the same category.

Main idea

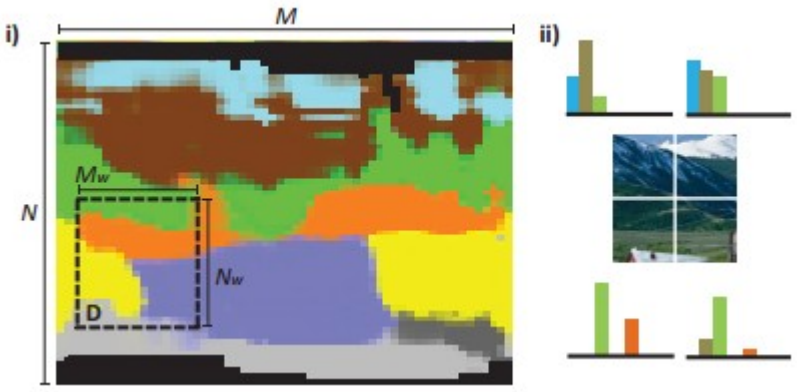
A popular way to deal with diversity of imaging conditions and geometric variation in objects or entire scenes is to simply represent images or image regions as disordered “bags” of image features. Ideally, these features should be highly discriminative so that most categories of images of interest are uniquely identifiable by the presence of a handful of features. In practice, however, individual features are not sufficiently discriminative, and modeling joint variation in feature counts becomes an interesting machine learning problem.

To understand this method better, let's see the one practical problem. For an

illustration, provides a synthetic example of several images *i)* of a train station, taken as windows into the larger scene *ii)*. Just for illustrative purposes, we hand-labeled the scene with feature labels as shown in *iii)*. Assuming that a few images are taken at random from the scene, we wonder if the feature counts in these images are sufficient to predict the possible feature counts in other images of the scene. In particular, we consider images taken from the regions close to A, B, and C in *iii)* and ask the question if the image D would fit the so defined train station class.



For this particular example, there is a need for interpolating between the feature count vectors for A,B,C and other images. However, this interpolation is best performed by spatial reasoning. Given that in some training images we see, from the top to bottom, roof, train, tracks, and in others mountain, grass, roof, train, we can infer that the existence of grass, roof, train, tracks combination is likelier than the existence of the mountain, roof, train, tracks combination of features. Furthermore, the proportions of different features in the images carry the information about the thickness of the layers of these features, which should be useful for inferring which previously unseen feature count combinations can be found elsewhere in the scene. Surprisingly, not much of the spatial organization of the features in the training images needs to be retained in order to perform the spatial reasoning about which feature combinations are likely. In illustration we show the counting grid inferred by iterating on the label counts from 50 windows into the scene taken at random, but avoiding all windows that contain all five of the features in D in any proportion. Each training image was represented as a set of 2*2 feature bags (upper left, lower left, upper right, lower right, see *ii)*), and without using the original location information, the counting grid was computed so that for each training image, a window into the counting grid can be found so that the appropriate sections have matching histograms. The resolution of the reconstructed feature layout of the large scene goes well beyond what would be expected from a crude 2*2 tessellation of the input images (the height of each section is roughly 20% of the large scene and only the feature counts in each section were used, not their spatial layout).



The counting grid model

The counting grid, $\pi(i;j;z)$ is a set of normalized counts of features indexed by *z* on the grid *(i; j)*. From numerical analysis we have an algorithm, which affords to maximize the possibility of good crossing ours pieces on grid space. Unfortunately, fourmulars are using in discribing of algorithm are so awful, that I did not presume to write them in this article.

Experiments

To show the power of this algorithm, let's analyze results of a set of experiments, executed by Microsoft's scientists.

On the picture *i)* we see a source image. Pictures *ii-v)* show a recovered image after a counting grids process with different parameters.

For example in experiment *iii)* they used *50* patches with 4 columns histogram and in the last experiment they used only *200* patches of size *16*16*.

Conclusion

In this article we saw an algorithm, which affords to recover scene by it's random pieces with a huge compression and rather good quality. Moreover, we learned to find a place on scene to new pieces.

However it is not the only way to use counting grids. The extending of *z*-axe in counting grid space allows us to analise video fragments and set a time interval for a bag of random reels.

Reference

1. http://profs.sci.univr.it/~perina/papers/cvpr2011.pdf
2. http://uai.sis.pitt.edu/papers/11/p547-jojic.pdf