Building part compositions for hierarchical object recognition

Greg Nicholas

May 15, 2009

1 Introduction

The visual signature of objects in our world follow a particular blueprint: every object is composed of an arrangement of smaller visual "parts". These parts can be thought of on many different scales. An image of an car can be considered to be pattern of wheel, window, door, and various other compositional parts, or on a much more granular scale as a massive pattern of interconnected line segment parts. These levels of granularity lend themselves well to a hierarchical organization of visual blueprints: the car can be decomposed into doors and other similarly-scaled parts, each door can be decomposed into a window, a handle, and other features, and this reduction in scale can continue until a suitable base level of elemental parts is reached.

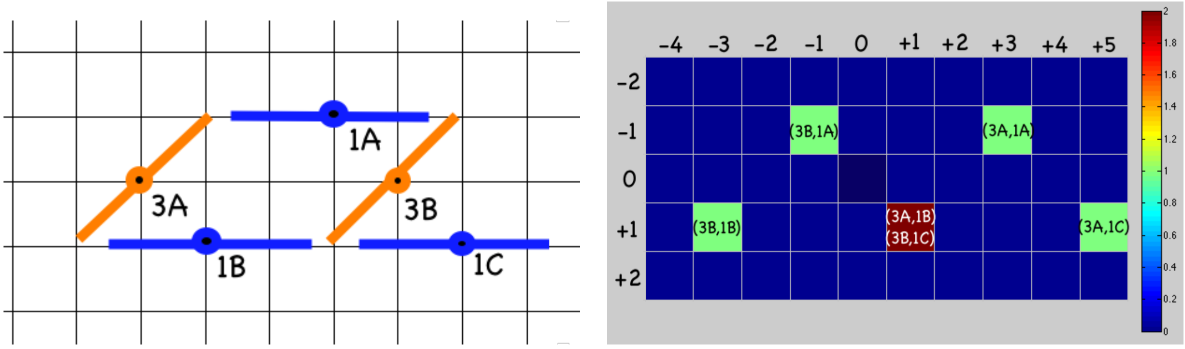
In their 2007 paper, "Towards Scalable Representations of Object Categories: Learning a Hierarchy of Parts" [3], Fidler and Leonardis described an algorithm to implement such a categorization scheme with minimal human intervention. Given a large corpus of images, they showed their algorithm organizing subparts into patterns completely automatically from the bottom elemental layer to parts three layers higher. At this level, parts became too specialized to learn with a generic image corpus, so further learning was done on manually separated sets of images grouped by categories (e.g. faces, mugs, etc.). According to the authors, this was the only source of significant human intervention in the process. This work has attempted to recreate their algorithm from scratch. Unfortunately, this task was not successfully completed. Although the entire framework of the algorithm was written, automating the core pattern processing part of it proved to be a very nuanced and difficult task to complete. This paper will discuss the nature of this difficulty and describe the various successes and setbacks in developing a principled way to process the patterns in the data automatically.

2 Spatial maps

Due to spatial constraints, it is necessary to defer to the source publication [3] for the full algorithm details. Before moving on however, the nature of the data from which the patterns are extracted must be discussed.

First, some termonology must be introduced. A *layer* is defined as a level in the hierarchy, starting with the elementary parts in layer 1, and building up from there. A *part definition* is a blueprint for describing how the composite subparts fit together. Note that part definitions in layer 1 are unique in that they are defined via features extracted directly from the image. More on this below. The first step in combining two part definitions (say, parts i and j) in layer n into a new composite part definition in layer $n+1$ is to create a *spatial map* for the two definitions. A spatial map for i and j is essentially a two-dimensional histogram that records the location of each occurrence of a "neighbor" part (say, j) relative to a "central" part (say, i) in a particular image, consolidated over all images. A spatial map created for a central part i with a neighbor part j is denoted as $\mathcal{S}_{i,j}$. A spatial map is created for each possible pairing of part definitions in a given layer (including pairs with identical definitions).

Figure 1 demonstrates the construction of a sample spatial map. The image on the left shows three occurrences of part 1 (colored blue) and two occurrences of part 3 (orange). The position of each occurrence of part 1 is measured relative to each location of part 3 and these coordinates are recorded in a 2D histogram. The spatial map shown on the right is this histogram. The source of each data point in the figure is labeled with the pair(s) of occurrences that generated it. Notice the global maximum of this histogram, location $(+1, +1)$. This configuration stands apart because the source image has two occurrences of the /_ pattern, and only one occurrence of the others. Note that this spatial map is built from only one source image, while a normal one is aggregated over an entire corpus of images. The source images used as data for this paper comes from a collection of 170 images featuring automobiles in outdoor scenes. The spatial maps extracted from these images record over 1.2 million points of data on average.



(a) Source image

(b) Spatial map

Figure 1: An example construction of $\mathcal{S}_{3,1}$. Each space on the grid is 1 pixel wide. Refer to the text for details.

The purpose of spatial maps is to measure the likelihood that a pair of parts will occur in a particular location relative to each other. Therefore, a location in map $\mathcal{S}_{i,j}$ which has a high value indicates a configuration that is a significant pattern between i and j. Assuming this configuration of two parts from layer n pass some other qualifications and thresholds described in [3], it will become the basis for a new part definition in level $n+1$. In particular, new definitions must include information about the spatial layout of its component parts, with some built-in variance information to allow for some flexibility. Given the spatial maps, the task at hand is to extract the significant peaks in the spatial maps and for each peak, find an area that captures a suitable amount mass related to that peak.

In this paper, the discussion of spatial maps will be constrained to those that map the location of two subparts relative to each other. Note that in [3] part definitions can include up to five component subparts, but the particulars of this is best deferred to the original publication.

In addition to the 2-subpart restriction, only spatial maps that describe configurations between layer 1 parts will be considered. Occurrences of parts in images for layer 2 and upward are of course discovered inductively, relying on the instantiations of the previous layer's parts. Level 1 parts, however, must be discovered directly from the image itself. In this work, these parts are defined as large responses to oriented Gabor filters (which have a strong correlation to biological models of low-level visual sensation, see [6]). A total of eight filters at different orientations are used (all in $[0, \pi)$, spaced apart by $\frac{\pi}{8}$). For more details on the Gabor filters, see [2].

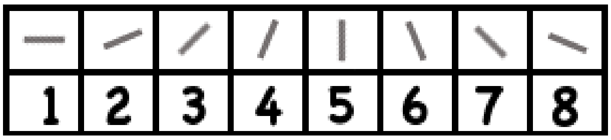
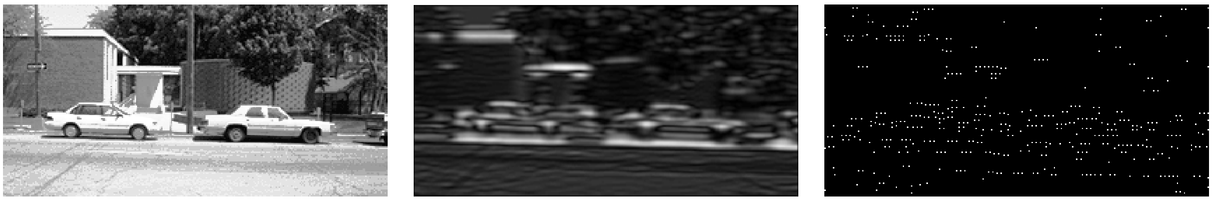


Figure 2: Instantiations of the eight layer 1 definitions.

Figure 2 shows eight example patterns that would result in a high response by each respective filter. Significant responses from these filters are found using a modified non-maxima suppression algorithm, and declared as instantiations of their respective layer 1 definition.

Figure 3 shows an image filtered by a Gabor filter designed to find horizontal patterns (definition 1), and the locations of the resulting instantiations.

Because of these constraints, the spatial maps discussed here can very literally be thought of as histograms showing the likelihood of different configurations of two small orientated image patches. This provides a fairly good intuition into why some of the spatial maps have the structure they do.



(*a*) (*b*) (*c*)

Figure 3: An image being processed to extract instantiations of part definition 1

For example, Figure 4a shows $\mathcal{S}_{1,1}$, which records the patterns between two horizontal patches. As might be expected, there is a highly significant pattern suggesting that two horizontal patches are often directly next to each other; any sustained horizontal line in an image (a common occurrence) falls under this pattern. Figure 4b shows $\mathcal{S}_{5,4}$, and again has an intuitive explanation: any gradual curve transitioning from a vertical patch in a clockwise direction will cause the peaks above and below the center of $\mathcal{S}_{5,4}$. However, most of the maps are not quite as clear cut. Figures 4c and 4d display $\mathcal{S}_{6,3}$ and $\mathcal{S}_{5,2}$, respectively, which are not nearly as intuitively laid out. These four examples show just how unique each map is compared to the others. Due to these unpredictable structures, finding a "one size fits all" procedure to automatically extract the peaks and a suitable area of variance around every spatial map is a difficult task. The remainder of this paper will discuss attempts at doing so.

3 Peak finding

At first glance, finding the peaks of the spatial map histograms seems to be a fairly straightforward task: simply extract them using some kind of maxima-finding technique. While this basic approach does end up being the most advisable method, other approaches to extracting the peaks are worth looking into as well.

3.1 Model fitting and clustering

As mentioned above, an area of variance around each peak must be found to complete the new part definition. Instead of attempting the peak and variance-finding as two separate tasks, it seems reasonable to combine the two into one approach. One way to do this is to assume that the landscape of a spatial map has a underlying parametric distribution, namely a mixture of Gaussians, and attempt to fit a model to it. The resulting parameters could then directly yield the locations of the peaks as well as their variances. However, this approach would assume a set number of peaks is known beforehand, which is a value that is not given.

Alternatively, it could approached as a clustering problem: cluster the occurrences measured in the spatial map and use the location and shape of these clusters to estimate the placement and variance of the peaks. While some clustering techniques may not be adept to this task for the same reason parametric model fitting isn't, there are others that do not require prior knowledge of the number of clusters. One of these, mean shift clustering, is the technique that was implemented, inspired by [1] and [4].

After running a mean shift procedure multiple times on a sample of points from the spatial map, the basins of attraction of the spatial map will begin to materialize. These basins form anywhere that the gradient of the spatial map's landscape will pull points towards the same "epicenter". This epicenter is a natural point to consider as a peak. Not only is this point most often a local maximum, but it is also a significant local maximum since the gradient leading to it is strong enough to overtake other, less significant maxima in the area.

In one sense, this strong magnetism is a desirable feature. Because of it, there is no real need to make a decision about which peaks should be strong enough to make the cut: mean shift procedure has already decided.

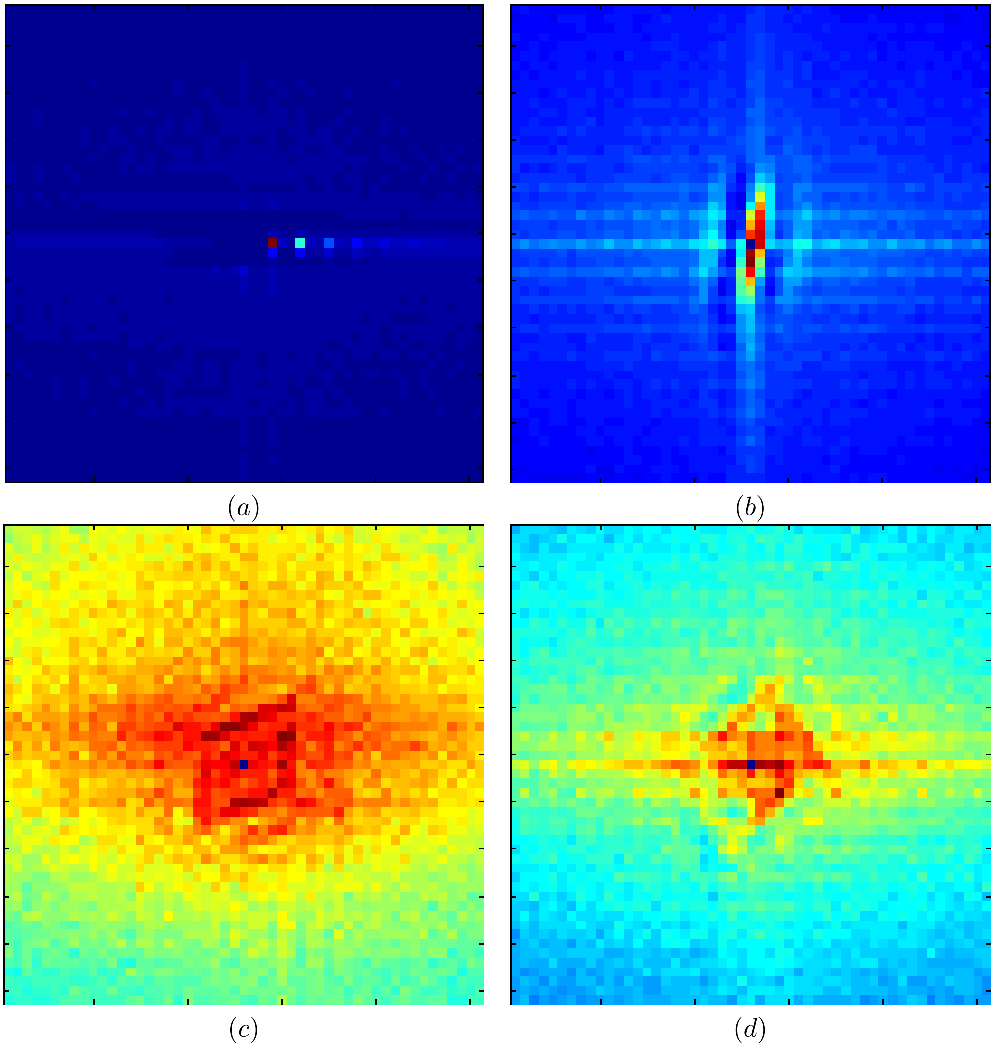


Figure 4: Sample spatial maps: (a) $\mathcal{S}_{1,1}$, (b) $\mathcal{S}_{5,4}$, (c) $\mathcal{S}_{5,2}$, (d) $\mathcal{S}_{6,3}$.

However, this benefit becomes a detriment when considering how to estimate a suitably tight variance around the peak. Since every point on the map lies in an attraction basin, the areas of most of the basins are far too large to just be directly taken as the area of variance around the peak. Actually calculating the variance around the peak of points within the cluster isn't much better; there are just far too many outliers exerting their influence on what should be a much more locally constrained estimation. As a result, clustering is not a successful solution to the variance problem, although the mean shift approach is very useful for peak finding on its own.

3.2 Non-maxima suppression

One run of the mean shift procedure on an entire map takes $O(kn^2)$ steps, where k is the number of iterations required for equilibrium and n is the number of locations in the map. While time was not a large consideration in deciding on a peak-finding algorithm, a much quicker $O(n)$ method proved to be more appropriate: a non-maxima suppression inspired algorithm. Although this technique is far less refined and yields many more spurious peaks than mean shift, a modification to Fidler and Leonardis' algorithm discussed further below makes the spurious peaks irrelevant. The details as to why this is the case can be found in Section 5.

Although "non-maxima suppression" has been referenced here, the term is used rather loosely. In reality, the algorithm suppresses the value of a 2D histogram at any point which:

1. has a neighbor touching it (out of the 8 surrounding pixels) with a larger value, and

2. has a nearby point (within an radius of r pixels) that has already been determined to be a max, where r is manually tuned per layer.

All of the non-suppressed points are then considered valid peaks. Condition 1 ensures that only local maxima make the cut, and condition 2 ensures that the points are far enough apart that they could be foci for separate patterns. For layer 1, r is set to 3. One important note is that the points in the histogram are processed in decreasing order of value. This ensures that higher values are not suppressed by lower values in condition 2.

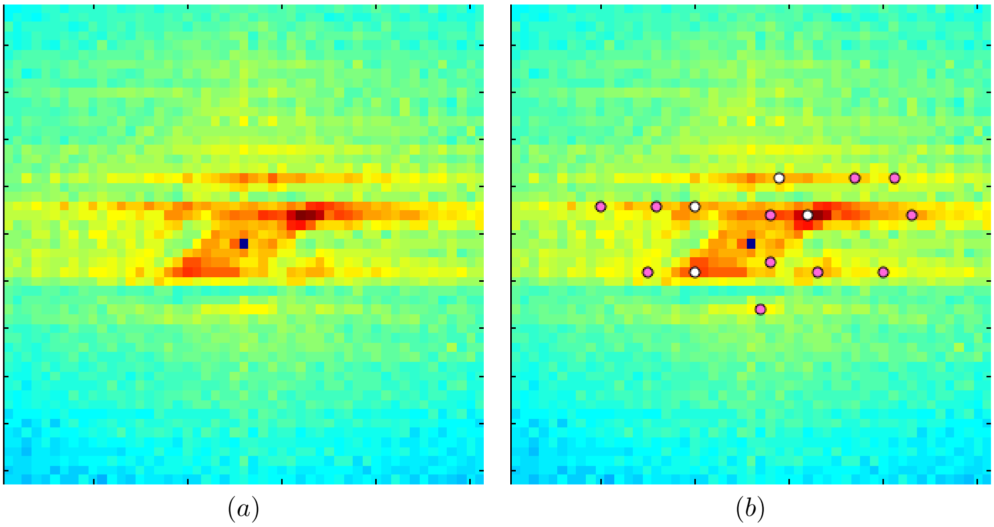


Figure 5: Peak finding from spatial map $\mathcal{S}_{1,3}$. The top 15 peaks are shown.

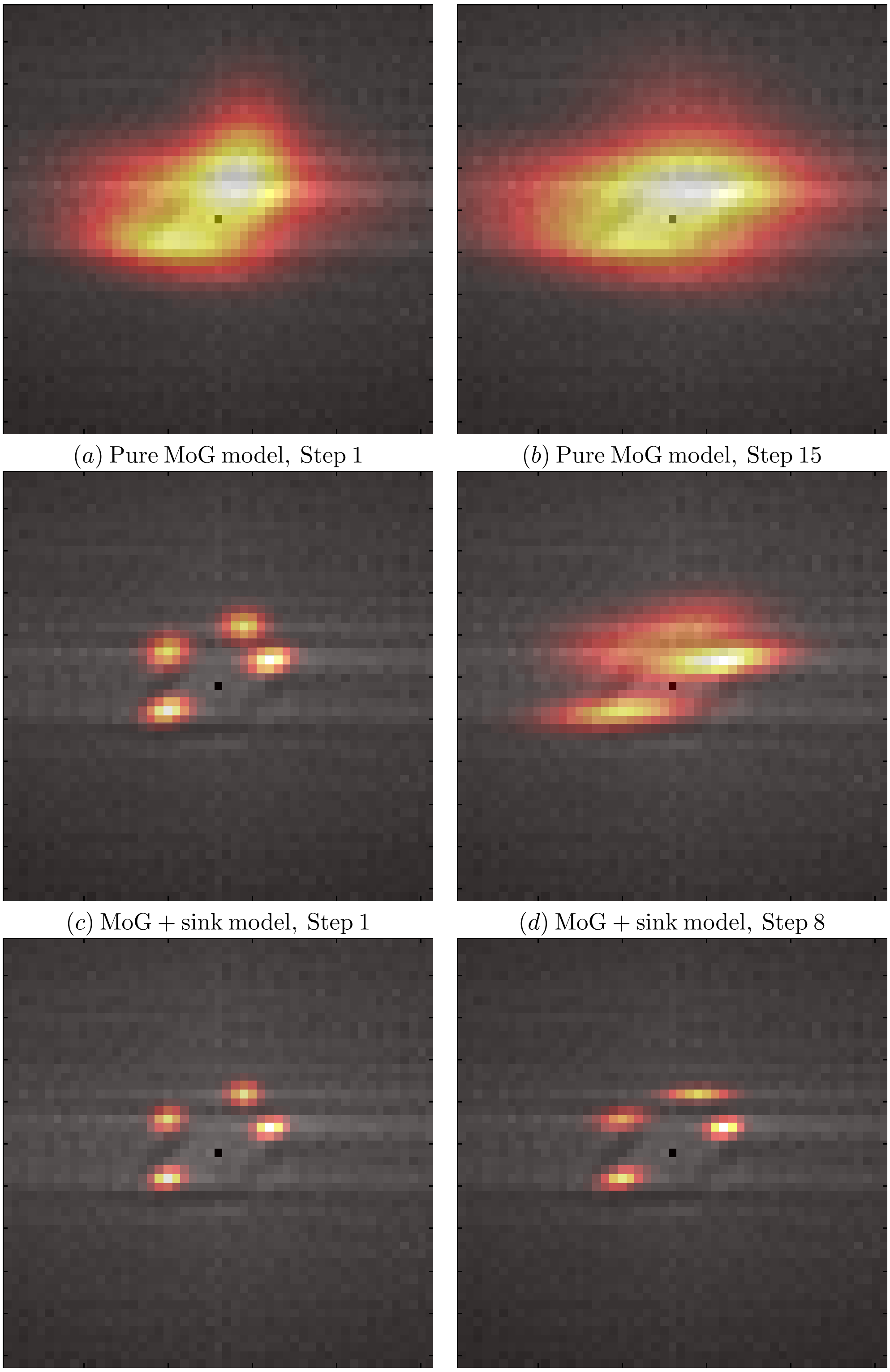
Figure 5 shows a spatial map along with it's 15 highest peaks extracted using this method. To slightly improve the performance of this technique, the spatial map can be filtered beforehand with any kind of averaging filter to smooth out a few of the bumps and spurious maxima. The number of spurious points, however, is immaterial due to the modification discussed further in Section 5. Before this enhancement is introduced however, the estimation of variance around the peaks will be discussed.

4 Variance estimation

In the previous discussion of peak finding, an approach involving parametric modeling was mentioned. While this technique proved to be inappropriate for finding the locations of peaks, finding a suitable area of variance around them is a very useful application of it. Intuitively, a spatial map should be a histogram that features true pattern locations (found at the peaks), with a significant height that dissipates out gradually (but quickly) from the true pattern location. This volume around the true pattern comes from many sources, including scale differences in input images, imperfections in noisy images, and just the natural flexible construction of objects in the world. This sparsely hilly data is naturally modeled as a mixture model of two-dimensional Gaussian distributions, one for each pattern laid out across the landscape. In addition to these hills, there is also a high level of baseline noise that comes from the massive amount of data that is not part of any appreciable pattern. Thus, the spatial maps can be modeled with a mixture of Gaussians sitting among a blanket of uniformly random noise that acts as a sort of "sink" to attract non-pattern data. This uniform assumption is not entirely true as will be discussed in Section 6, but it is sufficient for this discussion.

4.1 Mixture of Gaussian model

In a textbook mixture of Gaussian problem, both the mean and covariance must be found for each of the k distributions in the mixture. However, in this problem it can be assumed that the means have already been located through a peak finding procedure described in the previous section. Therefore, the means can be fixed



(a) Pure MoG model, Step 1

(b) Pure MoG model, Step 15

(c) MoG + sink model, Step 1

(d) MoG + sink model, Step 8

(e) MoG + sink + prior model, Step 1

(f) MoG + sink + prior model, Step 6

Figure 6: Variance estimation around four peaks in $\mathcal{S}_{1,3}$ found in Figure 5. Images show Gaussian estimates at different stages of the EM algorithm.

and only the covariances must be solved for in the parameter estimation algorithm. The algorithm used is Expectation Maximization (EM), which alternates between calculating expectations of hidden values in the problem (the "E step") and maximizing the model parameters based on these estimations for the hidden values (the "M step"). Since it is assumed that either the mixture of Gaussians or the uniform noise generates every data point in a spatial map, each location of that map is a "member" of either a Gaussian or the random noise;

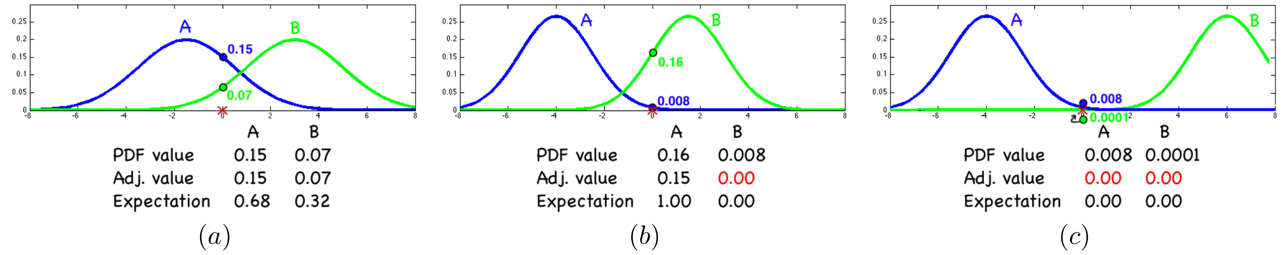
these membership values are the hidden variables estimated in the E step. The model parameters are of course the covariances of the Gaussians.

The first attempt at modeling the spatial map was to simply ignore the uniform noise underlying the whole map. That is, the hidden variables could only denote membership to one of the Gaussians, not membership to the bed of random noise. As Figures 6a and 6b shows, this simplification results in disastrous results. Without considering the noise's contributions to the spatial map, the EM algorithm attempts to account for every single location with a Gaussian. This is no different than the problem with getting variance out of the mean shift procedure. The resulting covariances bloat many of the Gaussians to sizes that are orders of magnitude larger than their intended scale. Obviously, the points that are not part of a pattern (which are the overwhelming majority of them) need to be handled in a different way.

4.2 Accounting for noise

Instead of trying to add another distribution into the mix in a principled way, similar results can be obtained with a simpler approach. The idea of the noise distribution is to explain away all of the data that shouldn't be accounted for by the tight Gaussian distributions. Therefore, when calculating the expected membership of a point, the expectation is set to 0 if the point lies safely outside the main volume of a particular Gaussian. To see how this change implicitly adds this "sink" behavior, see a one-dimensional analog in Figure 7.

The first frame shows a typical case where both Gaussians can make a significant claim to the datapoint in question, and the resulting membership values are split accordingly. In the second case, the point lies beyond some minimum density level (say, 95%) for A, and its membership is "revoked" for the distribution. In both of these cases, the random noise "sink" is not a factor because at least one Gaussian can explain the point. However, in the final example, 7c, the point does not yield a significant value from either Gaussian, so both expected membership values are clamped to zero. As a result, this data point will be irrelevant when the algorithm finds maximum likelihood estimates for parameters based on their membership variables in the M step. Since estimating a parameter for the random noise in unnecessary, keeping it's effect implicit in the algorithm is acceptable.



	A	B
PDF value	0.15	0.07
Adj. value	0.15	0.07
Expectation	0.68	0.32

(a)

	A	B
PDF value	0.16	0.008
Adj. value	0.15	0.00
Expectation	1.00	0.00

(b)

	A	B
PDF value	0.008	0.0001
Adj. value	0.00	0.00
Expectation	0.00	0.00

(c)

Figure 7: Fixing membership values at a datapoint marked by a red X in order to include a random noise sink. PDF stands for a Gaussian's probability distribution function. Note the red numbers where the value has been adjusted to 0.

While this modification improves performance slightly (Figures 6c and 6d), it still is not enough to constrain the covariance to an acceptable level. The Gaussians do not attempt to incorporate as much noise as in the previous frames, but they still bleed out too much. In spatial maps with a smaller density of peaks (e.g. 4a) this technique is sufficient, but for the messy maps more commonly seen, more inhibition is needed.

4.3 Bayesian priors

The spatial maps themselves do not have enough structure inherent in them to get the intended results through EM without a little intervention. Specifically, the addition of a Bayesian prior on the parameters in the Gaussian mixture is what is needed to derive this structure. These priors will impose a penalty term on Gaussians that want to expand too greatly. All part definitions within a certain layer should have similarly-sized areas of covariance, so while a prior may be more manual intervention than Fidler and Leonardis suggested, it does not add an unreasonable level of simplification to the problem. In particular, the priors are adapted from the discussion in [5], setting the hyperparameters prescribed within as follows:

- $\gamma_i = 1, \forall i \in \{1, 2, ..., k\}$. These parameters control a prior dealing with the relative strength of the Gaussians, which is not important for these purposes. Setting them all to 1 neutralizes this prior.
- $\eta = 0$. This controls a prior on the means of the Gaussians. Mean locations in this problem are fixed, so this prior is also set to a non-informative value.
- $\alpha = 1.5$. This parameter should scale relative to dimensionality of the data. [5] recommends setting this value to be $\frac{d+1}{2}$, so our two-dimensional data dictates fixing it to $\frac{3}{2}$.
- $\beta = 0.075I$, where I is the 2 x 2 identity matrix. This is the key parameter that inclines the covariance to be of a particular size. It's value was chosen manually through experimentation for layer 1. For different sets of spatial maps, a different value must be specified – an issue which will be discussed in the concluding remarks.

Using these parameters yields excellent results, as the final two frames of Figure 6 show. The priors do a great job of constraining the size of the Gaussians while still allowing considerable flexibility to the shape of the areas.

As the final image in Figure 6 shows, the mixture of Gaussian approach seems like the ideal approach to model the variances. However, for reasons that will be described in the next section, it will actually be more desirable to find the variance of a single pattern at a time. This modification changes nothing in the solution's framework. Focusing on one Gaussian at a time and setting $k = 1$ results in the desired behavior. At first it seems that the lack of interaction between Gaussians will change the results of the EM algorithm. However, since patterns in a spatial map should have minimal overlap, the Gaussians in the mixture model shouldn't be interacting in the first place. Therefore, estimating the variances all at once or one-by-one should yield very similar results.

5 Entropy-based selection

Previously, it has been mentioned that spurious peaks do not pose a problem due to a modification to the original algorithm proposed by Fidler and Leonardis. This change is described presently.

Although it was never explicitly stated, the authors seem to imply that a set number of patterns are gathered from each spatial map in their algorithm. If not a set value, at the very least it appears that each map was processed independently and separately to extract *some* number of patterns from each one. This approach ignores the fact that some spatial maps may be much more likely to produce patterns that others. Their algorithm had filters built into it farther down the pipeline that would disregard weak patterns at a later time. This results in wasted effort during the pattern matching stage and the possibility of skipping significant patterns from a quota system that treats all maps the same. A better solution would be to extract the patterns in order from strongest to weakest, regardless of which map they come from. To implement this strategy, the idea of entropy-based selection was developed.

Entropy-based selection operates on the assumption that significant patterns will come from spatial maps with low entropy. From information theory, entropy is a measurement of the uncertainty of a random variable. The higher the entropy, the more uncertainty is present in the variable (i.e. the more uniform the probability distribution is). More specifically, for a random variable X with possible values $\{x_1, x_2, ..., x_n\}$ entropy is defined as:

$$-k \sum_i \mathrm{P}(x_i) \ln \mathrm{P}(x_i)$$

where the constant can be ignored for our purposes.

We apply this concept to a spatial map to measure the uniformity of it's landscape. The random variable X covers any possible location on the spatial map, and $\mathrm{P}(x_i)$ is the probability of any datapoint being located at location i, or the ratio of the height at location i over the sum of all heights in the map. If a spatial map is a completely uniform surface, the entropy will reach it's maximum value for a map it's size. However, if the spatial map has some significant irregularities introduced, the entropy will begin to shrink. Maps with significant irregularities are exactly the ones that should be focused on: the more noticeably a peak stands out from the bed of noise, the more significant its pattern is likely to be. Therefore, by measuring the entropy, the best spatial maps to extract a pattern from can be visited in a strategic order. Instead, of extracting all patterns from a map at once, we can repeat these steps until a set number of definitions have been extracted:

```
until quota is reached:
    find the spatial map with the lowest entropy
    find the largest peak in the map (among the peaks found in Section 3)
    find the variance by modeling a Gaussian around this peak
    construct a new composite definition from this peak & variance
    flatten the area (described below)
    recompute the entropy of the current map
```

"Flatten the area" refers to a process where all of the points within the central 99% of the Gaussian's mass is driven down to some small value. Specifically, in this implementation, the value of each of these points is set to the mean value of the histogram. This way, future entropy calculations will not be significantly effected by the pattern already extracted, because its area will be fairly uniform with other parts of the map and contribute a higher level of entropy to the total.

By extracting the patterns one by one, the need for mixture of Gaussian modeling is removed, as mentioned in Section 4. In addition, as Section 3 promised, weak maxima that come from the simple peak finding algorithm are immaterial because specific maxima will only be visited if their spatial maps suggest that they are significant.

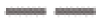
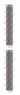
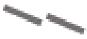
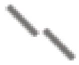
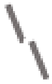
Rank	SpMap	Visual
1	1,1	
2	5,5	
3	8,8	
4	7,7	
5	6,6	

Figure 8: The five most significant patterns according to entropy-based selection

The patterns that this heuristic extracts in order make a good deal of intuitive sense. Figure 8 shows the first 5 patterns extracted from this technique, and they all seem like reasonable choices. Notice that they all include duplicate parts aligned in a fairly straight line, with their position relative to the subpart's angle. This commonality should not be too surprising: the layer 1 part definitions only describe 3-4 pixels, and images are full of straight lines at many different angles well over 4 pixels long.

Despite the success of these first few patterns, there are a few caveats with this selection procedure. It tends to focus a little *too* heavily on certain maps with extremely sharp peaks (e.g. $\mathcal{S}_{1,1}$ pictured in Figure 4a), even when the most significant ones have already been extracted. When variance is estimated for some of these more minor peaks, the area tends to explode (even despite the prior) because the peak is not strong enough. This expansion behavior is as of yet unexplained: it could be some singularity in the data at these points or simply a implementation bug. Of more concern is the fact that choosing these peaks appears to reveal a fault in entropy-based selection. No further modifications have been attempted presently, but one idea would be to add a penalty term to spatial maps whose more salient peaks have no appreciable mass surrounding them. Regardless, entropy-based selection is a solid first step in developing a heuristic to better direct the pattern finding procedure.

6 Unresolved Issues and Conclusion

In the end, creating a fully automatic process to extract the patterns necessary to create composite part definitions was not entirely successful. Although this attempt came close to achieving the goals for a specific subproblem, there is no guaranteeing this procedure would generalize seamlessly to spatial maps with more subparts or at higher layers. Some potential problems include the following:

- **Compositions as central parts**: In this discussion, all spatial maps have been simple pairings of 2 subparts. Since many composite objects are best described with more than 2 subparts, Fidler and Leonardis allow for the building of composite definitions encompassing up to 5 subparts. The spatial maps for these m-subpart patterns are built inductively, by using the already established (m-1)-subpart compositions as the central part. Since none of these maps have been generated yet, it cannot be assumed that the previously established techniques will be as successful on them.

- **Ranking all subparts in a layer**: A method to rank the significance of all extracted patterns must ultimately be used to select which compositions should constitute the next layer. Ranking compositions with the same number of subparts by frequency is simple, but comparing ones with different numbers of subparts is more difficult due to the inductive nature of how they're defined. A way to approach this is something that was not suitably addressed in [3].

- **Gaussian mixture priors**: It is fairly safe to assume that all patterns at the same layer with the same number of subparts should have similarly sized variances. However, it's unclear how simple it would be to specify (or even better, generate) a prior for Gaussians in situations beyond the first layer described here. Since composite parts can have up to 5 subparts, and their hierarchies reach up to 5 layers, at least 20 different priors would have to be determined to complete the algorithm.

- **Radius for peak inhibition**: On a related note, the "non-maxima suppression" technique discussed in Section 3 needs a radius to ensure that only one point within a certain neighborhood is granted peak status. This is admittedly a small consideration, but it is another example of a parameter that must scale for different layers at a rate that is not entirely clear.

- **Uniform levels of noise**: The assumption made in Section 4 about there being a uniform source of noise underlying the entire spatial map is not entirely true. For spatial maps of such a small scale where only peaks close to the center are considered, the uniform assumption works fine. However, as the borders of many spatial maps show (e.g. Figure 4d), a more curved landscape often begins to present itself as the spatial map expands. I can envision this tendency becoming more pronounced in higher layers, putting the uniform assumption in jeopardy.

In Fidler and Leonardis' paper, the processing of patterns in spatial maps is nearly overlooked. As their publication states, "We detect voting peaks in the learned spatial maps, and for each peak, a spatial area that captures most of the votes is formed." As discussed at length here, this task is far more complicated than a simple sentence can possibly describe. Their hierarchical scheme for object recognition is very intuitive and the framework is general enough to make it applicable in many situations. According to the results published, they were able to find a good amount of success with their approach, automatically building a hierarchical blueprint of many objects that matches well with human intuitions. However, the actual pattern extraction part of this algorithm is anything but intuitive, and warrants a much fuller explanation by the authors. Attempts to implement the process have come close to emulating the intended behavior, but this methodology is not as simple or robust as the original publication suggests it should be.

References

[1] Dorin Comaniciu, Visvanathan Ramesh, and Peter Meer. The variable bandwidth mean shift and data-driven scale selection. In *ICCV '01: Proceedings of the Seventh IEEE International Conference on Computer Vision*, volume 1, page 438, Los Alamitos, CA, USA, 2001.

[2] Sanja Fidler, Gregor Berginc, and Ales Leonardis. Hierarchical statistical learning of generic parts of object structure. *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 1:182–189, 2006.

[3] Sanja Fidler and Ales Leonardis. Towards scalable representations of object categories: Learning a hierarchy of parts. *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 0:1–8, 2007.

[4] Bogdan Georgescu, Ilan Shimshoni, and Peter Meer. Mean shift based clustering in high dimensions: A texture classification example. In *ICCV '03: Proceedings of the Ninth IEEE International Conference on Computer Vision*, page 456, Washington, DC, USA, 2003.

[5] Dirk Ormoneit and Volker Tresp. Improved gaussian mixture density estimates using bayesian penalty terms and network averaging. In *NIPS '95: Proceedings of the Eighth Neural Information Processing Systems Conference*, pages 542–548, 1995.

[6] Nikolay Petkov. Biologically motivated computationally intensive approaches to image pattern recognition. *Future Generation Computer Systems*, 11(4-5):451–465, 1995.