Lecture 1. Spectral Clustering. The input is a set of elements with affinities specified between them. The output is a decomposition of the elements into subsets of elements, so that elements in the same subset have big affinities to each other and small affinities to elements in other subsets. This is formulated by defining a graph where each node corresponds to a vertex and the weights on the graph edges between different nodes are given by the affinities.For example, this can be applied to image segmentation where the elements are the image pixels and the affinity is a measure of the similarity between different pixels (i.e. nearby pixel with similar intensities receive high weights, while pixels which are spatially separated and have different intensities will have low weights). To obtain these subsets, we define a Graph Laplacian. The subsets can then be found from the eigenvectors and eigenvalues of matrices constructed from the graph Laplacian. In computer vision, spectral clustering is often used to decompose the image into superpixels (within which the intensity changes slowly). These can be used for later processing. See `Superpixels.pdf’.

Lecture 2. Region Competition. We formulate a probability distribution which models an image as a set of non-overlapping regions where each region is generated by a different probability distribution on the image intensity. This is more advanced than the weak smoothness model because it allows many possible probability distributions to be used (e.g., for texture). The region Competition paper (Zhu and Yuille) used a limited class of models – and the model types were fixed for each image. The algorithm proceeded by estimating the positions of the boundaries of the regions and the parameters of the models describing the models alternatively. Estimating the positions of the boundaries was performed by steepest descent which led to regions competing for ownership of pixels on the boundaries, hence the name “region competition”. The algorithm was initialized started with multiple image seeds (i.e. over-segmentation) which could merged later. A more sophisticated approach was developed by Tu and Zhu (2002) which included multiple models and sophisticated Markov Chain Monte Carlo inference algorithm. See `region\_competition\_pami.pdf’.

Lecture 3. Lighting Models. The Lambertian lighting model specifies the image in terms of the geometry of the viewed object, its albedo, and the light source directions. The set of images of an object are a three-dimensional linear space if the object is Lambertian and if we ignore shadows. We can test this assumption by taking photographs of an object under different lighting conditions and performing principal component analysis (PCA) to determine the dimenisionality of the images. Experiments show that the image of many objects can be expressed in a low-dimensional space, which give support to the Lambertian assumption. If we assume the Lambertian lighting model we can attempt to estimate the object shape, albedo, and the light source directions. This can be formulated in terms of minimizing an energy function. This can be solved for by Singular Value Decomposition up to the Generalized Bas Relief (GBR) ambiguity, which is inherent to the Lambertian model. The GBR ambiguity means that we can only the object shape up to a three-dimensional transformation, unless we know the directions of the light sources. This is an extension of the well-known ambiguity that humans cannot tell the difference between convex objects lit from below or concave objects lit from above. Humans have a tendency to perceive objects as convex – for example if you are shown an inverted face mask you will probably perceive it as convex (i.e. like a normal face). See `GBR.pdf’ and `svd\_eccvwork96.pdf’.

Lecture 4. Active Appearance Models (AAMs) and FORMs. These models represent the appearance of an object as a linear weighted combination of eigenimages, which are obtained by PCA . This can be used, for example, to represent faces in terms of a limited number of eigenfaces. Next we can introduce spatial warps which allows the models to have linear transformations which can also be estimated from examples of each object by PCA if we know the correspondence between pixels from different examples of the object. If we do not know the correspondence, then we can use the EM algorithm. AAMs are good for representing objects that undergo limited types of deformation (e.g., faces). They are not good for objects like cows where different parts of the objects can move with respect to each other. FORMS is an example (one of many) of a system which represents an object in terms of a dictionary of elementary parts (whose shape defomations are modeled by PCA) which are joined together to form an object. To detect these parts, and the joints between them, we can detect the medial axes and discover where the axis split. See `AAM’s.pdf’ and `FORMS.pdf’.

Lecture 5. Deformable Templates. These are models objects by MRFs where the nodes represent the positions (and orientations) of parts of the object and the edges specify the spatial relationship between them. The unary terms of the MRF indicate how the parts of the object interact with the image (e.g., they may have high potential values at places in the image where there are edges) and the binary terms represent the variability of the relative positions of parts. If the graph does not have closed loops, then dynamic programming can be used to find the optimal configuration of the object – i.e. to detect its position in the image. If the graph has closed loops then other algorithms like belief propagation can be used. Similar techniques can be applied to the shape matching problem, where the object is matched to another object and not to an image. We can use features like shape context (which takes into account the local structure of the object) to make the matching less ambiguous.

Lecture 6. RCMs are hierarchical graphical models. These models are defined on hierarchical graphs. They represent structures, objects and images, in terms of compositions of elementary elements. The hierarchical structure enables us to represent context information at a range of scales enabling long-range interactions useful for labeling images and detecting objects. Inference can be performed on these models using dynamic programming adapted to closed loops (junction trees). The models can be learnt by structured max-margin methods (see Lecture6\_Day5\_Structure). They can be applied to object detection and image labeling. For some examples, for the Pascal Challenge, latent SVM’s (a machine learning approximation to the EM algorithm) is used because groundtruth is specified only for the presence or absence of an object, and not for the positions of the object parts. See `RCMs\_Lecture6\_Day5.pdf’.