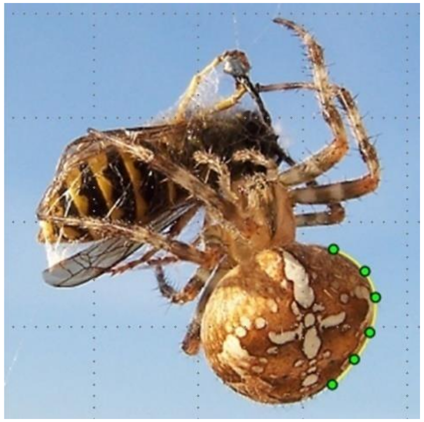


COMP 9517 Computer Vision

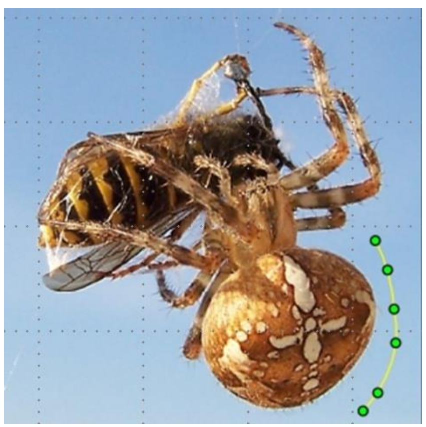
Segmentation and Feature Tracking

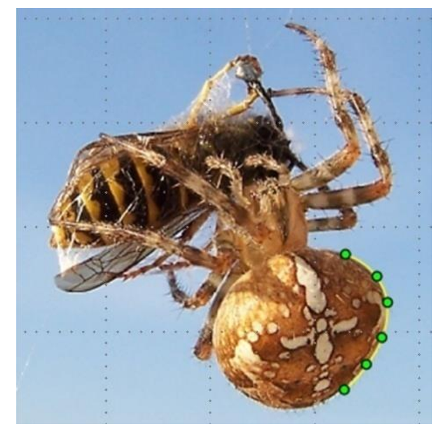
Curves and Segmentation

- curves corresponding to object boundaries are common, especially in natural environment
- one approach to segmentation is to locate object boundary curves in images

Active Contours

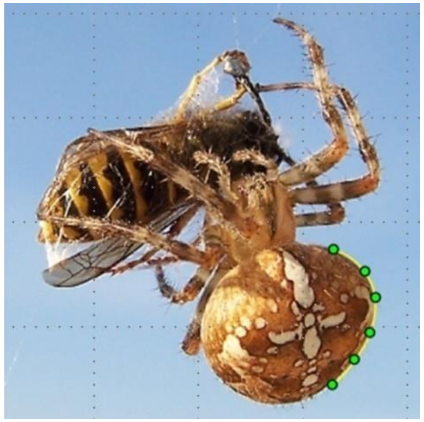
- Aim: to locate boundary curves in images
- How: boundary detectors iteratively move towards their final solution under a combination of *image , smoothness and optional user-guidance forces*

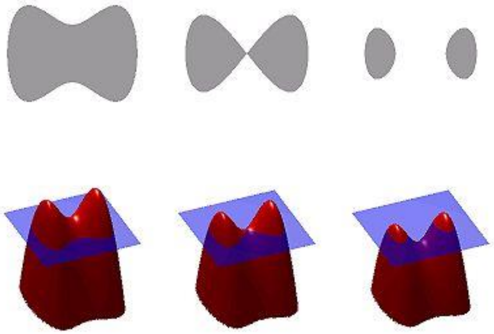


Active Contours

- Examples of implementations:
  - Snakes
  - Level sets





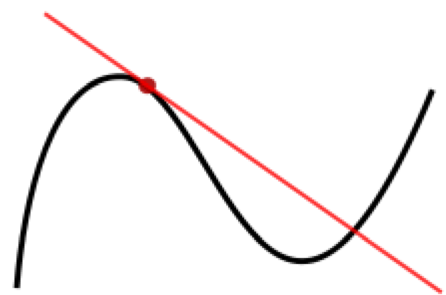
- Active contours can also be used in a wide variety of object-tracking applications

Global Optimization Techniques

- formulate the goals of the desired transformation (eg segmentation) using some optimisation criterion, then infer the solution that best meets this criterion
- Problem: finding a smooth surface that passes through a set of measured data points
  - ill-posed: many possible surfaces can fit this data
  - ill-conditioned: small changes in input can sometimes lead to large changes in the fit
  - inverse problems: recover the unknown function from which data points were sampled
- therefore, regularisation is needed
  - to fit models to data that badly underconstrain the solution space

Global Optimisation

- in order to quantify what it means to find a smooth solution, we can define a norm on the solution space
- the **derivative** is a measure of how a function changes as its input changes.



  - a derivative can be thought of as how much one quantity is changing in response to changes in some other quantity
  - for example, the derivative of the position of a moving object with respect to time is the object's instantaneous velocity

$$m = \frac{\Delta f(x)}{\Delta x} = \frac{f(x+h) - f(x)}{(x+h) - (x)} = \frac{f(x+h) - f(x)}{h}.$$

Global Optimisation

- For one dimensional function f(x), to find a smooth solution, we can define a norm on the solution space:
  - integrate the squared first derivative of the function

$$\varepsilon_1 = \int f_x^2(x)dx$$

  - integrate the squared second derivative

$$\varepsilon_2 = \int f_{xx}^2(x)dx$$

- For two dimensions (e.g. images), the corresponding smoothness functions are partial derivatives:

$$\varepsilon_1 = \int f_x^2(x,y) + f_y^2(x,y)dxdy = \int \left\|\nabla f(x,y)\right\|^2 dxdy$$

$$\varepsilon_2 = \int f_{xx}^2(x,y) + 2f_{xy}^2(x,y) + f_{yy}^2(x,y)dxdy$$

Global Optimization

- We also require a data term (or data penalty)
- For scattered data interpolation, the data term measures distance between function f (x,y) and a set of data points d_i = d (x_i, y_i)
- By adding the norm and the data term, we get a global energy term that can be minimized

Snakes

- Snakes are a 2-D generalisation of the 1-D energy minimising splines:

$$\varepsilon_{\text{int}} = \int \alpha(s) \left\| f_s(s) \right\|^2 + \beta(s) \left\| f_{ss}(s) \right\|^2 ds$$

- discretised version of the energy function:

$$E_{\text{int}} = \int \alpha(s) \left\| f(i+1) - f(i) \right\|^2 / h^2 + \beta(s) \left\| f(i+1) - 2f(i) + f(i-1) \right\|^2 / h^4$$

- additional external image-based and constraint-based potentials:

$$\varepsilon_{image} = \omega_{line}\varepsilon_{line} + \omega_{edge}\varepsilon_{edge} + \omega_{term}\varepsilon_{term}$$

Snakes

- In practice, only the edge term is used
  - directly proportional to the image gradients:

$$E_{edge} = \sum_i -\left\| \nabla I(f(i)) \right\|^2$$

  - or a smoothed version of the image Laplacian:

$$E_{edge} = \sum_i -\left\| (G_\delta \circ \nabla^2 I)(f(i)) \right\|^2$$

  - distance map to the edges
- user-placed constraints, for e.g.

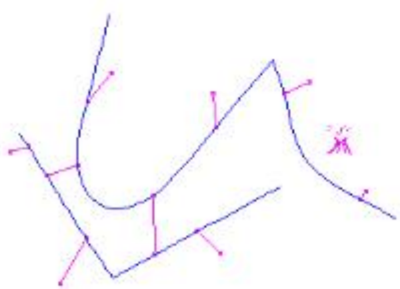
$$E_{spring} = k_i \left\| f(i) - d(i) \right\|^2$$

Snakes

- Put them together:

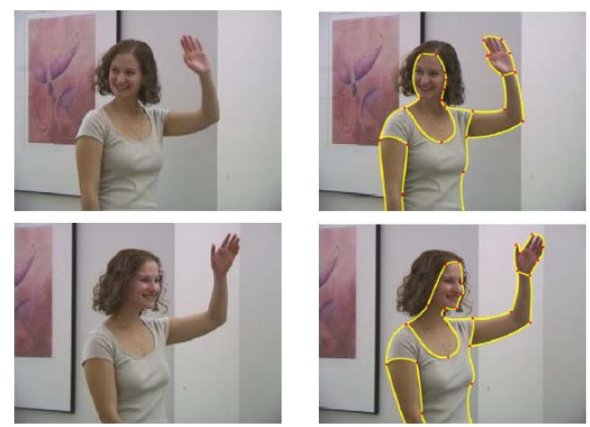
$$E = E_{\text{int}} + E_{image} + E_{spring}$$

- To minimize E, let E be zero, then energy E can be iteratively minimised by moving the curve



(a)

(b)



Dynamic Snakes

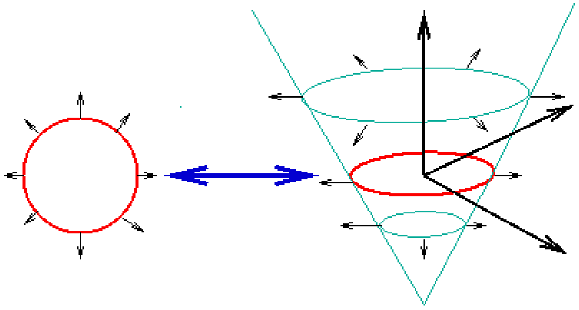
- When the object of interest is being tracked from frame to frame, prediction of new estimates is needed
- How? Using dynamic models:
  - Kalman Filtering
  - Particle Filtering

These will be covered when studying Motion

Level Set Methods

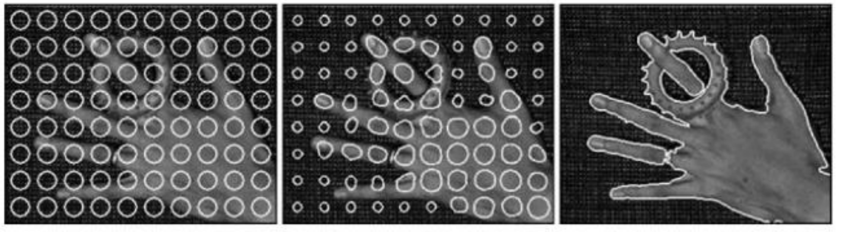
- Limitations of active contours based on parametric curves:
  - challenging to change the topology of the curve as it evolves
  - curve reparameterisation may be required when the shape changes dramatically
- Use the zero crossing of a signed distance function is used to define the curve:

$$\frac{\partial \varphi}{\partial t} = v|\nabla \varphi|.$$

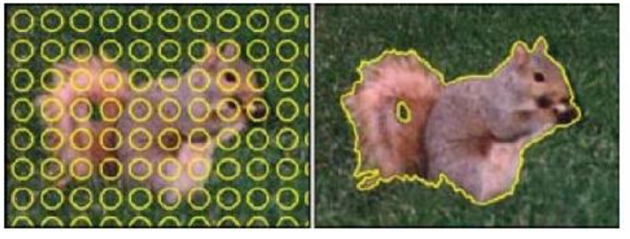


Level Set Methods

- Recent approaches recast the problem in a segmentation framework, where the energy measures the consistency of the image statistics inside and outside the segmented regions

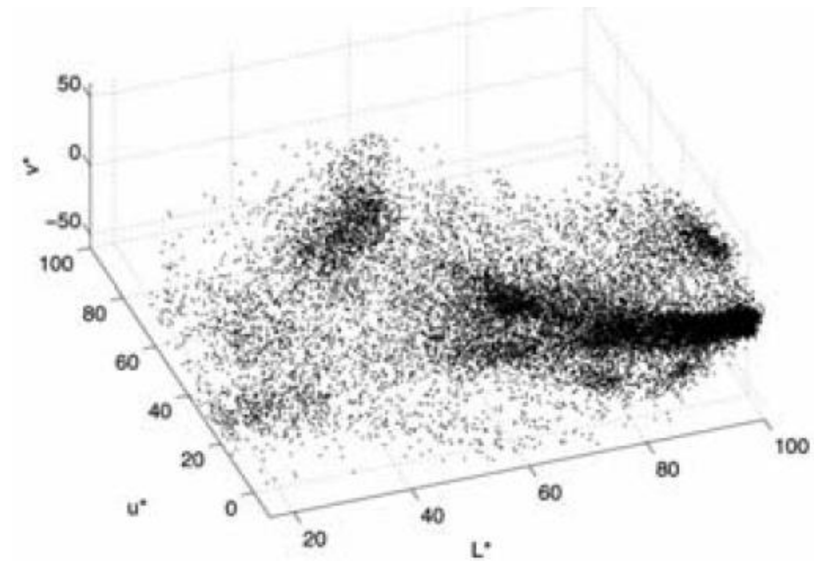


(a)



Mean Shift and Model Finding

- How would you segment this image based on colour alone?



Mean Shift and Model Finding

- K-means and mixtures of Gaussians technique uses a parametric model of the density function to answer this question
  - assume the density is the superposition of a small number of simple distributions (e.g., Gaussians) whose locations (centres) and shape (covariance) can be estimated
- Mean shift smoothes the distribution and finds its peaks, as well as the regions of feature space that correspond to each peak (non-parametric)

K-mean Clustering

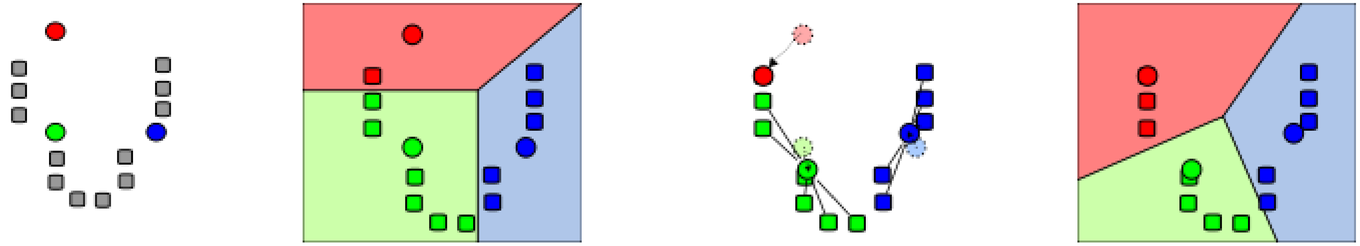
- K-means
  - models the probability density as a superposition of spherically symmetric distributions
  - assumes we are given the number of clusters k
  - randomly initialises sampling centres
  - iteratively updates the cluster centre location based on the samples that are closest to each centre
  - techniques exist for splitting or merging cluster centres to accelerate the process

K-means Clustering

- *Iterative K-means clustering*
  - set iteration count *ic* to 1
  - randomly choose a set of K means $m_1(1)$, $m_2(1),\ldots,m_K(1)$
  - for each vector x_i compute distance $D(x_i, m_k(ic))$ for each $k = 1,\ldots,K$ and assign x_i to the cluster C_j with the nearest mean
  - increment *ic* by 1 and update the means to get a new set $m_1(ic)$, $m_2(ic),\ldots,m_K(ic)$
  - Repeat step 3 and 4 until $C_k(ic) = C_k(ic+1)$ for all k

K-means Clustering

- *Iterative K-means clustering*



K-means and Mixtures of Gaussians

- beside means, each cluster centre is augmented by a covariance matrix re-estimated from the corresponding samples
- instead of using nearest neighbours to associate input samples with cluster centres, a Mahalanobis distance is used:

$$d(x_i, \mu_k; \Sigma_k) = \|x_i - \mu_i\|_{\Sigma_k^{-1}} = (x_i - \mu_k)^T \Sigma_k^{-1} (x_i - \mu_k)$$

- samples can be associated with the nearest cluster centre, in *hard assignment* of membership or, *softly assigned* to several nearby clusters

– corresponds to iteratively re-estimating the parameters for a mixture of Gaussians density function:

$$p(x \mid \{x_i, \mu_k, \Sigma_k\}) = \sum_k \pi_k N(x \mid \mu_k, \Sigma_k)$$

$$N(x \mid \mu_k, \Sigma_k) = \frac{1}{|\Sigma_k|} e^{-d(x_i, \mu_k; \Sigma_k)}$$

K-means and Mixture of Gaussians

Expectation Maximisation (EM) can be used to iteratively compute maximum likely estimate for the unknown mixture parameters

The expectation stage (E step) estimates the responsibilities:

$$z_{ik} = \frac{1}{Z_i} \pi_k N(x_i \mid \mu_k, \Sigma_k) \qquad \sum_k z_{ik} = 1$$

How likely a sample was generated from cluster k

The maximisation stage (M step) updates the parameter values

$$\mu_k = \frac{1}{N_k} \sum_i z_{ik} x_i$$

$$\Sigma_k = \frac{1}{N_k} \sum_i z_{ik} (x_i - \mu_k)(x_i - \mu_k)^T$$

$$\pi_k = \frac{N_k}{N}$$

$$N_k = \sum_i z_{ik}$$ No of samples in cluster k

Mean Shift

- procedure for locating the maxima of a density function given *discrete data* sampled from that function
- uses a smooth continuous non-parametric model to model the probability density function being segmented
- efficiently finds peaks in this high-dimensional data distribution, without ever computing the complete function explicitly

Mean Shift

- Smooth the data by convolving with a fixed kernel function (kernel density estimation)

$$f(x) = \sum_i K(x - x_i) = \sum_i k\left(\frac{\|x - x_i\|^2}{h^2}\right)$$

– Calculate the gradient:

$$\nabla f(x) = \sum_i (x_i - x) g\left(\frac{\|x - x_i\|^2}{h^2}\right) = \sum_i (x_i - x) G(x - x_i), g(r) = -k'(r) \Rightarrow$$

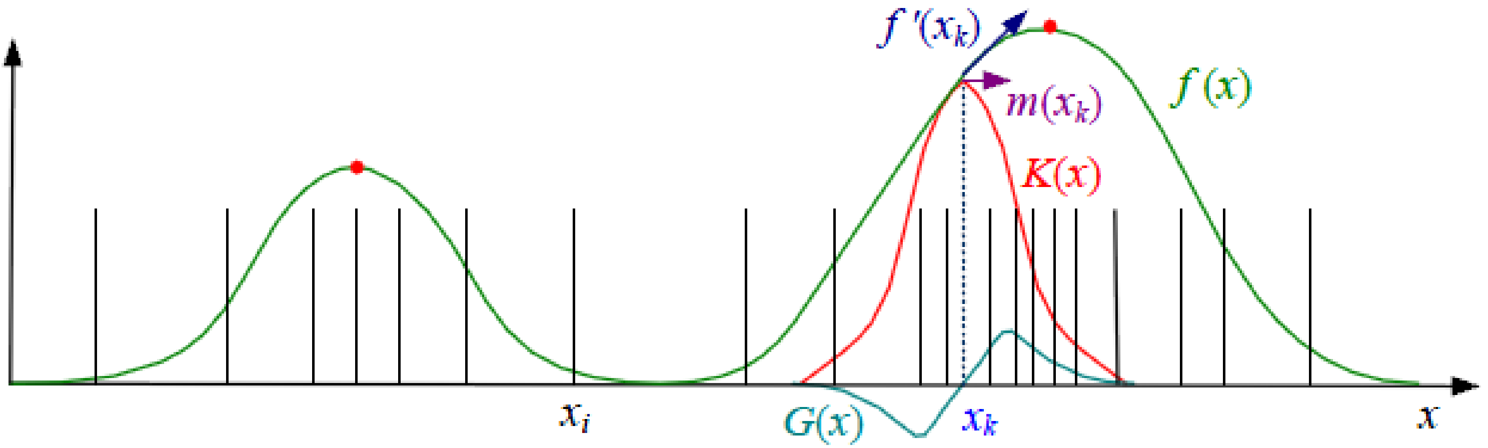
$$\nabla f(x) = \left[\sum_i G(x - x_i)\right] m(x), G = DeviativeKernel$$

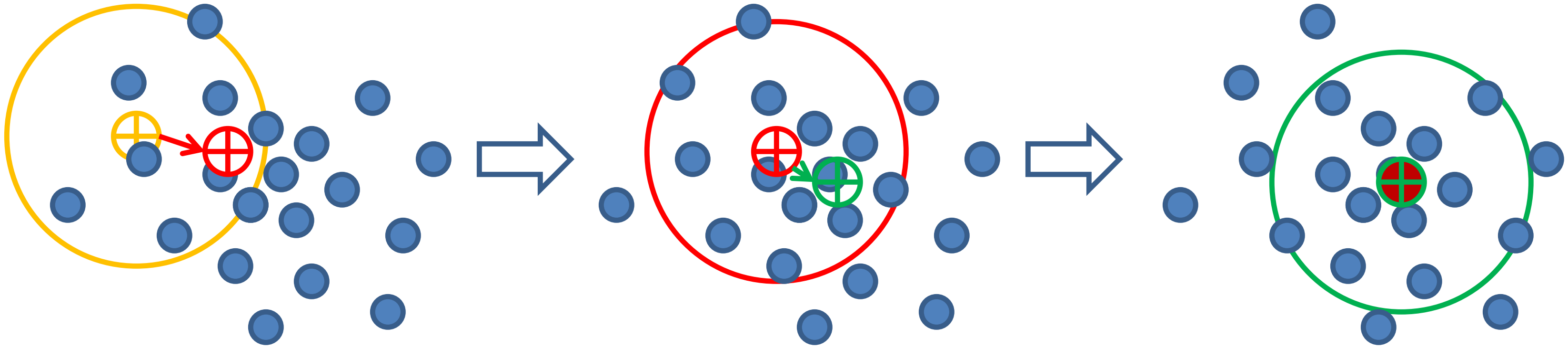
where the vector $m(x) = \frac{\sum_i x_i G(x - x_i)}{\sum_i G(x - x_i)} - x$ is called ***mean shift***, which is the difference between the weighted mean of the neighbours x_i around x and the current value of x

– The current estimate $y_{k+1} = y_k + m(y_k) = \frac{\sum_i x_i G(y_k - x_i)}{\sum_i G(y_k - x_i)}$

Mean Shift

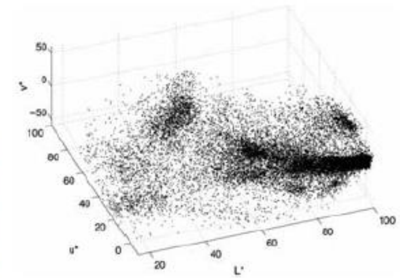
- 1-D visualization of kernel density estimate f(x), kernel K, its derivative G, and a mean shift m, weighted nearby points for re-estimation of the mean, red dots are local maxima to which mean shifts converge to



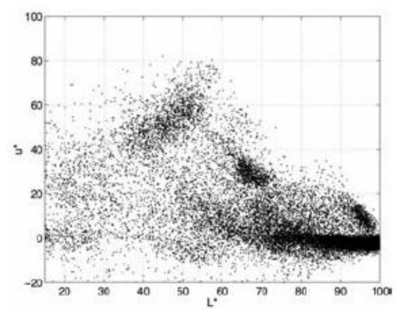


Mean Shift

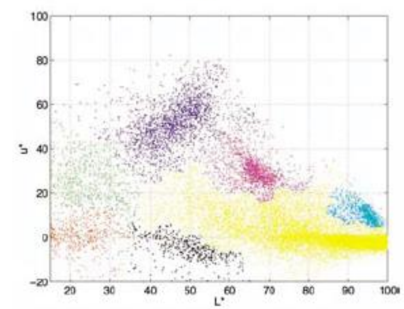
(a)



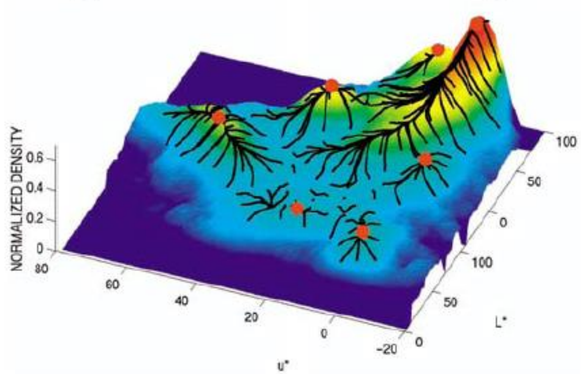
(b)



(c)



(d)



(e)

Mean Shift

- Mean Shift for Visual Tracking
  - mean shift algorithm can be used for visual tracking
  - create a confidence map in the new image based on colour histogram of the object in the previous image
  - tracking target object in sequence by matching colour density
  - use mean shift to find the peak of a confidence map near the object's old position

References and Acknowledgements

- Chapter 3, 5 Szeliski 2010
- Some images drawn from Szeliski 2010 and web
- Other references:

  - T.F. Cootes and C.J. Taylor and D.H. Cooper and J. Graham. "Active shape models - their training and application". *CVIU, 1995*.
  - D. Comaniciu, V. Ramesh, and P. Meer. Real-time tracking of non-rigid objects using mean shift. CVPR, 2000