CSE398/498-011
Advanced Topics in Mobile Robotics

Stanley Road Classification Overview
13 Feb 07

Group Updates

- General (Doug/Jason):
  - *Status of PW protected Wiki*
- LIDAR Group (Jason, Doug):
  - *RS422 Adapters functional w/driver?*
  - *Benchmark these (scans/second). TBC:*
  - *Changing the PD sensitivity. TBC:*
- Vision Group I (Chris)
  - *GUI Update*
  - *Chao et al have sample project to work from?*
- Vision Group II (Chao, Thomas, Mark)
  - *Focus on segmenting the road from the Goodman site video*

Group Updates

- GPS (Charles)
  - *Status?*
  - *Suspense?*
- Vehicle Group (Jason, Anthony, Matt)
  - *Mount status*
  - *Integrate Bumblebee 2*
  - *Interfaced w/car*
- Test Group (Jason, Anthony, Matt)
  - *Need test procedures list for all to use posted on the Wiki*
  - *Want additional data collected when practicable*
  - *Lane markings?*

Stanley Road Classification

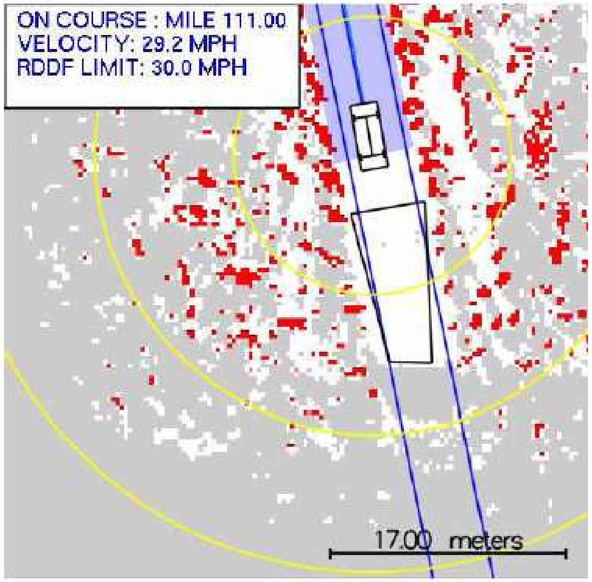
- "Self-supervised Monocular Road Detection in Desert Terrain," Dahlkamp et al, *RSS* 2006

The Process

1. Extracting close range road location from sensors invariant to lightning conditions
2. Removing sky and shadow areas from the visual field
3. Learning a visual model of the nearby road
4. Scoring the visual field by that model
5. Selecting identified road patches
6. Constructing a sky-view drivability map

Extracting Close Range Road

- Trust LIDAR
- Scan the near-field
- Reference these to the vehicle via a 6 DOF pose estimation system
- Map these measurements to the camera FOV to identify the "drivable" surface
- The resulting quadrangle is the "training area" for the vision system



Removing Sky and Shadows

- Shadows are simply removed from the road using a threshold on image intensity and with blue content greater than the red and green

$$Shadow = \mathrm{Im}_Y < Y_{\min} \;\&\; \mathrm{Im}_B > \mathrm{Im}_R \;\&\; \mathrm{Im}_B > \mathrm{Im}_G$$

- Sky identified using horizon finding algorithm (Ettinger et al)
- Used other image processing techniques (e.g. flood fill, morphological operations) to clean this up

Learning a Visual Model

- Mixture of Gaussians (MOG) model in RGB space with *k* Gaussians
- Classify pixels via k-means clustering, and then model each by its mass, mean, and covariance
- There are also *n* “learned” models (*n*>*k*)
- If the new image data match a model such that

$$(\mu_L - \mu_T)^T (\Sigma_L + \Sigma_T)^{-1} (\mu_L - \mu_T) \leq 1$$

  then it is assimilated to the new model
- Otherwise, the lowest mass learned model is replaced by the new terrain model

Background: First & Second Order Statistics/Moments: Expected Value & Variance of the State

- The *expected value* for a random variable X is (*i.e.* the mean) defined as

$$\mu = E(X) = \sum_{i=1}^{n} p_i x_i \quad \text{for discrete } X$$

$$\mu = E(X) = \int_{-\infty}^{\infty} x p_X(x) dx \quad \text{for continuous } X$$

- The *variance* of X about the mean is defined as

$$\sigma^2 = E[(x-\mu)^2] = \sum_{i=1}^{n} p_i (x_i - \mu)^2 \quad \text{for discrete } X$$

$$\sigma^2 = E[(x-\mu)^2] = \int_{-\infty}^{\infty} (x-\mu)^2 p_X(x) dx \quad \text{for continuous } X$$

What is Covariance?

- When X is a vector, the variance is expressed in terms of a *covariance matrix* **C** where

$$c_{ij} = E[(\vec{x}_i - \vec{\mu}_i)^T (\vec{x}_j - \vec{\mu}_j)]$$

- The resulting matrix has the form

$$C = \begin{bmatrix} \sigma_1^2 & \rho_{12}\sigma_1\sigma_2 & \cdots & \rho_{1n}\sigma_1\sigma_n \\ \rho_{12}\sigma_1\sigma_2 & \sigma_2^2 & \cdots & \rho_{2n}\sigma_2\sigma_n \\ \vdots & \vdots & \ddots & \vdots \\ \rho_{1n}\sigma_1\sigma_n & \rho_{2n}\sigma_2\sigma_n & \cdots & \sigma_n^2 \end{bmatrix}$$

where ρ_{ij} corresponds to the degree of correlation between variables X_i and X_j

Properties of the Covariance Matrix

- The *covariance matrix* ***C*** is symmetric and positive definite
- Through a *similarity transform* with the proper rotation matrix ***R***, ***C*** can be decomposed as ***C=RDR^T***, where

$$C = \begin{bmatrix} EVE_1 & \cdots & EVE_n \end{bmatrix} \begin{bmatrix} \sigma_1^2 & 0 & 0 \\ 0 & \ddots & 0 \\ 0 & 0 & \sigma_n^2 \end{bmatrix} \begin{bmatrix} EVE_1 \\ \vdots \\ EVE_n \end{bmatrix}$$

- That is, the columns of ***R*** correspond to the eigenvectors of ***C***, and the elements of the diagonal matrix ***D*** its eigenvalues
- These also correspond to the primary axes of the PDF and the variances, respectively

Example

- QUESTION: Given the covariance matrix

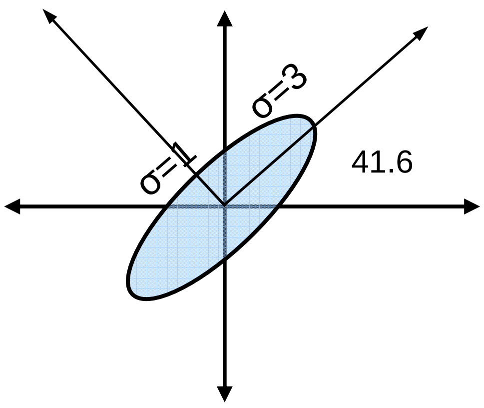
$$C = \begin{bmatrix} 5.4729 & 3.9719 \\ 3.9719 & 4.5271 \end{bmatrix}$$

characterize the PDF

- SOLUTION: Solving for the eigenvalues, we get

$$(\lambda - 5.47)(\lambda - 4.53) - 3.97^2 = 0$$

$$\Rightarrow \lambda_1 = 9, \lambda_2 = 1 \Rightarrow \sigma_1^2 = 9, \sigma_2^2 = 1$$

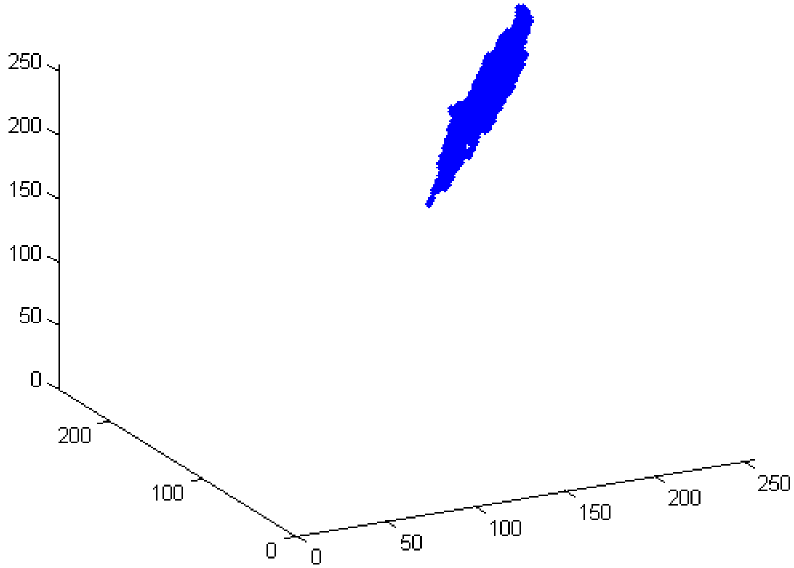


and solving for the first EVE we obtain

$$x_1 = \begin{bmatrix} 0.747 \\ 0.664 \end{bmatrix} \Rightarrow \tan^{-1}\left(\frac{0.664}{0.747}\right) = 41.6^\circ$$

Learning a Visual Model

$$(\mu_L - \mu_T)^T (\Sigma_L + \Sigma_T)^{-1} (\mu_L - \mu_T) \leq 1$$



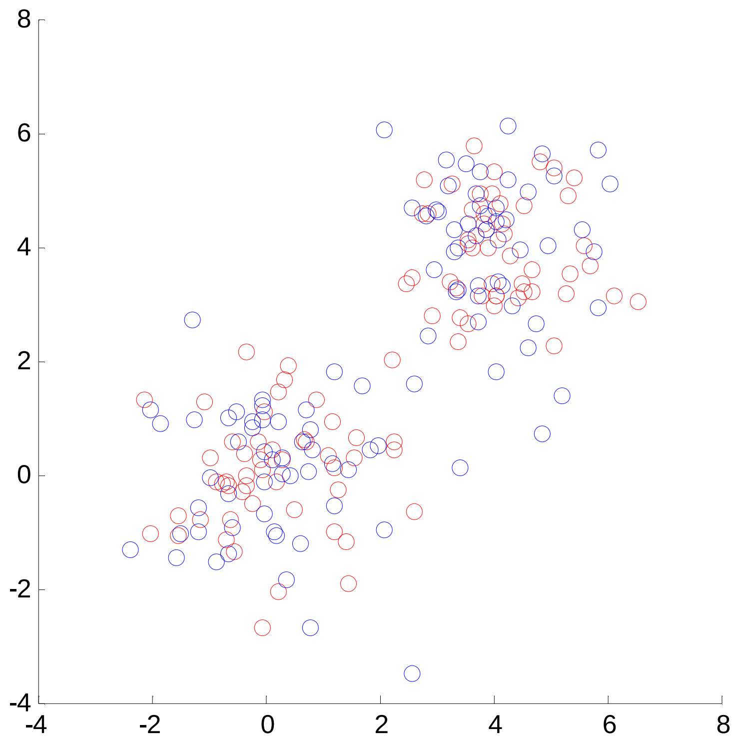
k-Means Clustering

1. Split the samples arbitrarily into *k* equal-sized groups
2. Calculate the mean (centroid) of each of the *k* groups
3. Re-assign each sample to its closest group based upon the distance to the mean
4. Goback to 2 until the algorithm converges (no samples switch groups or the centroids no longer change)

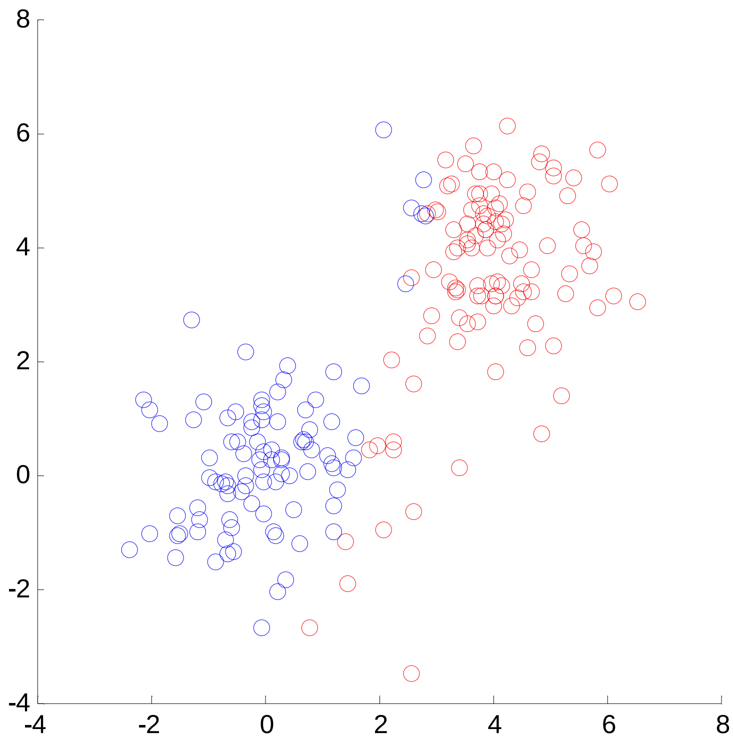
Convergence is guaranteed in *k*-means clustering, but it is not guaranteed to be an optimal set partioning that minimizes

$$V = \sum_{i=1}^{k} \sum_{x_j \in S_i} |x_j - \mu_i|^2$$

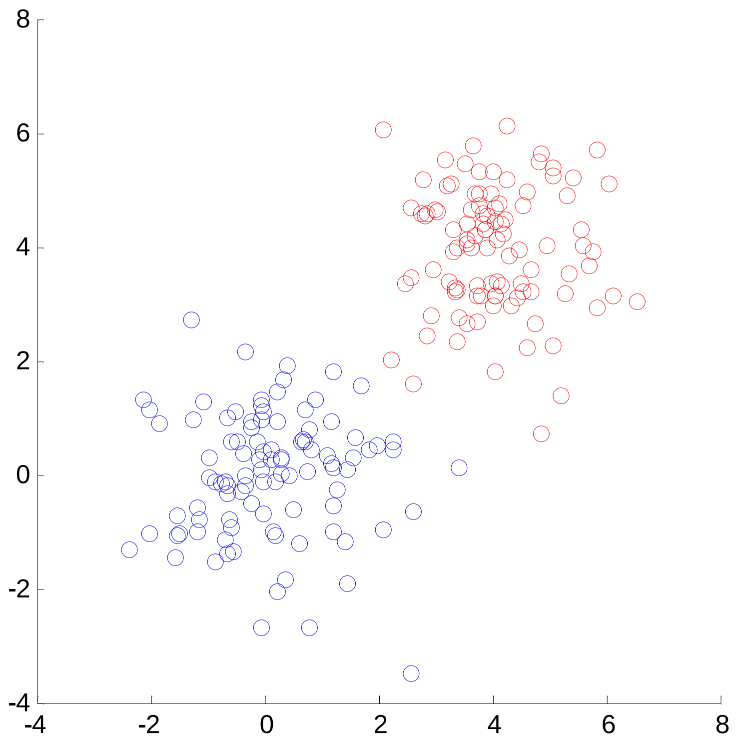
k-Means Clustering



k-Means Clustering



k-Means Clustering

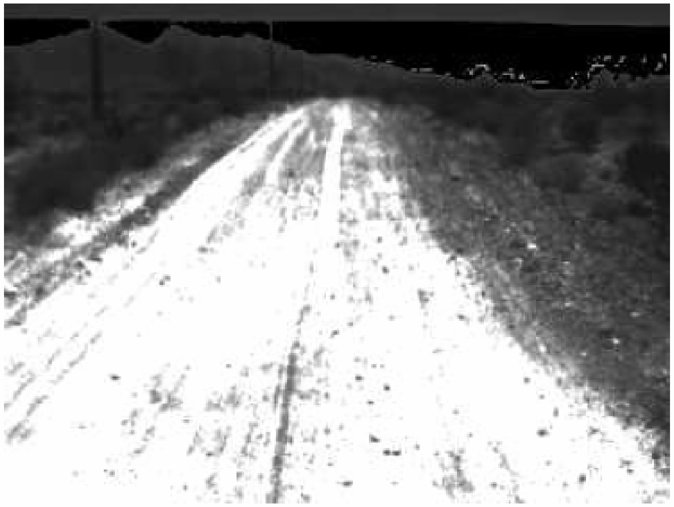


Scoring the Visual Field

- Use Mahalanobis distance for each point (minus the sky and shadows) to the closest learned “viable” road model

$$D_M(x) = \sqrt{(x-\mu)^T\Sigma^{-1}(x-\mu)}.$$

- This assigns a “roadness”score to each point in the image

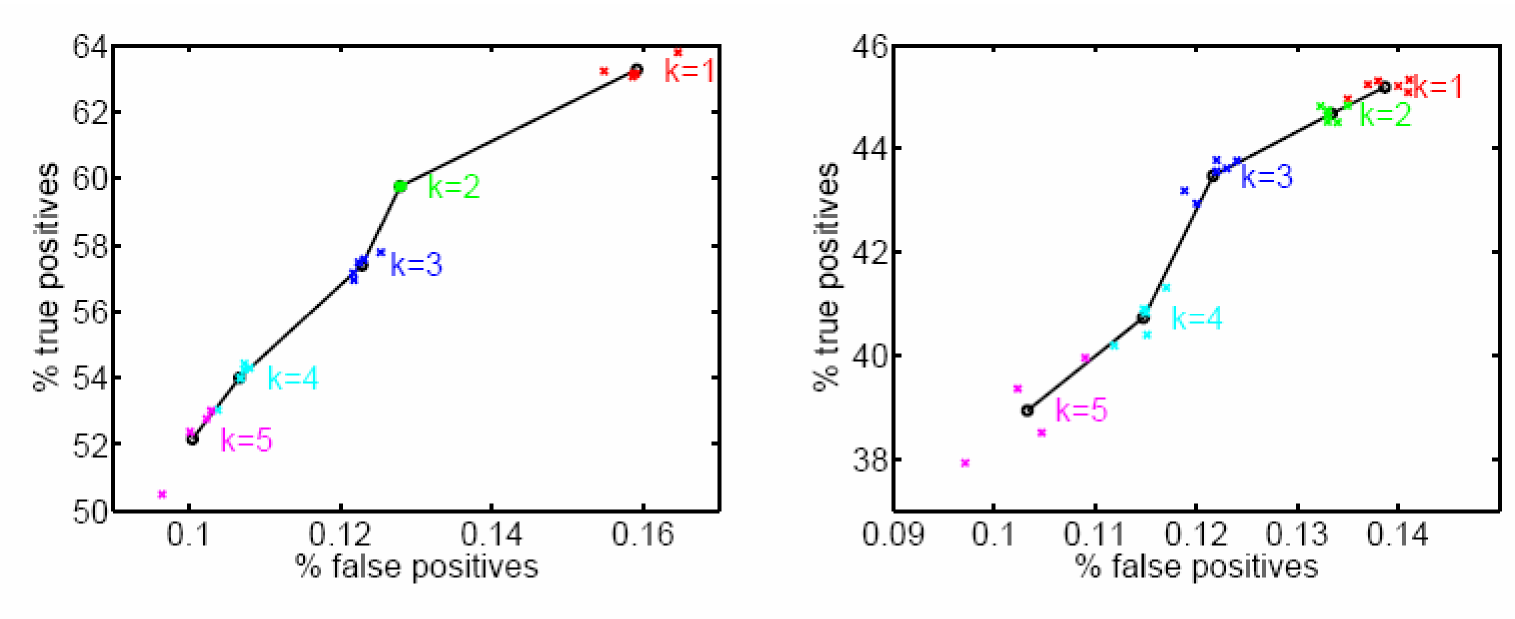


Selecting Identified Road Patches

- Threshold the Mahalanobis image at 3σ to get a binary "drivability" image
- Outlier pixels (leaves, stones, etc.) are cleaned up using morphorlogical operators (erosion & dilation)
- Only accept pixels that have a connected path to the training model to remove additional clutter

Parameter Selection

- Using the actual camera data, system performance was unacceptable.
- Tradeoff between true positives and false positives with size of *k*
- Performance insufficient for real-world use



Solution: *Hack*

- Corrupt the training data by introducing additional "noise" term
- Account for differences in illumination further out
- Artificially adds "texture" to the image data