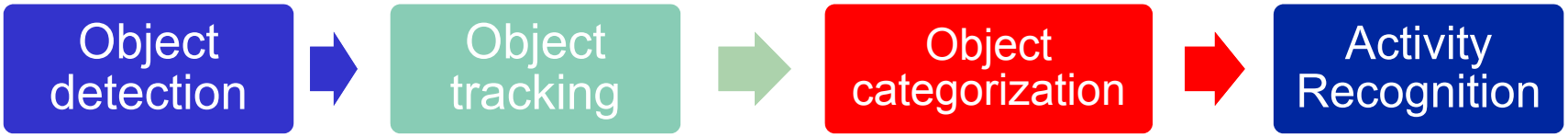
Detection & Tracking

Lecture-18

Tracking Results

Video Analysis Pipeline

Processing pipeline of current automated surveillance systems:



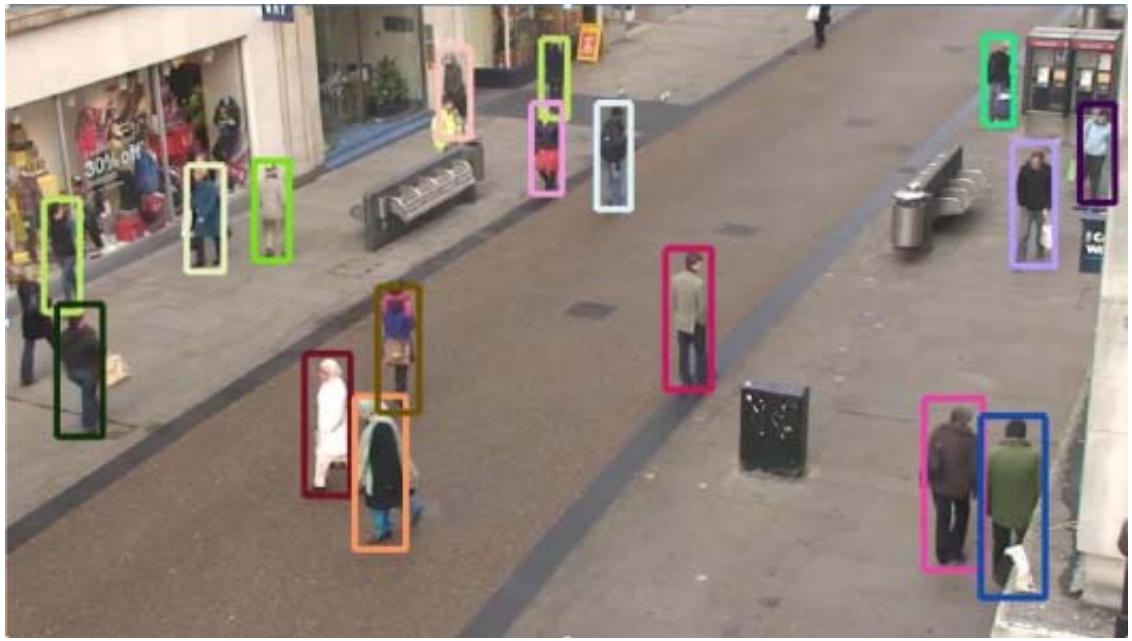
Motivation: Detection and Tracking

- The first step in the process of automated surveillance and tracking.
- Focus of attention method greatly reduces the processing-time

Object Detection

- Single image-based detection (static information)
  - Human detection, vehicle detection
- Motion Based Detection
  - Consecutive frame difference
  - Background subtraction
  - Optical flow-based detection

Human Detection



Objectives:

- Given a sequence of images from a stationary camera identify pixels comprising ‘moving’ objects.
- We call the pixels comprising ‘moving’ objects as ‘foreground pixels’ and the rest as ‘background pixels’

General Solution

–Model properties of the scene (e.g. color, texture e.t.c) at each pixel.

–Significant change in the properties indicates an interesting change.

Introduction: Motion-based Object Detection

Problems in Realistic situations:

- Moving but uninteresting objects
  - e.g. trees, flags or grass.
- Long term illumination changes
  - e.g. time of day.
- Quick illumination changes
  - e.g. cloudy weather
- Shadows
- Other Physical changes in the background
  - e.g. dropping or picking up of objects
- Initialization

Object Detection

- Consecutive Frame Difference

- Background Modeling
  - First frame of sequence
  - Mean of initial frames
  - Median of initial frames
  - Mixture of Gaussians

A Video Clip

Consecutive Frame Difference

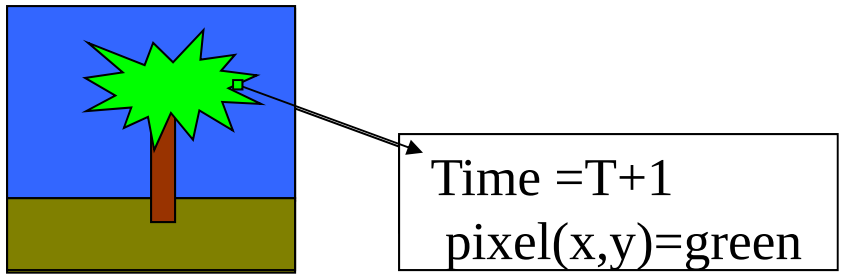
Background Difference

Background Modeling: Issues

- Adaptivity
  - Background model must be adaptive to changes in background.
- Multiple Models
  - Multiple processes generate color at every pixel. The background model should be able to account for these processes.
- Weighting the observations (models)
  - The system must be able to weight the observation to make decisions.

Pixel level Color Modeling

- Multiple Processes are generating color '*x*' at each pixel
  - Where $x=[R,G,B]^T$

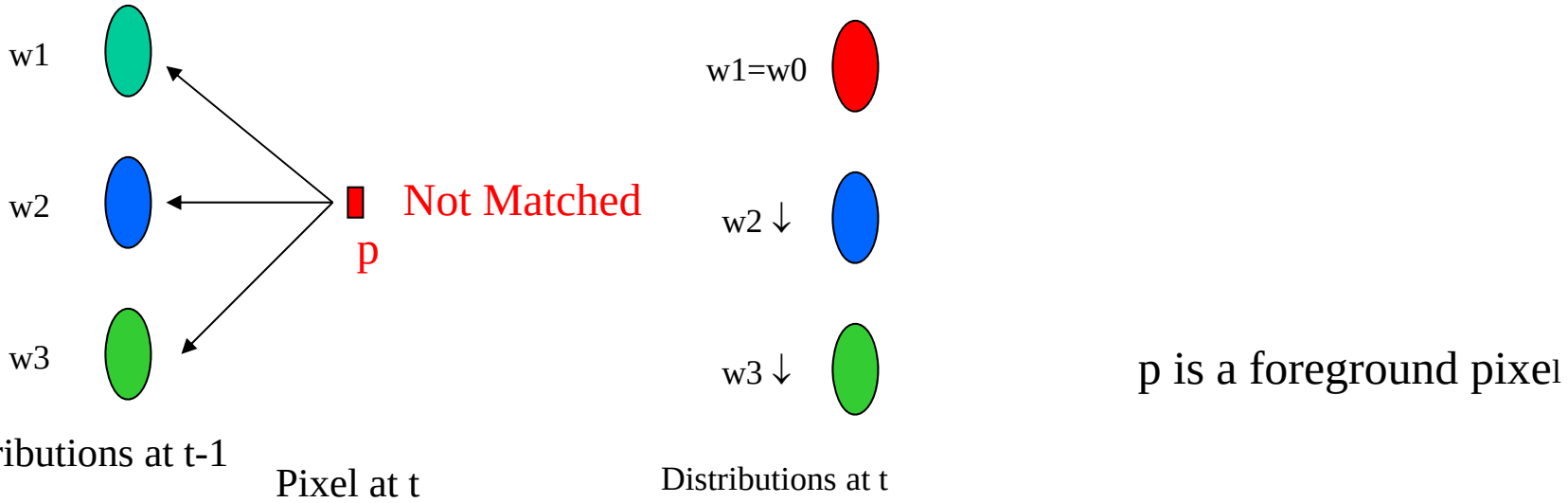


Color based Background Modeling

At each frame

For each pixel

- Calculate distance of pixel's color value from each of the associated K Guassian distributions



$w1 \leq w2 \leq w3$

Color based Background Modeling

For each pixel *(i,j)* at time '*t*' each process is modeled as a Gaussian distribution.

- Guassian distribution is described by a mean '*m*' and a covariance matrix Σ.

$$N(x_{i,j}^t \mid m_{i,j}^t, \sum_{i,j}^t) = \frac{1}{(2\pi)^{\frac{n}{2}} \mid \sum_{i,j}^t \mid} e^{-\frac{1}{2}(x_{i,j}^t - m_{i,j}^t)^T (\sum_{i,j}^t)^{-1} (x_{i,j}^t - m_{i,j}^t)}$$

$x_{i,j}^t$ is 3x1 vector (RGB value) at pixel (i,j) at time t

$m_{i,j}^t$ is 3x1 mean vector of Gaussian at pixel (i,j) at time t

$\sum_{i,j}^t$ is 3x3 covariance matrix at pixel (i,j) at time t

- Each Pixel is modeled as a mixture of Gaussians.
  - –Weight associated with each distribution signifying relevance in recent time.

Mean, Variance and Covariance

Let two features x and y and n observations of each feature be $x_1, x_2, \ldots, x_n$ and $y_1, y_2, \ldots, y_n$ respectively.

Mean: $$m = \begin{bmatrix} m_x & m_y \end{bmatrix}^T = \frac{1}{n}\begin{bmatrix} \sum_{i=1}^{n} x_i & \sum_{i=1}^{n} y_i \end{bmatrix}^T$$

Variance: $$\sigma_x^2 = \frac{1}{n-1}\sum_{i=1}^{n}\left(x_i - m_x\right)^2 \quad \sigma_y^2 = \frac{1}{n-1}\sum_{i=1}^{n}\left(y_i - m_y\right)^2$$

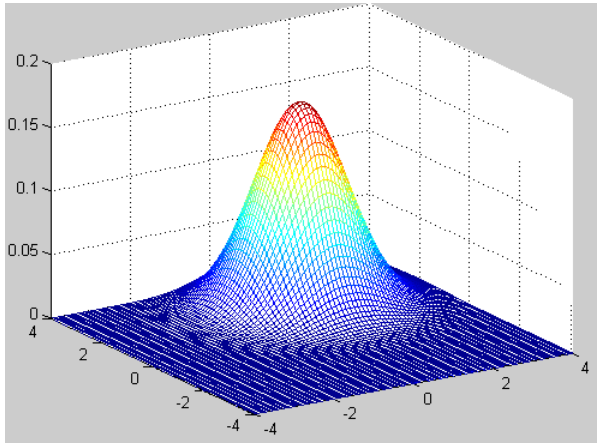
Covariance: $$\sigma_{xy}^2 = \frac{1}{n-1}\sum_{i=1}^{n}\left(x_i - m_x\right)\left(y_i - m_y\right)$$

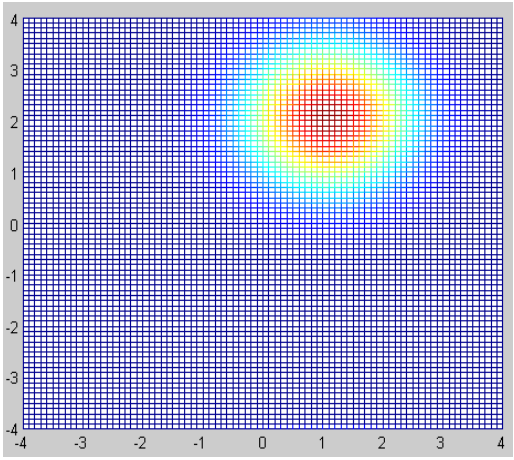
Covariance Matrix: $$\sum = \begin{bmatrix} \sigma_x^2 & \sigma_{xy}^2 \\ \sigma_{xy}^2 & \sigma_y^2 \end{bmatrix}$$

2D Gaussian

$$m = \begin{bmatrix} 1 \\ 2 \end{bmatrix}$$

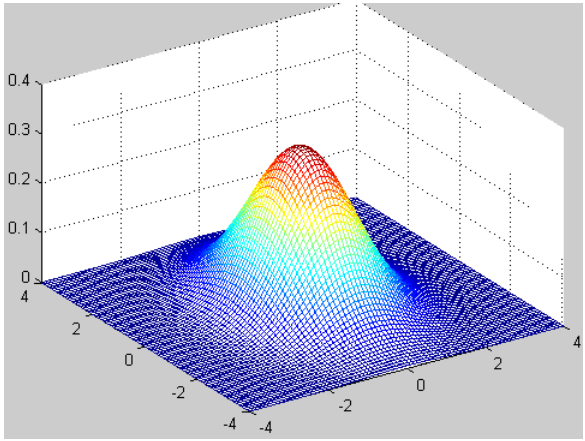
$$\Sigma = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

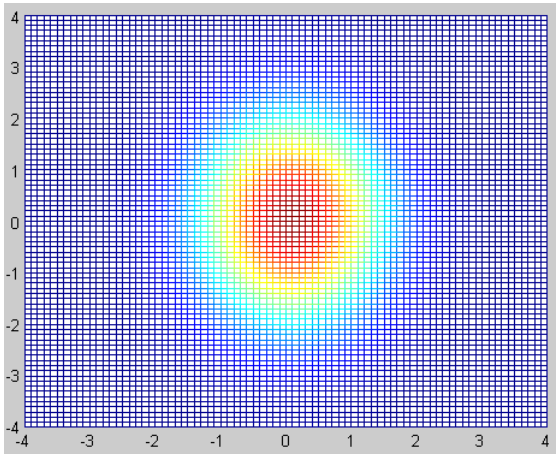




$$m = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

$$\Sigma = \begin{bmatrix} 1 & 0 \\ 0 & 2 \end{bmatrix}$$

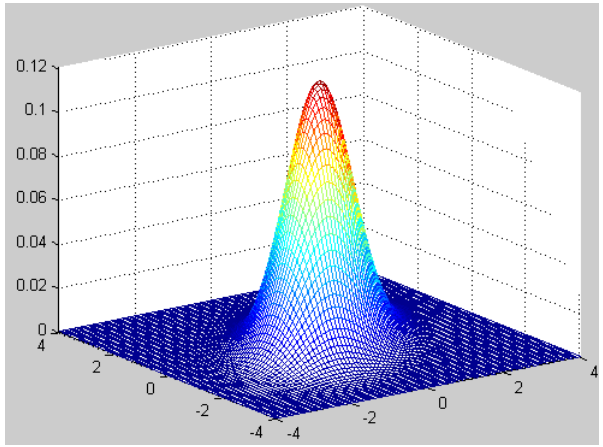


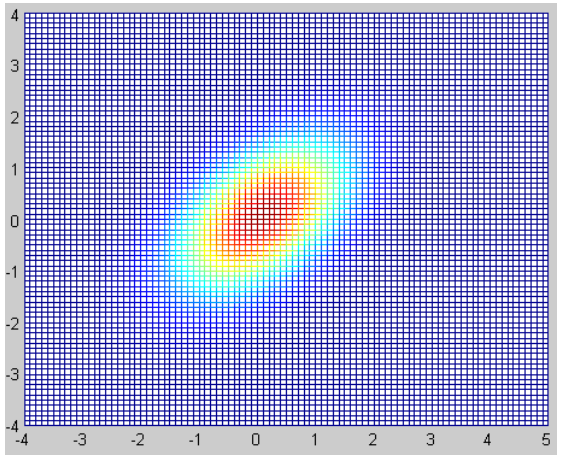


2D Gaussian

$$m = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

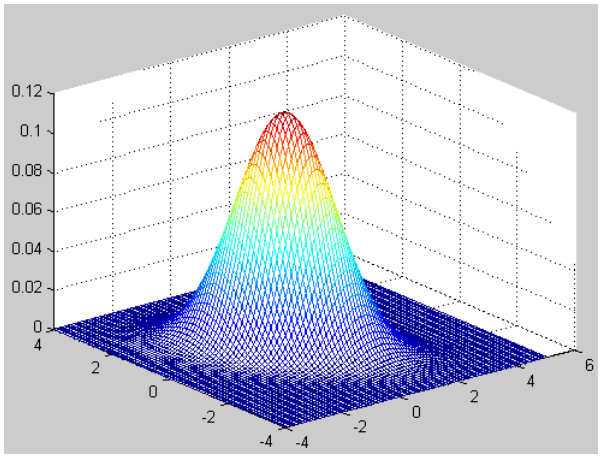
$$\Sigma = \begin{bmatrix} 1 & 0.5 \\ 0.5 & 1 \end{bmatrix}$$

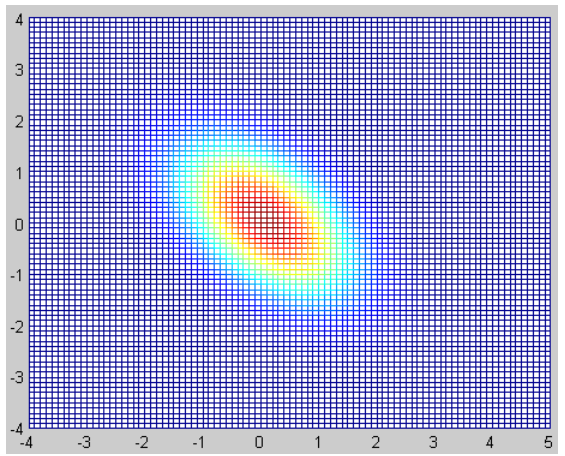




$$m = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

$$\Sigma = \begin{bmatrix} 1 & -0.5 \\ -0.5 & 1 \end{bmatrix}$$





Mahalanobis Distance

Given a vector x, and a normal distribution $N(m,\sum)$, the Mahalanobis distance from feature vector x to the sample mean m is given by

$$d = \sqrt{(x-m)^T (\sum)^{-1} (x-m)}$$

Parameter Update

Let $x_1, x_2, \ldots, x_n$ be the n observations and m_n and σ_n^2 be the mean and variance of these observations respectively. Let x_{n+1} be a new observation, then the updated mean and variance are given by

$$m_{n+1} = \frac{1}{n+1}\sum_{i=1}^{n+1} x_i = m_n + \frac{1}{n+1}\left(x_{n+1} - m_n\right)$$

$$\sigma_{n+1}^2 = \frac{1}{n}\sum_{i=1}^{n+1}\left(x_i - m_{n+1}\right)^2 = \frac{n-1}{n}\sigma_n^2 + \frac{1}{n+1}\left(x_{n+1} - m_n\right)^2$$

Parameter Update

– If a match is found with the k^{th} Gaussian, update parameters

$$m_{i,j}^{t,k} = (1-\rho)m_{i,j}^{t-1,k} + \rho x_{i,j}^{t}$$

$$\sum_{i,j}^{t,k} = (1-\rho)\sum_{i,j}^{t-1,k} + \rho(x_{i,j}^{t} - m_{i,j}^{t})(x_{i,j}^{t} - m_{i,j}^{t})^{T}$$

•where p is a learning parameter

Color based Background Modeling

- If a match is not found
  - Replace lowest weight distribution with a new distribution such that

$$m_{i,j}^{t,new} = x_{i,j}^{t}$$

$$\sum_{i,j}^{t,new} = \sum^{initial}$$

- The prior weights of K distributions are adjusted as

$$\omega_{i,j}^{t} = (1-\alpha)\omega_{i,j}^{t-1} + \alpha(M_{i,j}^{t-1})$$

  - M is1 for model that matched and 0 for others

Color based Background Modeling

- Foreground= Matched distributions with weight< T
  + Unmatched pixels

Summary

- Each pixel is an independent statistical process, which may be combination of several processes.
  - Swaying branches of tree result in a bimodal behavior of pixel intensity.

- The intensity is fit with a mixture of K Gaussians.

$$N(x_{i,j}^t \mid m_{i,j}^t, \sum_{i,j}^t) = \frac{1}{(2\pi)^{\frac{n}{2}} \mid \sum_{i,j}^t \mid} e^{-\frac{1}{2}(x_{i,j}^t - m_{i,j}^t)^T (\sum_{i,j}^t)^{-1} (x_{i,j}^t - m_{i,j}^t)}$$

- For simplicity, it may be assumed that RGB color channels are independent and have the same variance σ^2. In this case $\sum_{i,j}^t = \sigma^2 I$, where I is a 3x3 idenity matrix.

Summary

- Every new pixel is checked against all existing distributions. The match is the distribution with Mahalanobis distance less than a threshold.

- The mean and variance of unmatched distributions remain unchanged. For the matched distributions they are updated as

$$m_{i,j}^{t,k} = (1-\rho)m_{i,j}^{t-1,k} + \rho x_{i,j}^{t}$$

$$\sum_{i,j}^{t,k} = (1-\rho)\sum_{i,j}^{t-1,k} + \rho(x_{i,j}^{t} - m_{i,j}^{t})(x_{i,j}^{t} - m_{i,j}^{t})^{T}$$

Summary

- For the unmatched pixel, replace the lowest weight Gaussian with the new Gaussian with mean at the new pixel and an initial estimate of covariance matrix.

- The weights are adjusted:

$$\omega_{i,j}^{t} = (1-\alpha)\omega_{i,j}^{t-1} + \alpha(M_{i,j}^{t-1})$$

$$M_{ij}^{t-1} = \begin{cases} 1 & \text{if distribution matches} \\ 0 & \text{otherwise} \end{cases}$$

  - Foreground= Matched distributions with weight< T
    + Unmatched pixels

Connected Components

- So far each pixel is declared background or foreground, we need to connect them into blobs.
- We need to apply connected component algorithm

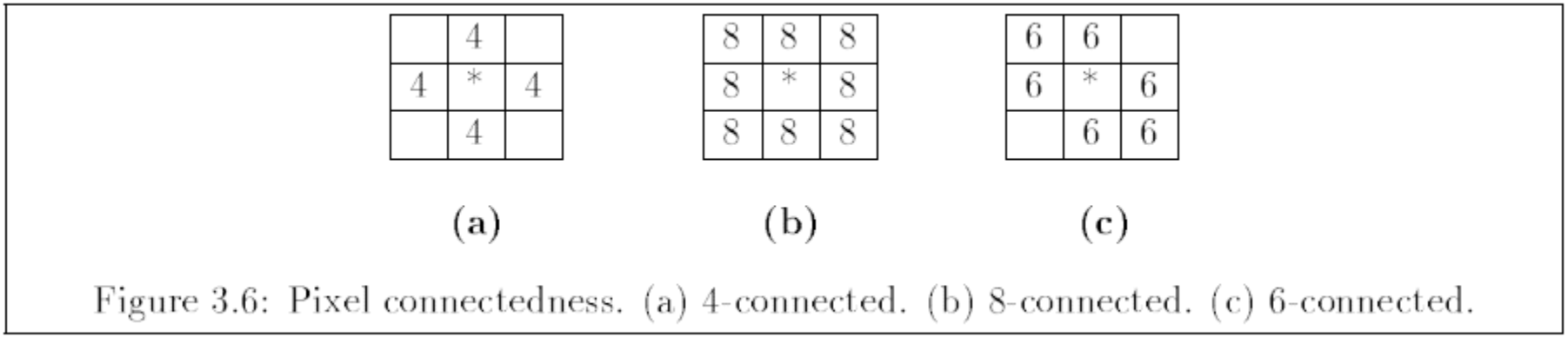


Figure 3.6: Pixel connectedness. (a) 4-connected. (b) 8-connected. (c) 6-connected.

Section 3.4 in Fundamental of Computer Vision

Recursive Algorithm

1. Scan the binary image left to right, top to bottom.
2. If there is an unlabeled pixel with a value of '1' assign a new label to it.
3. Recursively check the neighbors of the pixel in step 2 and assign the same label if they are unlabeled with a value of '1'.
4. Stop when all the pixels of value '1' have been labeled.

Figure 3.7: Recursive Connected Component Algorithm.

Sequential Algorithm

1. Scan the binary image left to right, top to bottom.
2. If an unlabeled pixel has a value of '1', assign a new label to it according to the following rules:

$$\begin{matrix} & 0 \\ 0 & 1 \end{matrix} \rightarrow \begin{matrix} & 0 \\ 0 & \mathrm{L} \end{matrix} \qquad \begin{matrix} & 0 \\ \mathrm{L} & 1 \end{matrix} \rightarrow \begin{matrix} & 0 \\ \mathrm{L} & \mathrm{L} \end{matrix}$$

$$\begin{matrix} & \mathrm{L} \\ 0 & 1 \end{matrix} \rightarrow \begin{matrix} & \mathrm{L} \\ 0 & \mathrm{L} \end{matrix} \qquad \begin{matrix} & \mathrm{L} \\ \mathrm{M} & 1 \end{matrix} \rightarrow \begin{matrix} & \mathrm{L} \\ \mathrm{M} & \mathrm{L} \end{matrix} \quad (\text{Set } L = M).$$

3. Determine equivalence classes of labels.
4. In the second pass, assign the same label to all elements in an equivalence class.

Figure 3.8: Sequential Connected Component Algorithm.

Results

Color based Background Modeling

Pros

–Handles slow changes in illumination conditions

–Can accommodate physical changes in the background after a certain time interval.

–Initialization with moving objects will correct itself after a certain time interval.

Cons

– Cannot handle quick changes in illumination conditions e.g. cloudy weather

– Initialization with moving objects

–Shadows

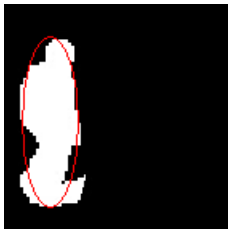
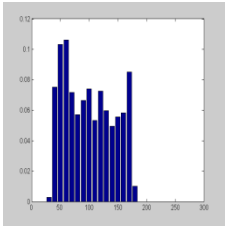
–Physical Changes in Background

Tracking Objects

- Task Definition
  - Establish correspondence between observations over time

Tracking: Solution

- For each object P_k

–Shape is modeled by a Gaussian Distribution, $g_k(x)$.

–Color is modeled by a normalized histogram, h_k.

Tracking: Solution

–Motion is modeled by a linear prediction model.

$$pos_t = \sum_{k=1}^{p} a_k (pos_{t-k}) \qquad pos_t = pos_{t-1} + (pos_{t-2} - pos_{t-1})$$

–Size is in terms of number of pixels, n_k

Tracking: Solution

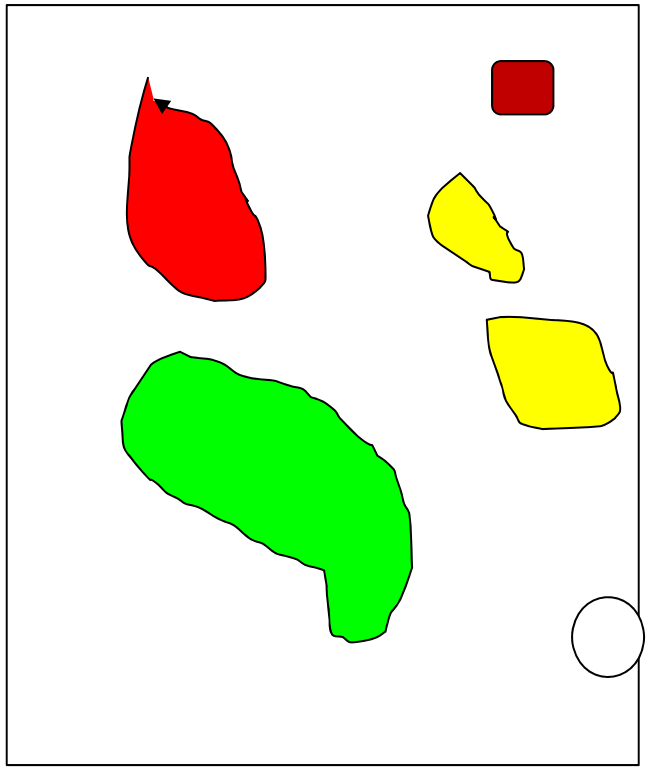
- Establishing Correspondence
  - Each pixel x , where $x \in R_i$, votes for a label l,

    where $$l = \arg\max_k (g_k(x) h_k(c(x))$$

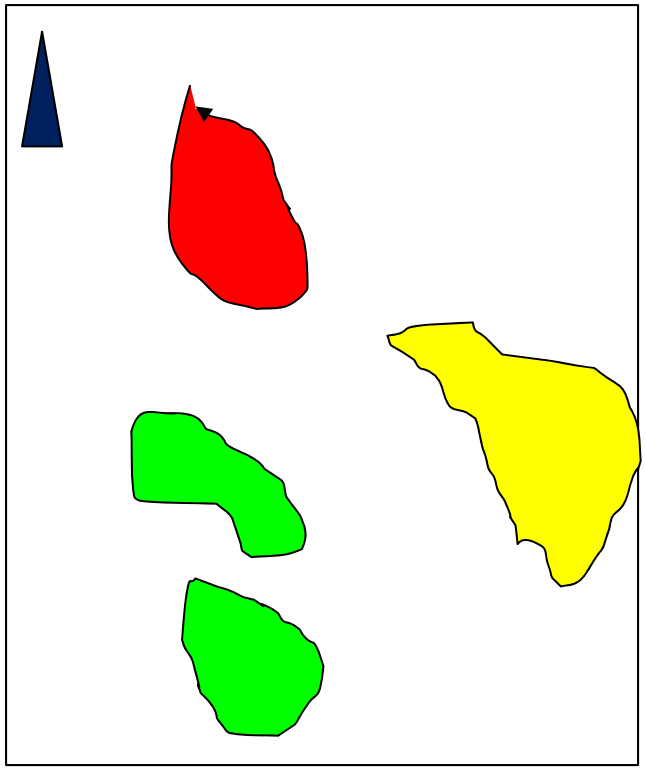
  - Let $V_{i,k}$ be the number of votes from R_i to P_k.

An Example

Frame -1



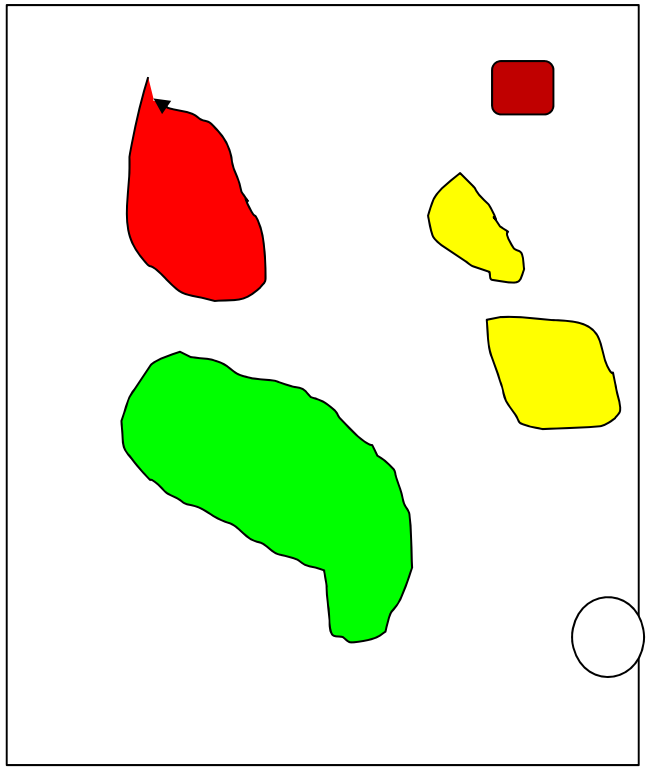
Frame -2



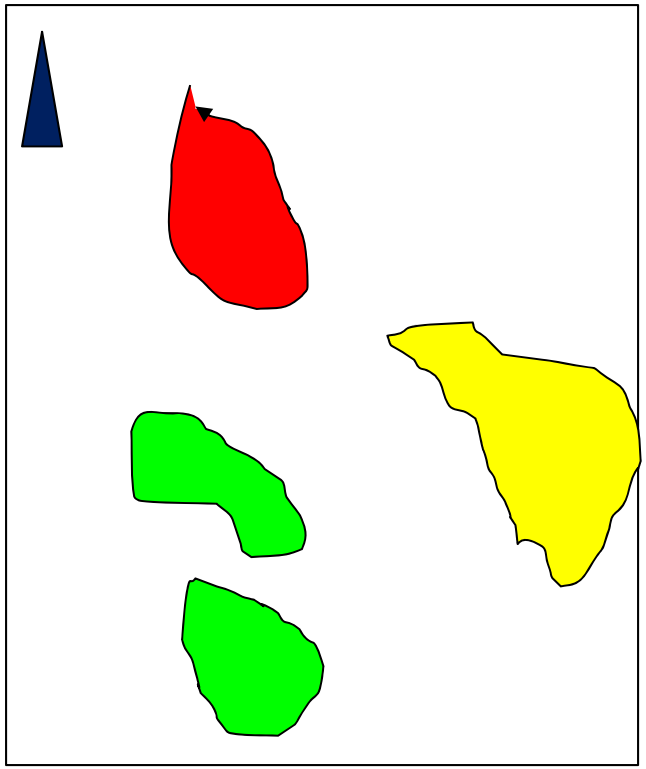
Algorithm

– if $(V_{i,k}/n_k)>T_c$ & $(V_{i,q}/n_q)<T_c \quad \forall q \neq k$ then all pixels in R_i are used to update P_k's models. In case more than one region satisfies this condition then it is splitting.

Frame -1



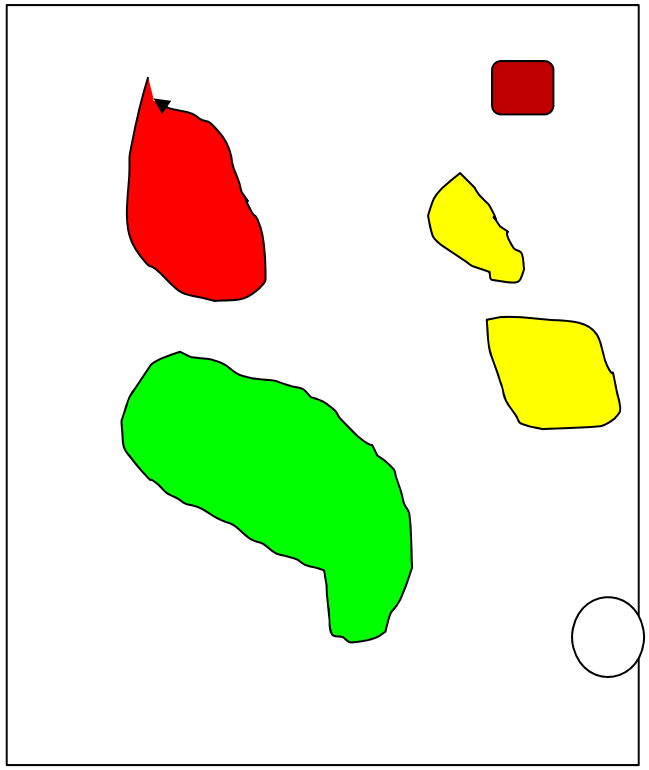
Frame -2



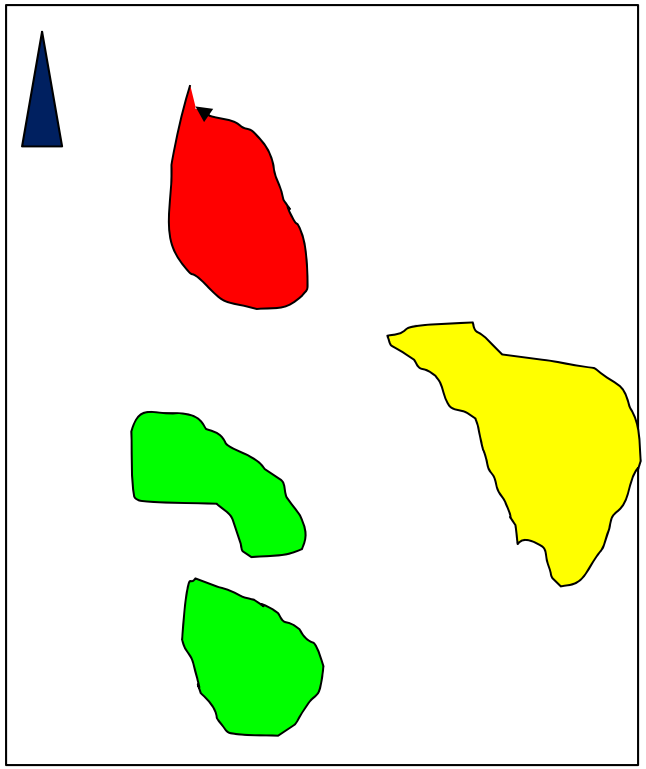
Algorithm

– if $(V_{i,k}/n_k)>T_c$ & $(V_{i,q}/n_q)>T_c$ then pixels that voted for R_i are used to update P_k's models. (merging)

Frame -1



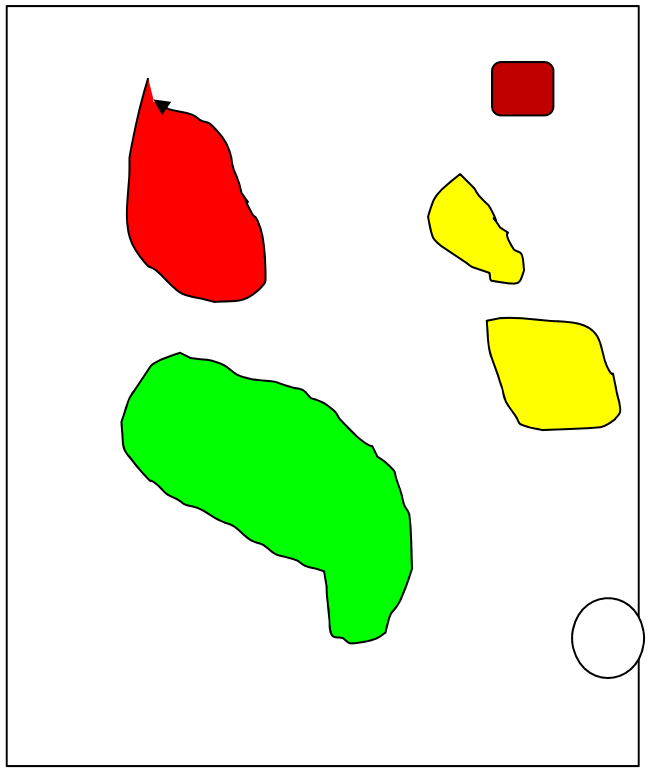
Frame -2



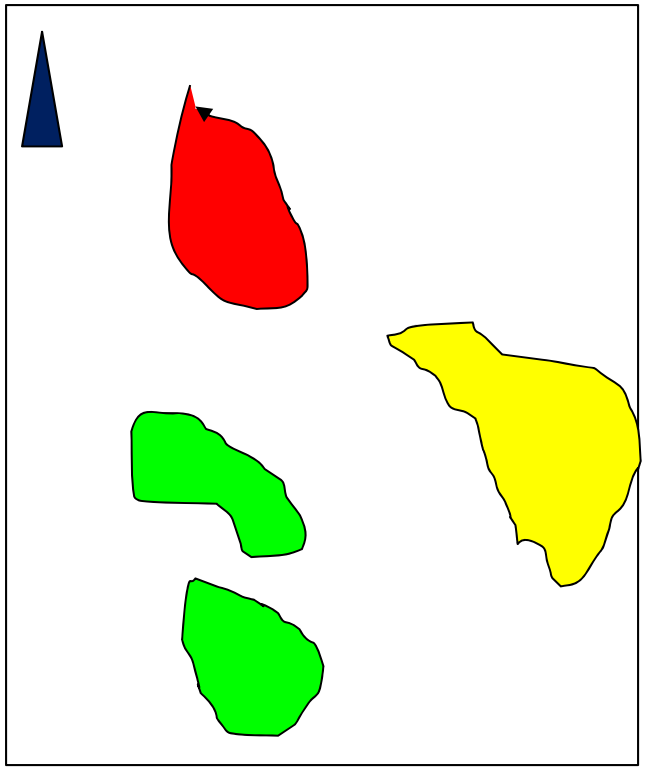
Algorithm

– if $(V_{i,k}/n_k)<T_c \quad \forall i$, then prediction is used to update spatial model. (No observation matches the model k: occlusion. If prediction is close to boundary then exit.)

Frame -1



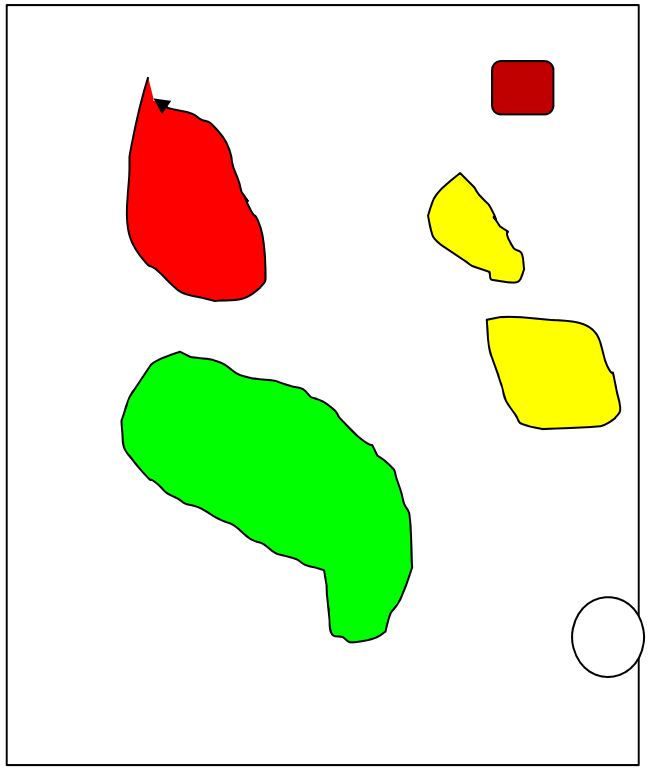
Frame -2



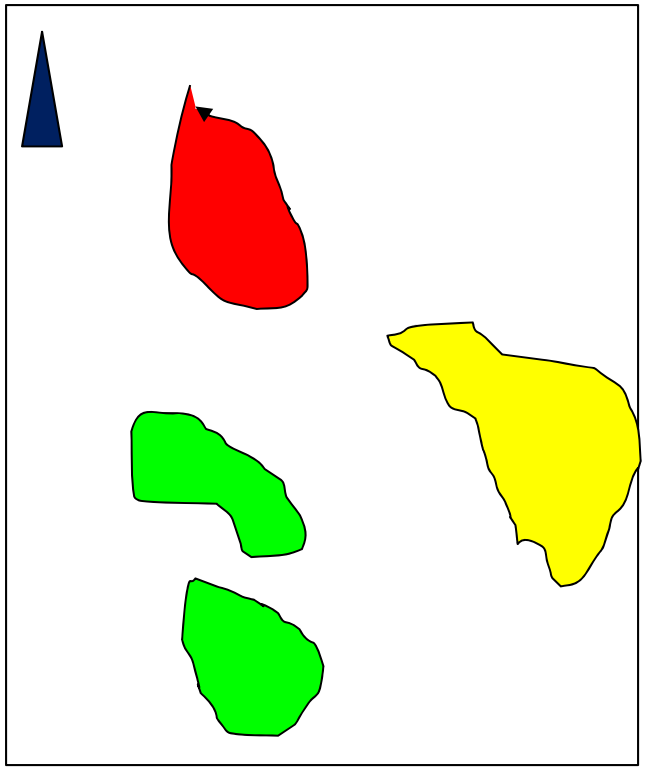
Algorithm

– if $(V_{i,k}/n_k)<T_c \ \forall_k$, then region R_i does not match any model. This is a new entry.

Frame -1

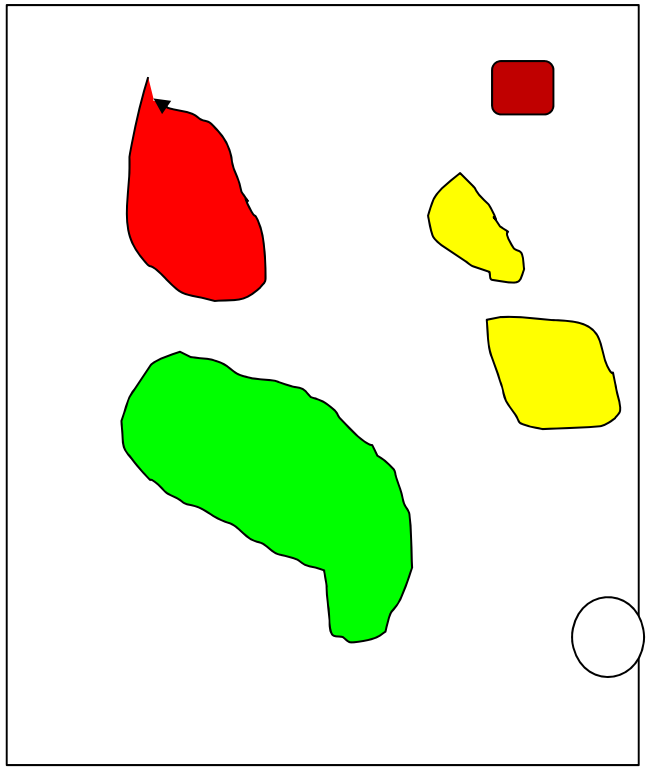


Frame -2

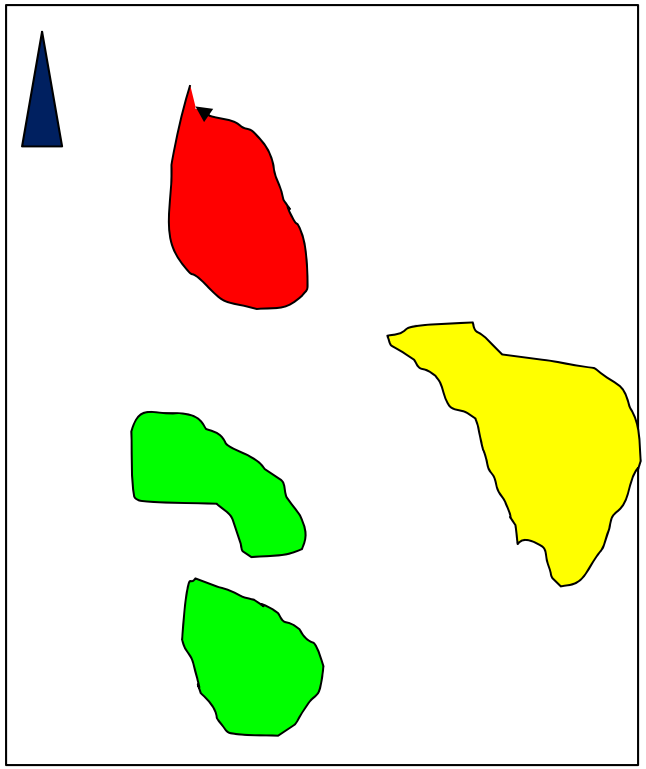


An Example

Frame -1



Frame -2



Tracking: Solution

– if $(V_{i,k}/n_k)>T_c$ & $(V_{i,q}/n_q)<T_c \quad \forall q \neq k$ then all pixels in R_i are used to update P_k's models. In case more than one region satisfies this condition then it is splitting.

– if $(V_{i,k}/n_k)>T_c$ & $(V_{i,q}/n_q)>T_c$ then pixels that voted for R_i are used to update P_k's models. (merging)

– if $(V_{i,k}/n_k)<T_c \quad \forall i$, then prediction is used to update spatial model. (No observation matches the model k: occlusion. If prediction is close to boundary then exit.)

– if $(V_{i,k}/n_k)<T_c \quad \forall_k$, then region R_i does not match any model. This is a new entry.

Tracking Results

Tracking Results

‘KNIGHT’: A Real Time Surveillance System

Reading Material

- C. Stauffer and W.E.L. Grimson, “Learning patterns of activity using real time tracking,” IEEE Trans. On PAMI, 22(8):747-757, Aug 2000.

- Scene monitoring with a forest of cooperative sensors by Javed, Omar, Ph.D., University of Central Florida, 2005, 175 pages. http://server.cs.ucf.edu/~vision/papers/theses/omar_theses.pdf Section 4.2

- Section 3.4: Connected Component Algorithms In Fundamental of Computer Vision