Detecting abnormal events

Jaechul Kim

Purpose

- Introduce general methodologies used in abnormality detection
- Deal with technical details of selected papers

Abnormal events

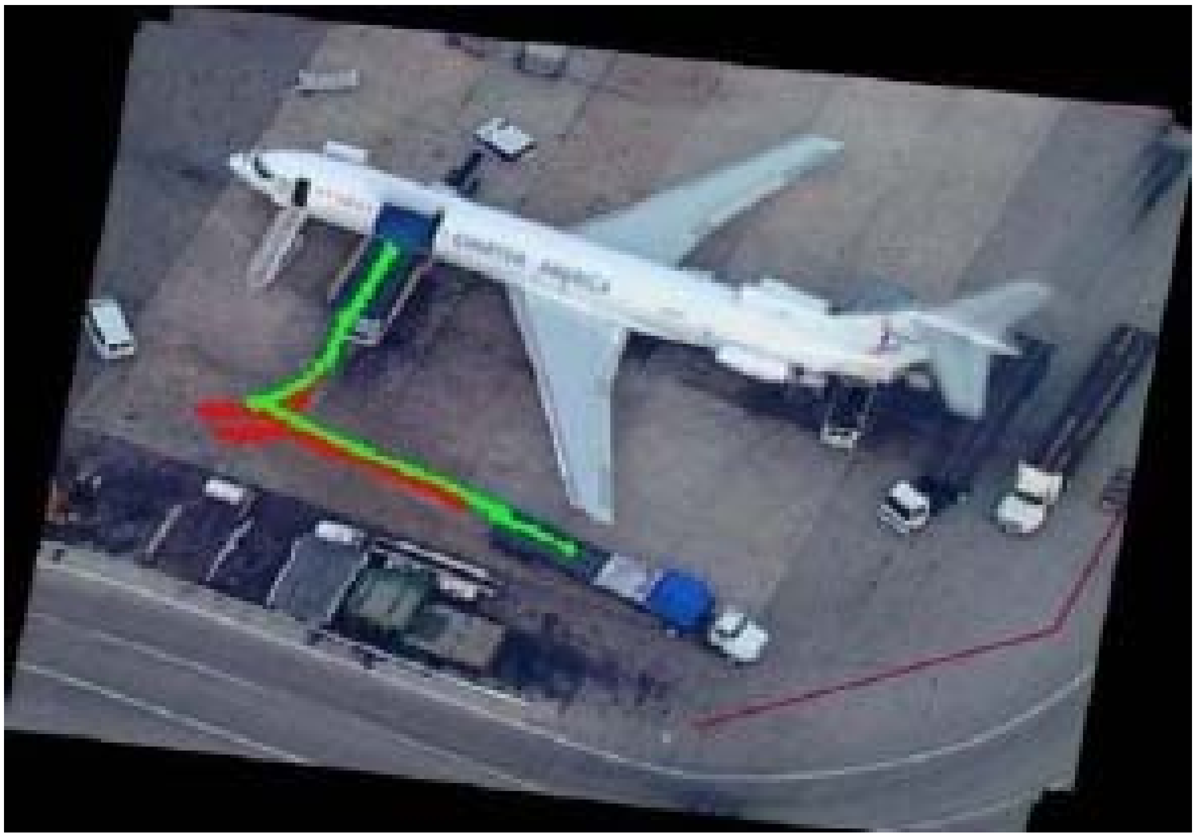
- Easy to verify, but hard to describe
- Generally regarded as rare events or unseen events
  - Detection of outliers

Overview: Taxonomy of approaches

- What representations are used to describe individual event?
  - Tracked trajectory based representation
    - Intuitive way to describe an event
  - Low-level feature based representation
    - Robust to the cluttered scene
    - Recently more preferred

Overview: Taxonomy based on event representation

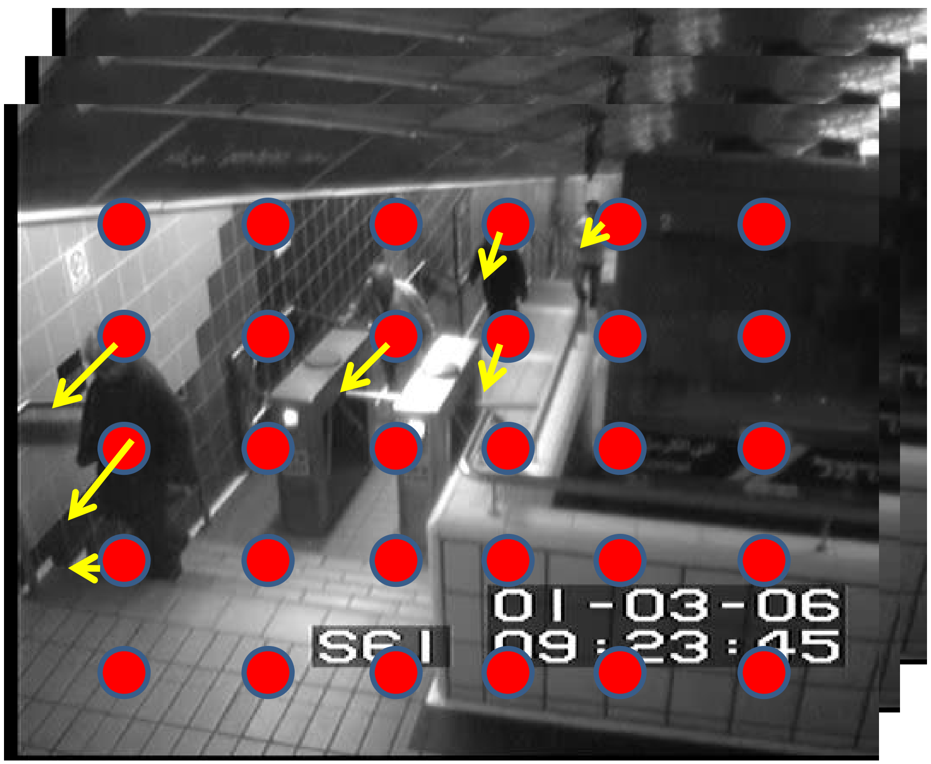
- Tracked trajectory based representation



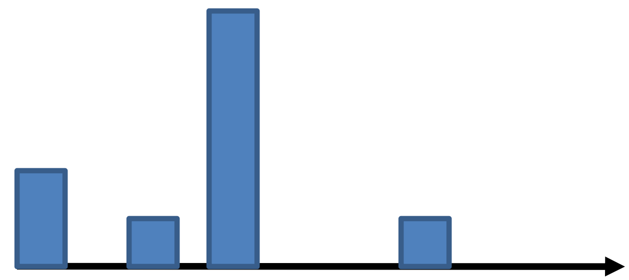
Tracked path of an interest object defines a single event.

Overview: Taxonomy based on event representation

- Low-level feature based representation



Optical Flows, Blob motion, etc



Histogram of optical flows

[0,0,0,4,1,0,
10,0,8,4,0,0,
10,0,0,0,0,0,
1,0,0,0,0,0,
0,0,0,0,0,0]

Feature vector concatenating each optical flows

Overview: Taxonomy of approaches

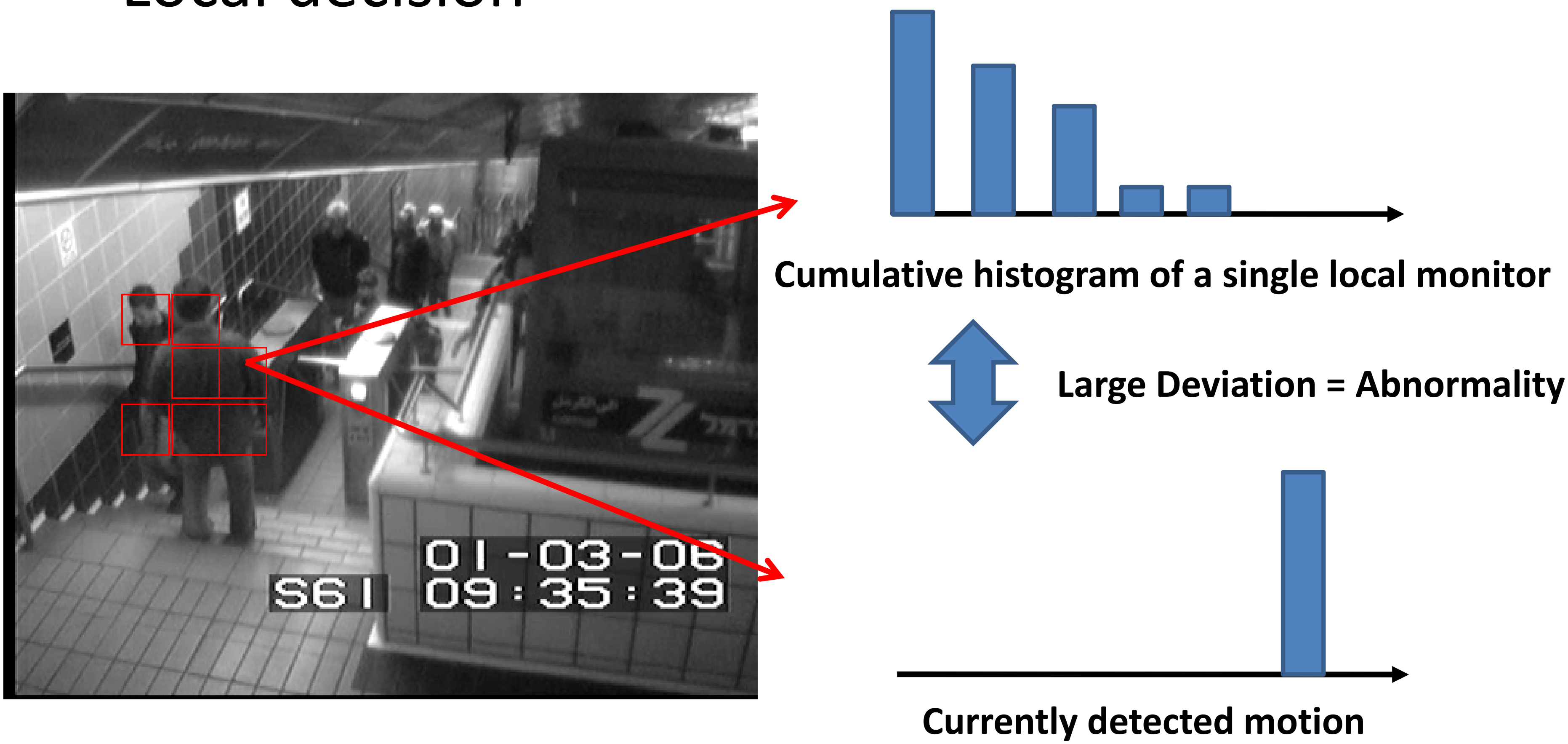
- What techniques are used to determine anomaly of the event?
  - Local decision
    - Decide an anomaly solely based on the observation of locally detected features
  - Learning-based method
    - Detect statistical outliers using the learnt patterns
  - Search-based method
    - Search the similar images to the input in the dataset

Overview: Taxonomy based on anomaly decision method

- Local decision
  - Each local region independently flags an alert to anomaly

Overview: Taxonomy based on anomaly decision method

- Local decision



Overview: Taxonomy based on anomaly decision method

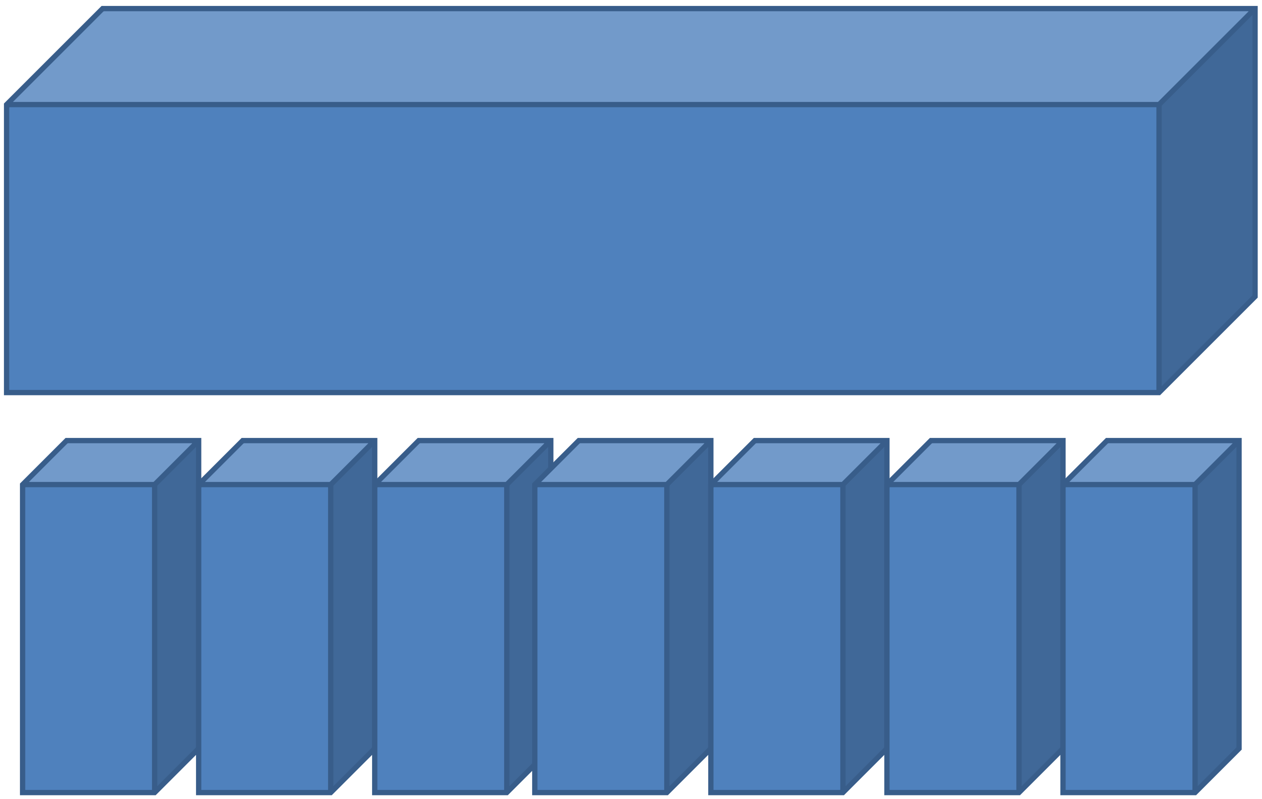
- Pros
  - Easy to implement, fast to compute
- Cons
  - Hard to handle a relationship between co-occurring events in a single frame or an ordering of event sequences over multiple frames

Overview: Taxonomy based on anomaly decision method

- Learning-based method
  - Learn normal activities first, and then detect abnormal events as an outlier of the learnt patterns

Overview: Taxonomy based on anomaly decision method

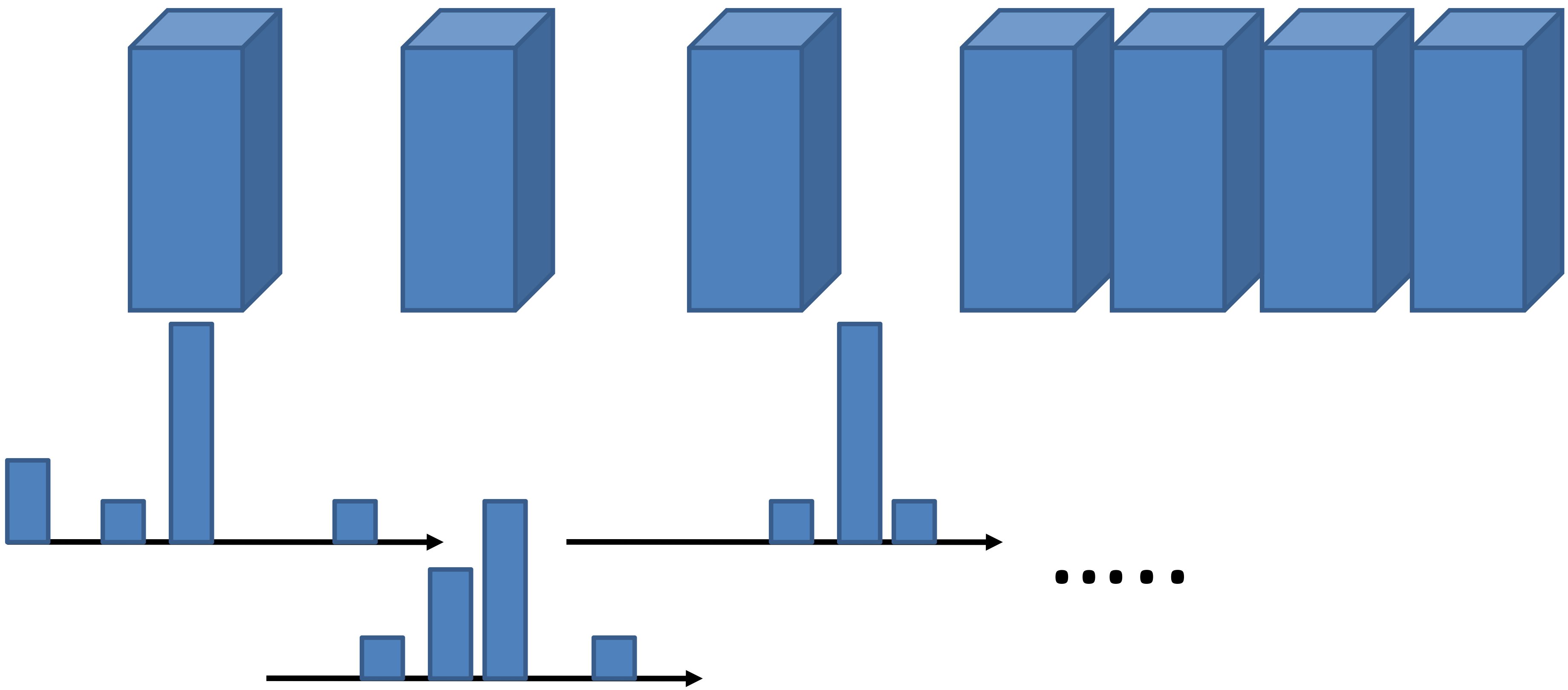
- Learning-based method



Step1: Divide a video into segments(=a single activity unit)

Overview: Taxonomy based on anomaly decision method

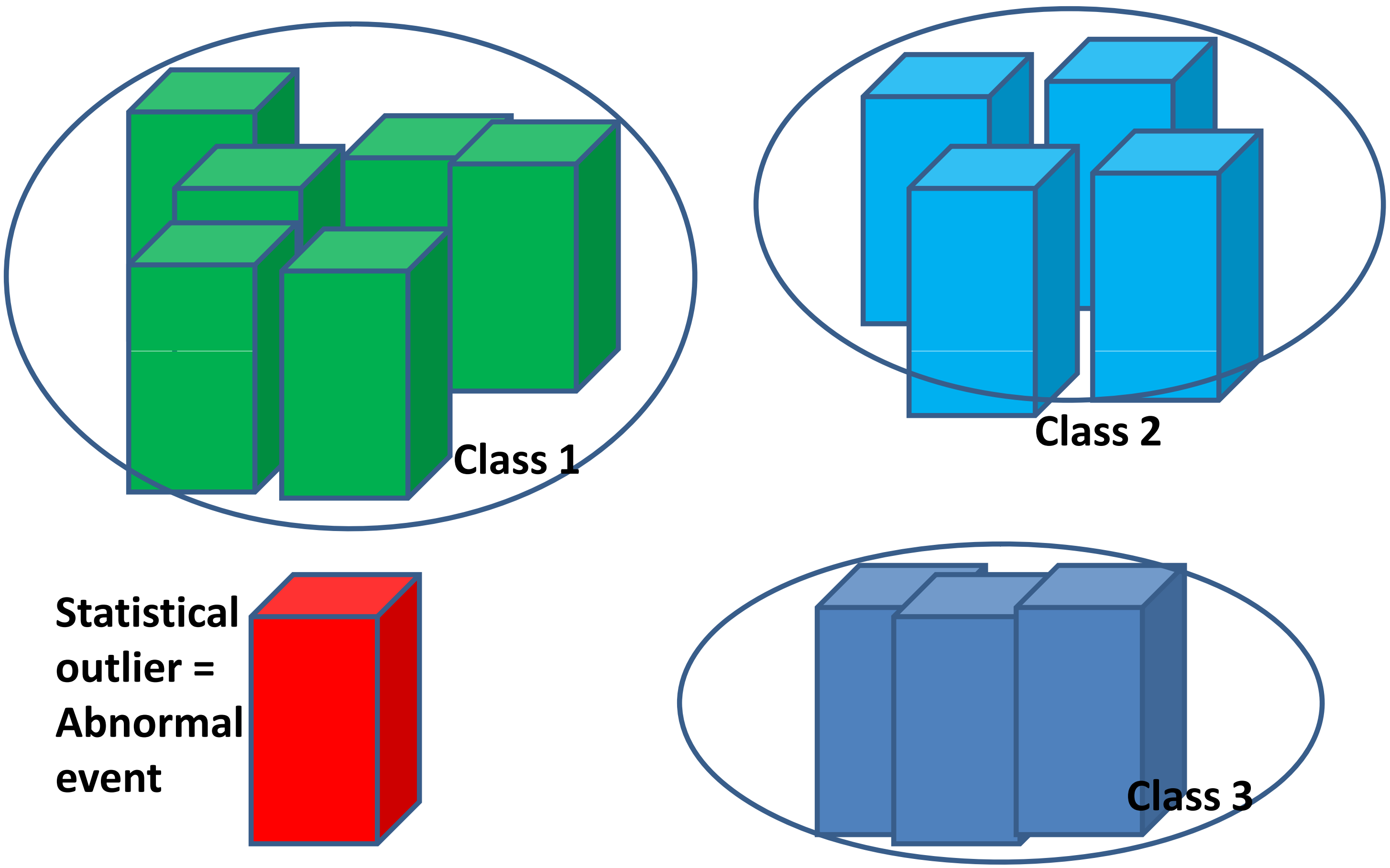
- Learning-based method



Step2: Compute a similarity measure between each segment

Overview: Taxonomy based on anomaly decision method

- Learning-based method



Step3: Learn a classifier that recognizes normal activities

Overview: Taxonomy based on anomaly decision method

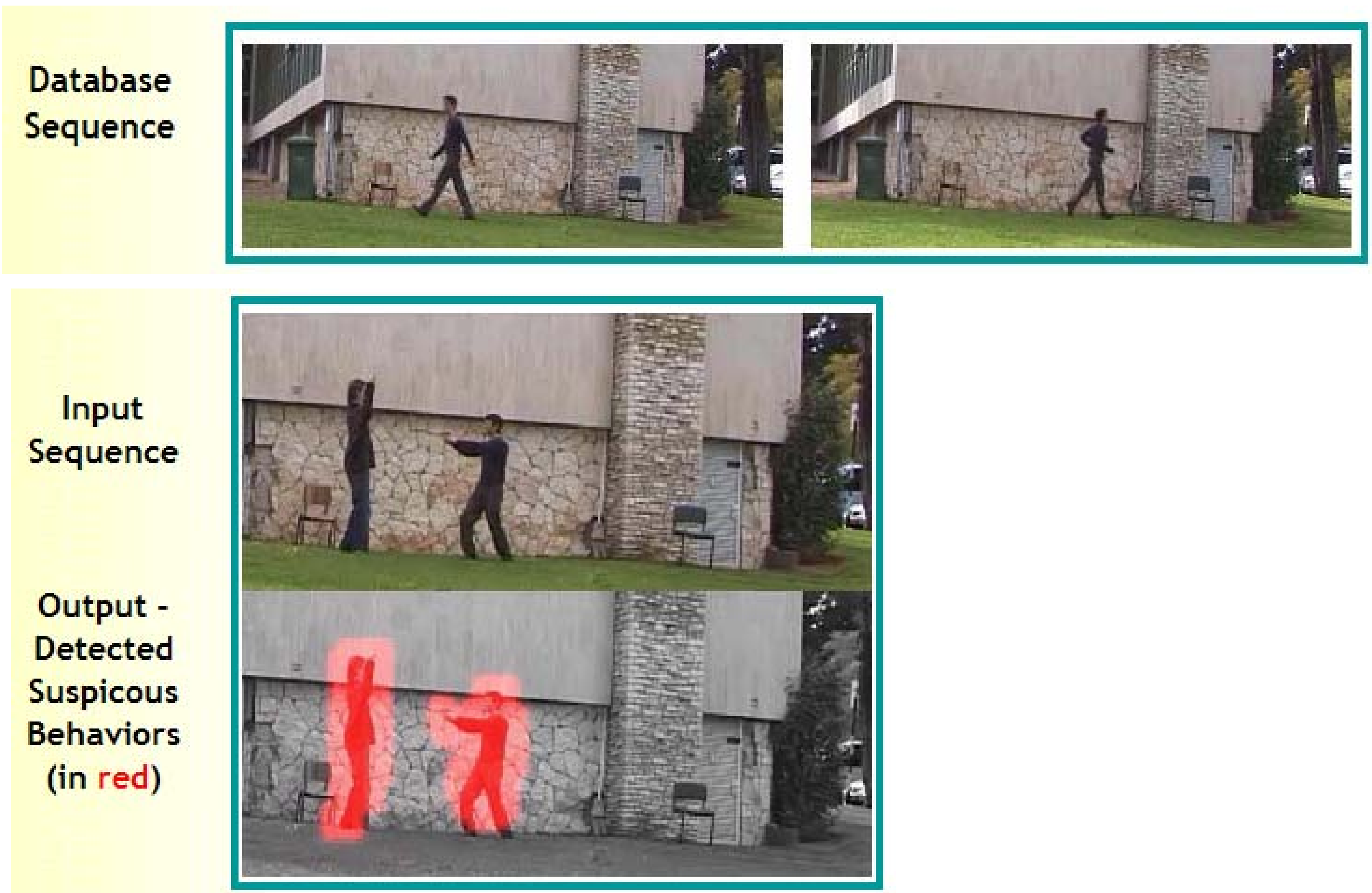
- Pros
  - Principled way to considering an ordering of events as well as co-occurring events
- Cons
  - Hard to handle the evolution of activities
    - Inadequate to online application
  - Hard to localize an abnormality

Overview: Taxonomy based on anomaly decision method

- Search-based method
  - Search whether the input image has similar images exist in the database

Overview: Taxonomy based on anomaly decision method

- Search-based method



Overview: Taxonomy based on anomaly decision method

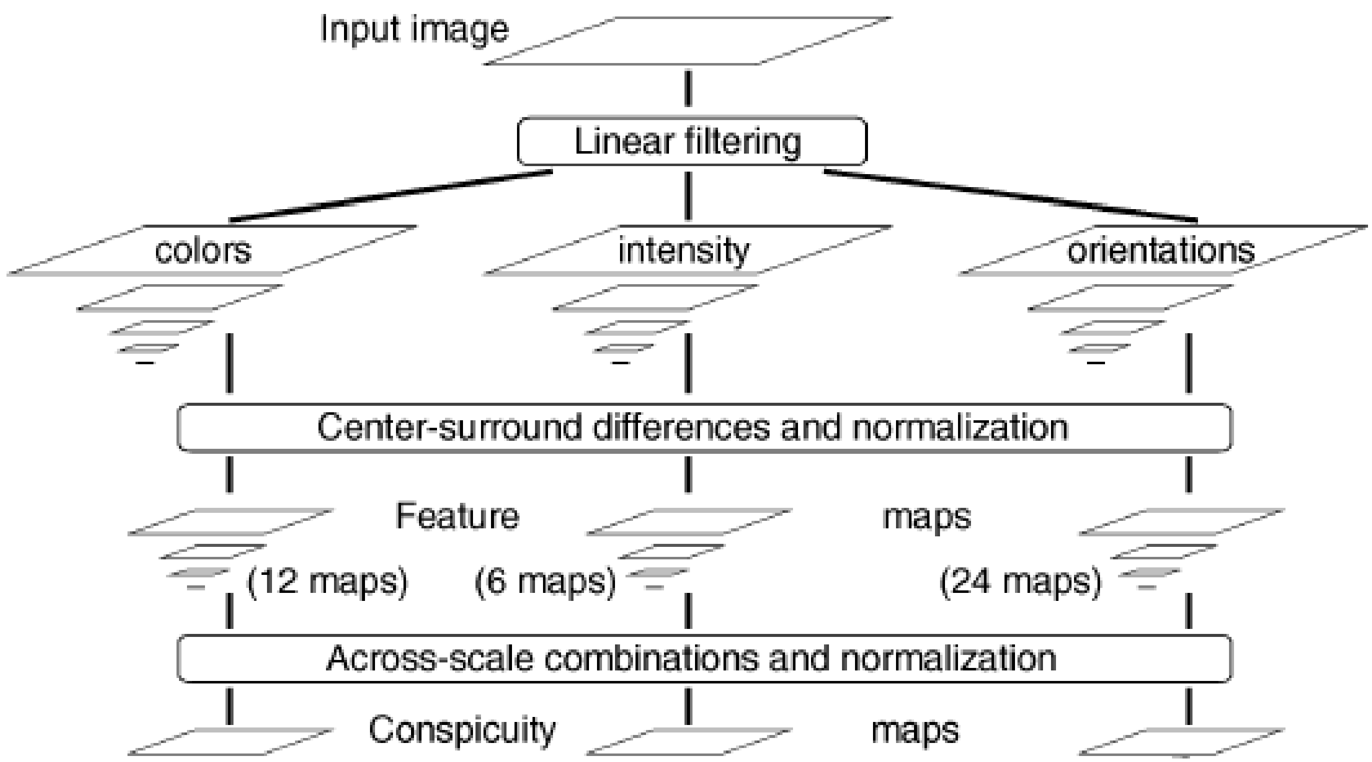
- Pros
  - Accurate detection from exhaustive search
- Cons
  - Time-consuming

Case study 1 : Local decision method

- "A principled approach to detecting surprising events in video", Laurent Itti and Pierre Baldi, CVPR 2005

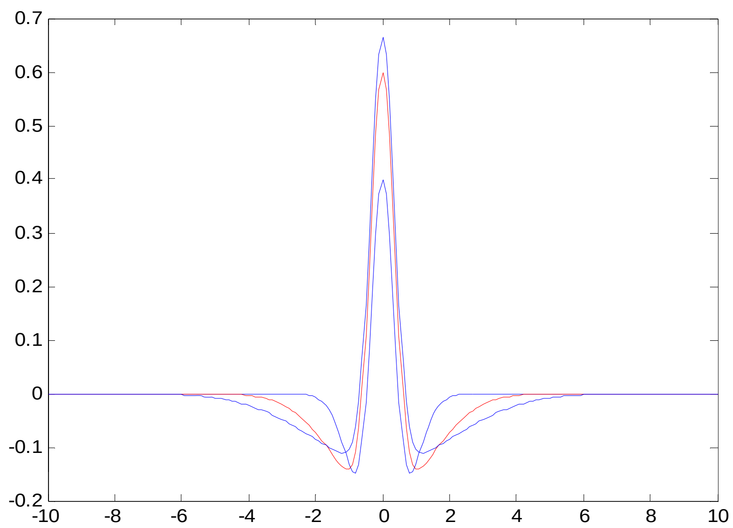
Case study 1 : Local decision method

- Step 1: Detect local features in all pixels over multiple scales and multiple channels



Case study 1 : Local decision method

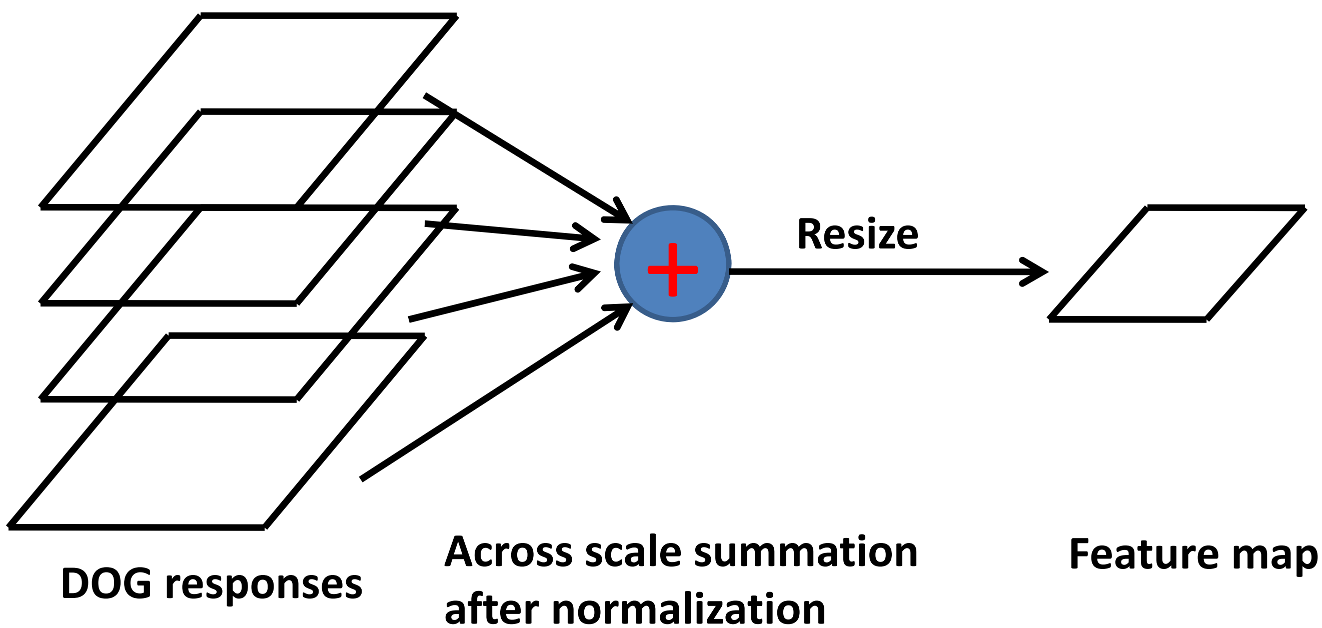
- Step1
  - For each channel, DOG filters over multiple scales are applied to the image: Blob like features are extracted from each channel (motion, intensity…)



DOGs in several scale differences (1D case)

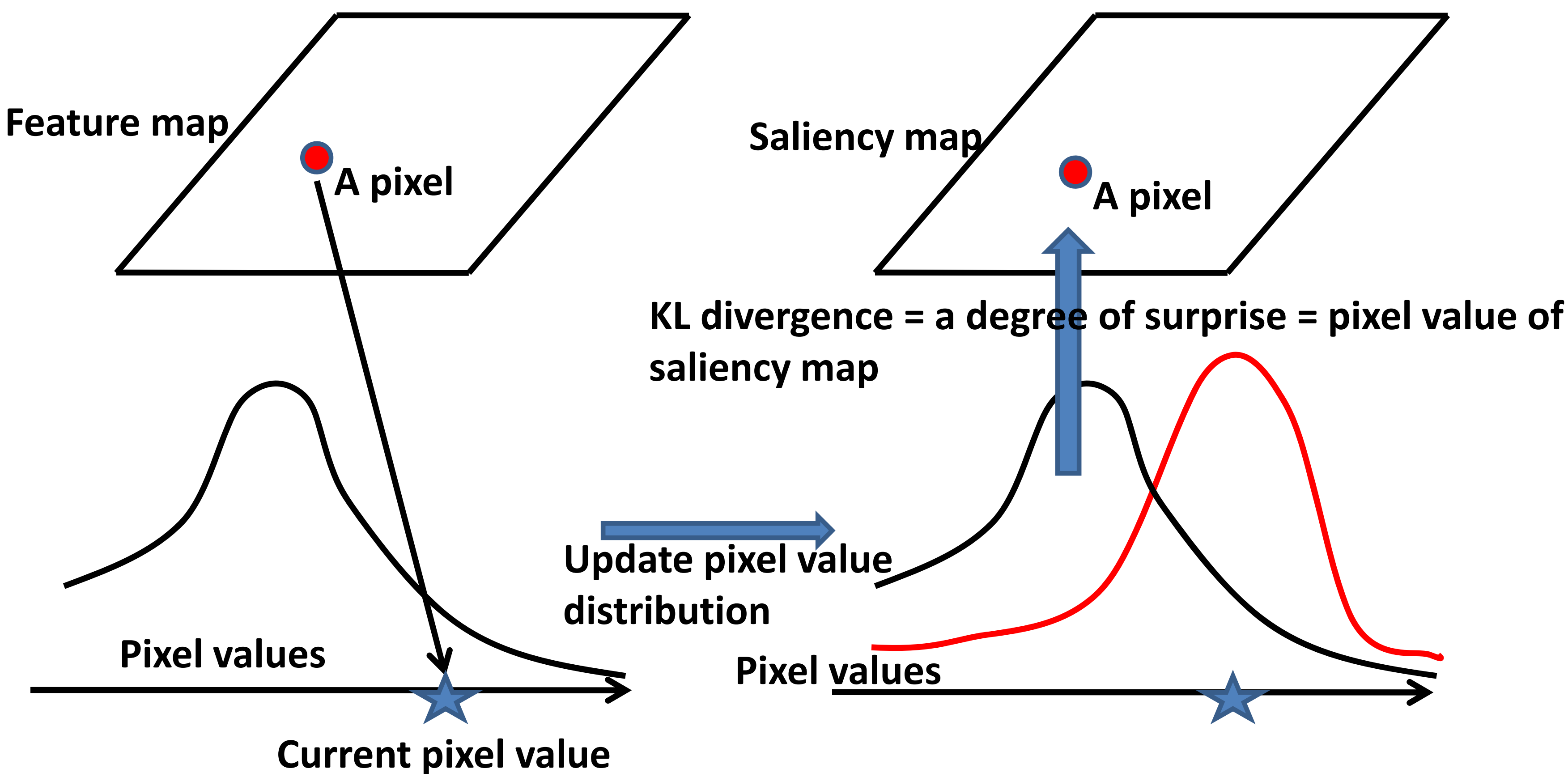
Case study 1 : Local decision method

- Step1
  - Filter responses from each DOG are added into a small size of feature map



Case study 1 : Local decision method

- Step 2: Compute a saliency map from feature maps

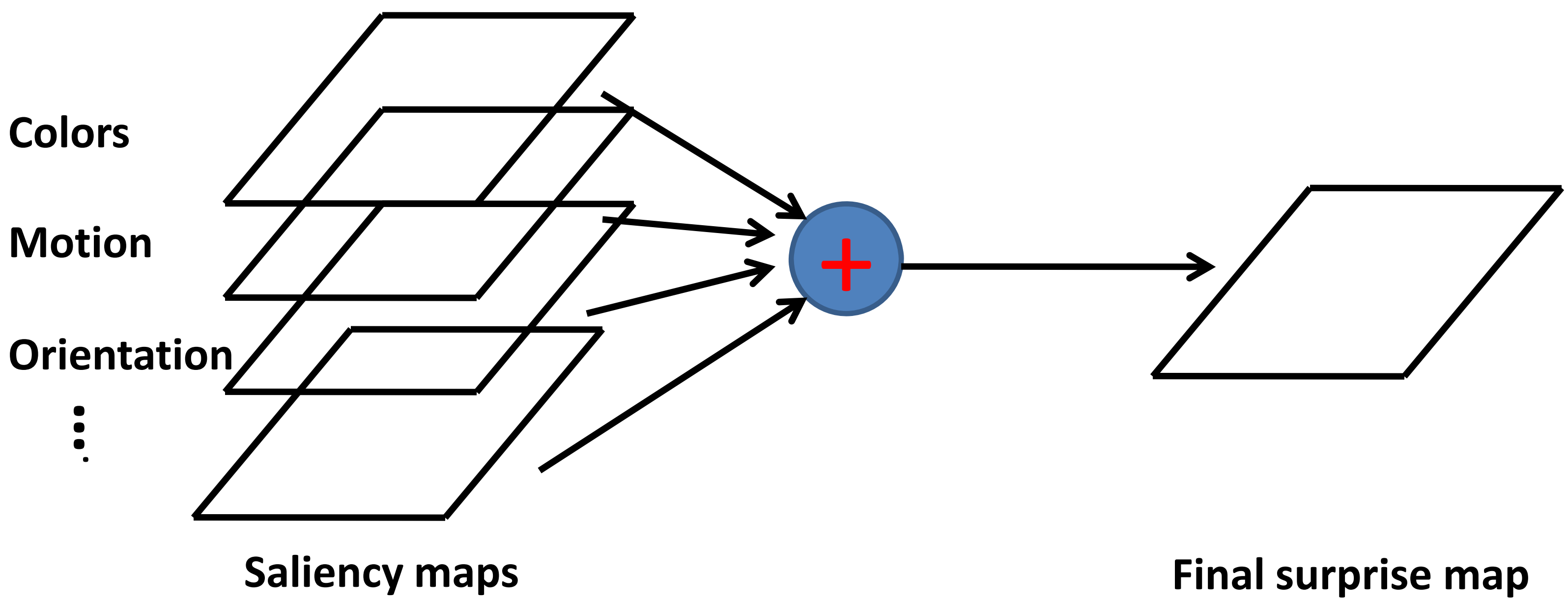


Case study 1 : Local decision method

- Step2
  - For each pixel of feature map, a saliency value is computed
  - Pixel value distribution of each pixel of feature map is modeled as Gamma distribution
  - Given newly observed pixel value, update a pdf of Gamma distribution
  - Using KL-divergence, compute a deviation between prior and posterior Gamma distribution
  - Assign a KL-divergence as saliency value

Case study 1 : Local decision method

- Step3 : Integration of saliency maps over multiple channels

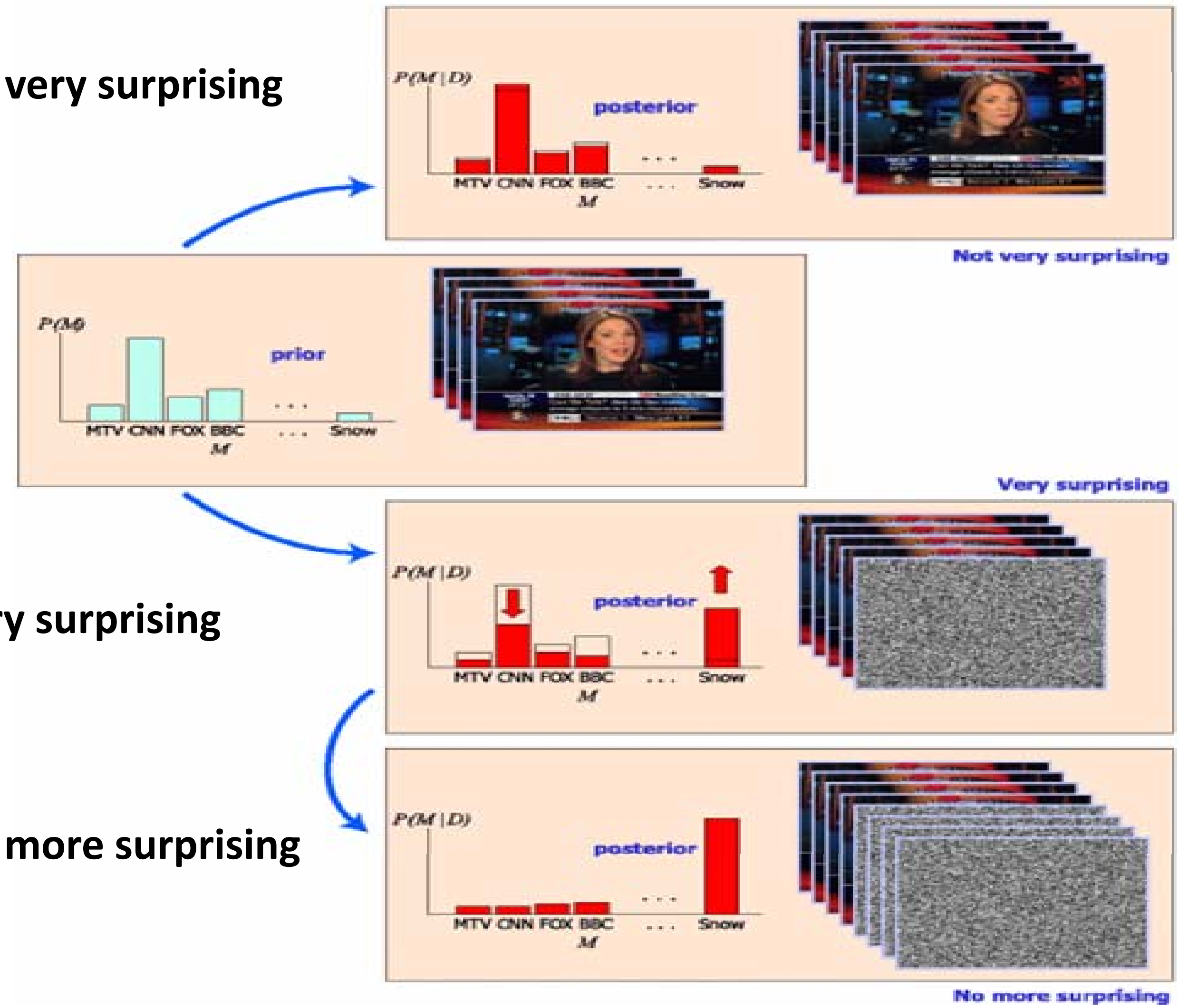


Case study 1 : Local decision method

Not very surprising

Very surprising

No more surprising



Case study 1 : Local decision method

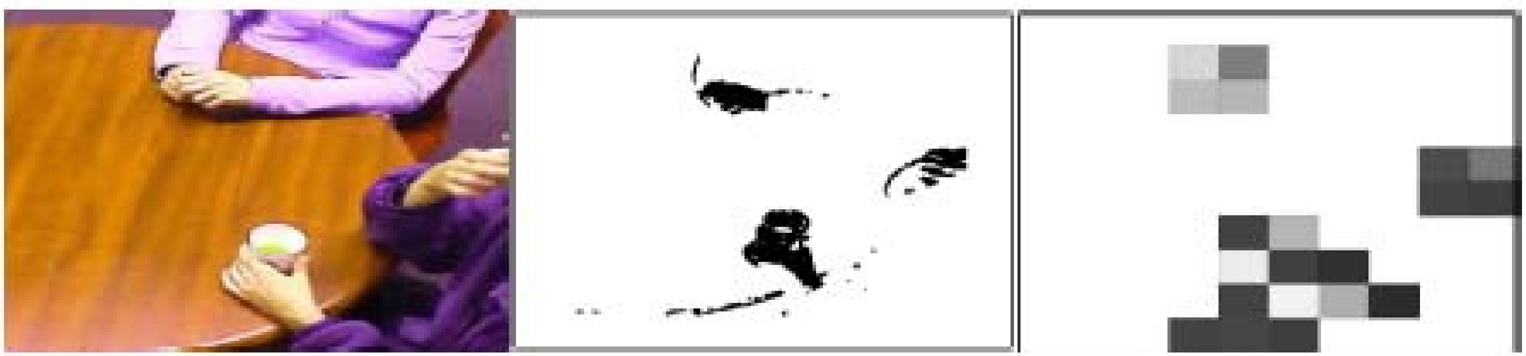
- Conclusion
  - Act as a “change” detector rather than abnormality detector
  - Forget the past very fast
    - Current observation is strongly weighted (50%) in the update of Gamma distribution
  - No experimental result on the application of abnormality detection
    - More focused on the attention problem

Case study 2: Clustering of activities

- "Detecting Unusual Activity in Video", Hua Zhong, Jianbo Shi, and Mirko Visontai, CVPR 2004
  - Find clusters of activities based on co-occurrence of local motion features
  - Clustering is performed based on segmentation using eigenvectors
  - Abnormal events are defined as activities belonging to the clusters much deviated from others

Case study 2: Clustering of activities

- Step 1: Local feature extraction
  - Intensity gradient along the temporal axis is computed for each pixel
  - Histogram is built for each image based on the magnitude of intensity gradient



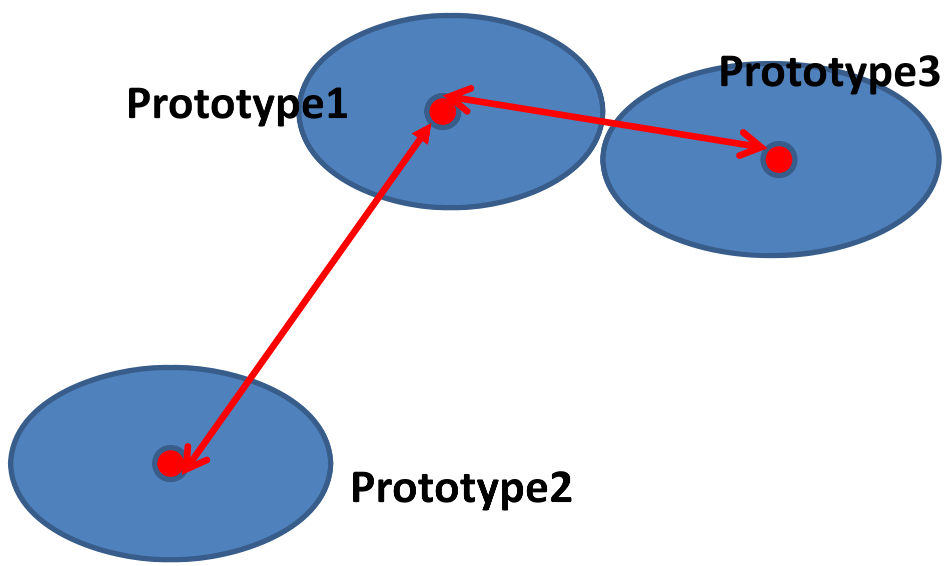
$$M(x,y,t)=\left\|\frac{\partial I(x,y,t)}{\partial t}\right\|_2$$

$$\sum M(x,y,t)$$

Summation in each sub-region

Case study 2: Clustering of activities

- Step2 : K means of histograms
  - Each Histogram is mapped to one of K prototypes
  - Compute pair-wise similarity of prototypes S(i,j) based on similarity in histograms of cluster centers



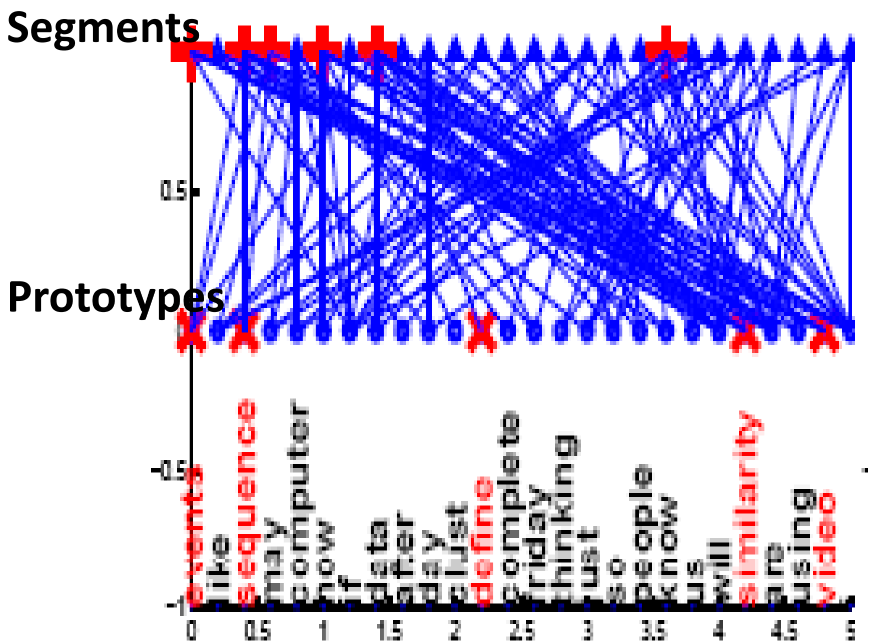
Case study 2: Clustering of activities

- Step3: Slice the video into T second long segments
  - Compute the co-occurrence matrix C between prototypes and segment

	Prototype1	Prototype2	Prototype3	Prototype4	…
Segment1	1	1	0	0	…
Segment2	0	1	1	1	…
Segment3	0	0	0	0	…
…					

Case study 2: Clustering of activities

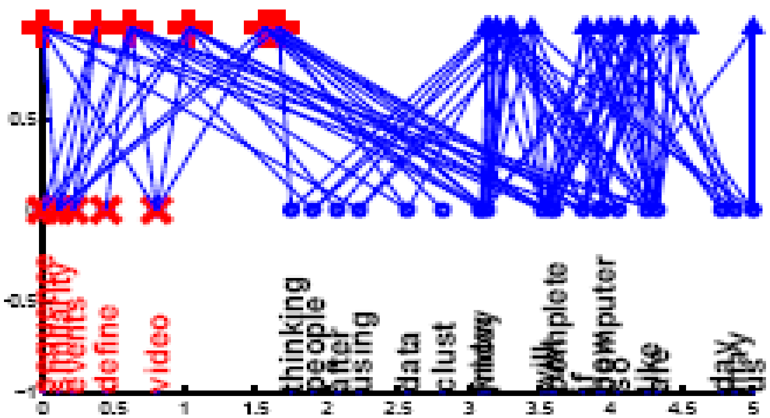
- Step4: Construct a similarity matrix with associated weight reflecting the similarities between segments and prototypes



$$
\begin{array}{l|ccc|ccc}
 & \text{Seg1} & \text{Seg2} & \ldots\ldots & \text{Prt1} & \text{Prt2} & \ldots\ldots \\
\hline
\text{Seg1} & & & & & & \\
\text{Seg2} & & \mathbf{I} & & & \mathbf{C}^{\mathbf{T}} & \\
\ldots & & & & & & \\
\hline
\text{Prt1} & & & & & & \\
\text{Prt2} & & \mathbf{C}^{\mathbf{T}} & & & \mathbf{S} & \\
\ldots & & & & & & \\
\end{array}
$$

Case study 2: Clustering of activities

- Step5: Solve generalized eigenvalue problems on the similarity matrix
  - Eigenvectors from the largest one provide coordinates of each vertex of graph
  - Vertices with similarity tends to be close each other in computed coordinates



Case study 2: Clustering of activities

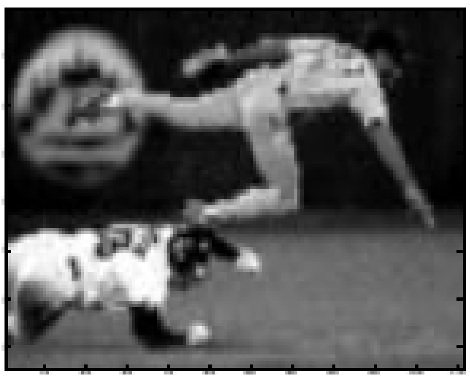
- Segmentation using eigenvector
  - Define a similarity matrix between vertices
  - Similarity matrix is denoted by W
  - Normalize W by degree matrix D (diagonal matrix)

$$\mathbf{D(i,i)} = \sum_{\mathbf{j}} \mathbf{W(i,j)},\ \mathbf{N} = \mathbf{D}^{-1/2}\mathbf{W}\mathbf{D}^{-1/2},\ \mathbf{N(i,j)} = \mathbf{W(i,j)}/\sqrt{\mathbf{D(i,i)D(j,j)}}$$

  - Construct a n by m matrix V whose columns are the first m eigenvectors of N
  - The i_{th} row of V provides a new coordinate of i_{th} vertex in the m dimensional space
    - Similar vertices get closer in the m dimensional space

Case study 2: Clustering of activities

- Segmentation using eigenvector



Input image

Define a similarity W of each pair of pixels based on intensity, position, etc

Solve the eigenvector problem on N and get V

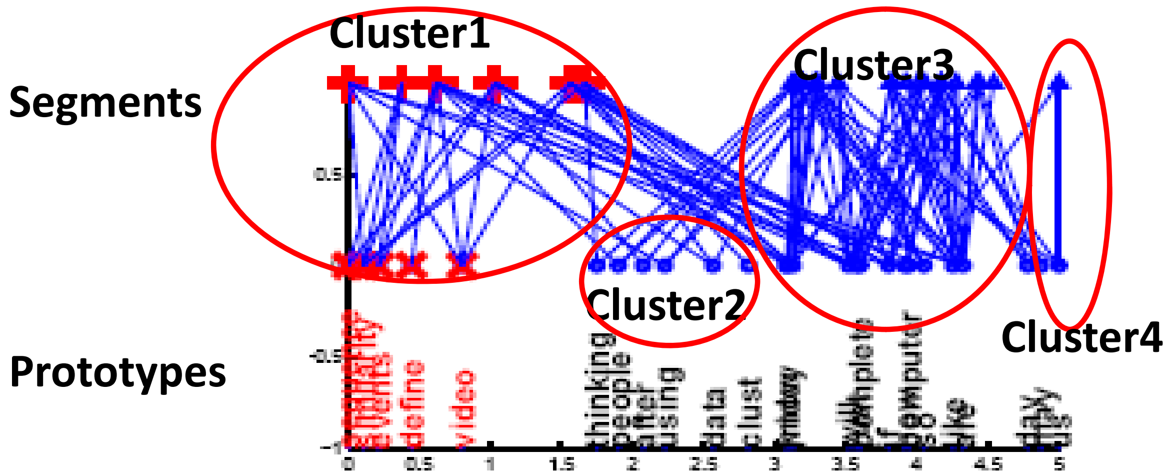
A row of $\mathbf{Q} = \mathbf{V}\mathbf{V}^{\mathbf{T}}$

Different row of $\mathbf{Q} = \mathbf{V}\mathbf{V}^{\mathbf{T}}$

Q(i,j) gives us a correlation between pixel i and j in the k-dimensional space

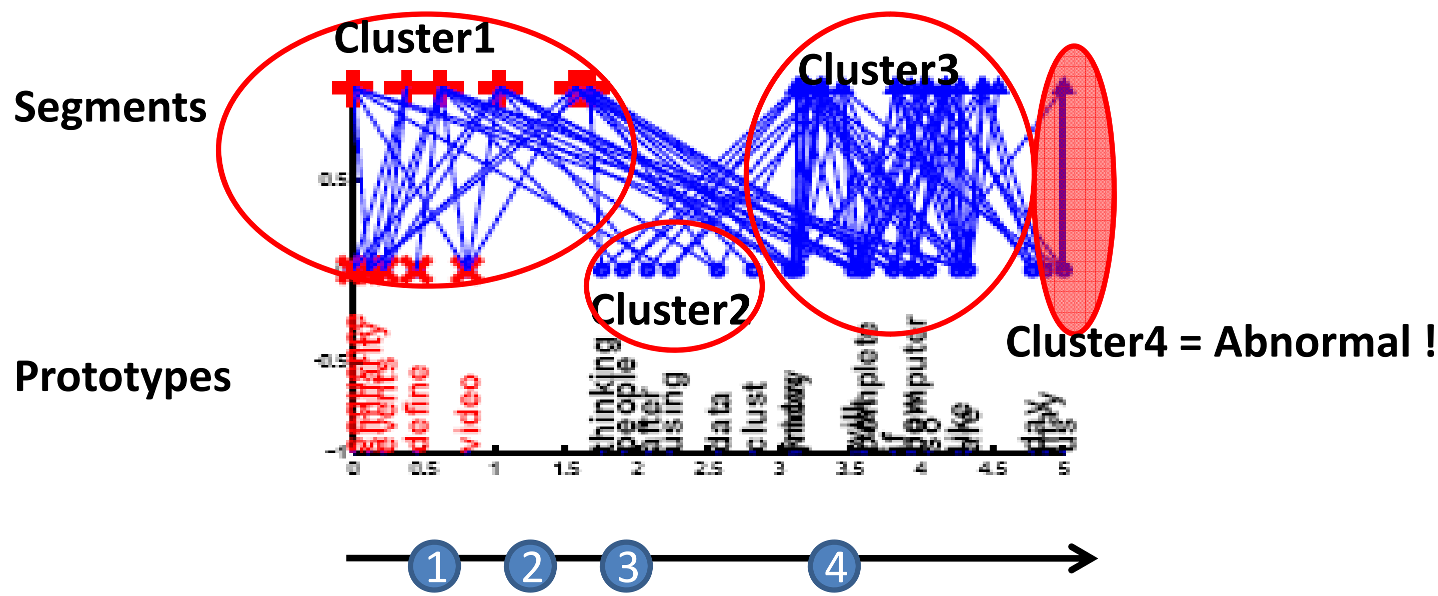
Case study 2: Clustering of activities

- Step6: Clustering of video segments and prototypes in the m dimensional space using K means



Case study 2: Clustering of activities

- Step7: Detect abnormal video segment by computing inter-cluster distance
  - A cluster having large inter-cluster distance is flagged as being abnormal



Case study 2: Clustering of activities

- Experimental result

A1 B1 C1 D1 E1

A2 B2 C2 D2 E2

Detected cheating (A-C) Non-detecting False alarm

Case study 2: Clustering of activities

- Conclusion
  - Simple computation in clustering video segment
    - But arbitrary in defining the number of clusters in m-dimensional space
    - Also, it is unclear how to choose the number of eigenvectors, m.
  - Hard to be applied to online application

Case study3 : Learning based activity clustering

- “Video Behavior Profiling and Abnormality Detection without Manual Labelling,” Tao Xiang and Shaogang Gong, ICCV05
  - HMM based training of each video segment
  - Defining similarity between segments by comparing HMM networks of each segment
  - Clustering video segments with automatic selection of number of clusters

Case study3 : Learning based activity clustering

- Step1 : Slice the video into segments and detect local features through the video
  - Foreground pixel detector + Connected component → Blob of foreground pixels
  - Seven dimensional blob feature vector

$$v = \{\bar{x}, \bar{y}, w, h, R, Mx, My\}$$

Case study3 : Learning based activity clustering

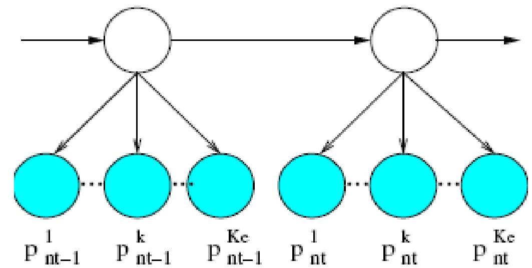
- Step2: Clustering of Blob features into K_e classes
  - Gaussian Mixture model with automatic model order selection based on Bayesian Information Criterion(BIC)
  - Feature vector of video segment V_n with T_n frames

$$P_n = \{ p_{n1}, \ldots, p_{nt}, \ldots, p_{nT_n} \}$$

$$p_{nt} = \{ p_{nt}^1, \ldots, p_{nt}^k, \ldots, p_{nt}^{k_e} \}$$

Case study3 : Learning based activity clustering

- Step3: Training of HMM for each video segment
  - For N segments, N HMMs are trained



    - Each HMM has K_e states (arbitrary)
    - Observation : video segment feature vector P_n
    - Parameters of HMM : transition probability, conditional pdf of observation given a state
  - Output of training : Parameters of HMM
    - A kind of EM algorithm (called Baum-Welch) is used to iteratively optimize joint probability of states and optimal parameters

Case study3 :
Learning based activity clustering

- Step4: Compute similarity between video segments based on trained HMM

$$S(i,j)=\frac{1}{2}\left\{\frac{1}{T_j}\log\Pr(P_j\mid B_i)+\frac{1}{T_i}\log\Pr(P_i\mid B_j)\right\}$$

$\Pr(P_j \mid B_i)$ Likelihood of video segment V_j given a HMM trained on segment V_i

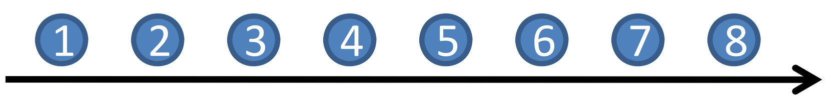
Case study3 : Learning based activity clustering

- Step5: Assign a k-dimensional coordinate to each video segment based on segmentation using eigenvectors of normalized similarity matrix
  - Use the same technique as the one in case study 2
  - But, number of eigenvectors, k, is automatically chosen

Case study3 :
Learning based activity clustering

- How to select the number of eigenvectors
  - i th element of j th eigenvector is a j th coordinate of i th vertex
  - The values of eigenvector's each element should be tightly clustered to have a discriminating power

Meaningful eigenvector



Useless eigenvector

Case study3 : Learning based activity clustering

- How to select the number of eigenvectors
  - Select eigenvectors with desirable property above mentioned

$$P(e_{kn}|\theta_{e_{kn}}) = (1 - R_{\mathbf{e_k}})P(e_{kn}|\theta^1_{e_{kn}}) + R_{\mathbf{e_k}}P(e_{kn}|\theta^2_{e_{kn}})$$

Single-mode Gaussian   Two-modes Gaussian

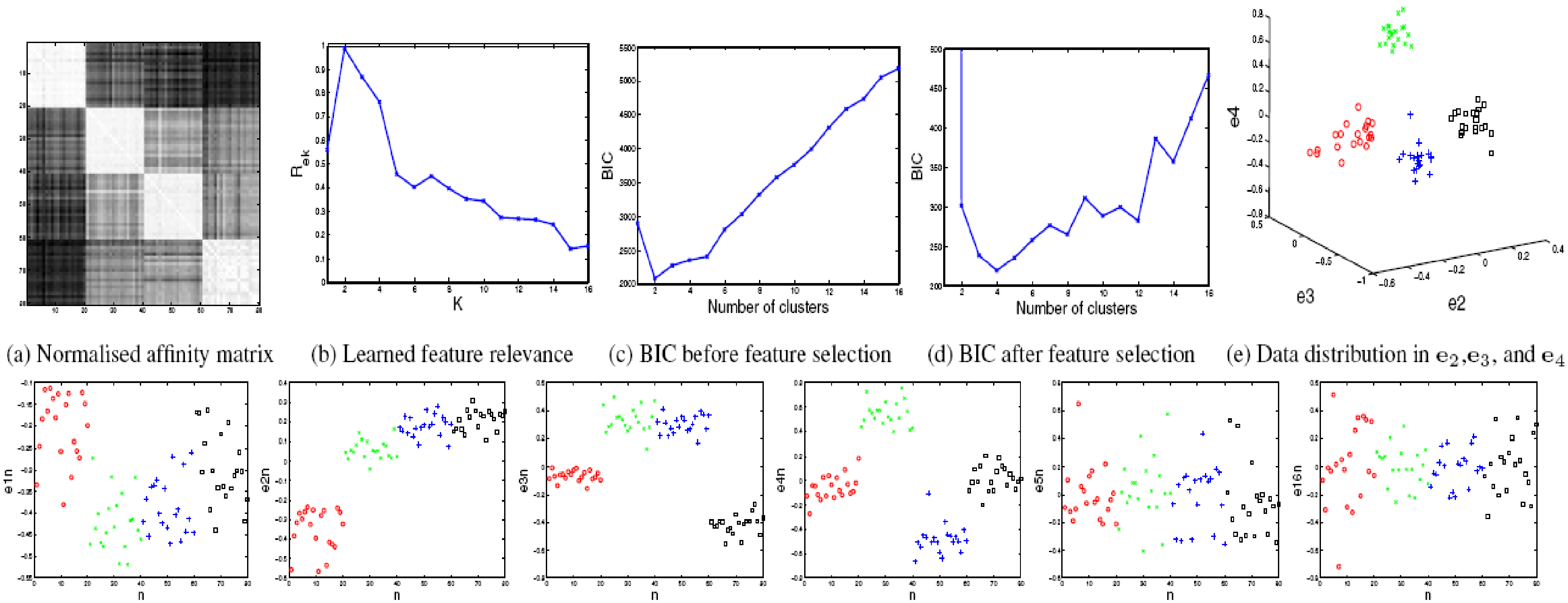
$$P(e_{kn}|\theta^1_{e_{kn}}) = \prod_{n=1}^{N} \frac{1}{\sqrt{2\pi}\sigma_{k1}} \exp\left[-\frac{1}{2}\left(\frac{e_{kn} - \mu_{k1}}{\sigma_{k1}}\right)^2\right]$$

$$P(e_{kn}|\theta^2_{e_{kn}}) = \prod_{n=1}^{N} \left(\begin{array}{c} \frac{w_k}{\sqrt{2\pi}\sigma_{k2}} \exp\left[-\frac{1}{2}\left(\frac{e_{kn}-\mu_{k2}}{\sigma_{k2}}\right)^2\right] \\ + \frac{1-w_k}{\sqrt{2\pi}\sigma_{k3}} \exp\left[-\frac{1}{2}\left(\frac{e_{kn}-\mu_{k3}}{\sigma_{k3}}\right)^2\right] \end{array} \right)$$

  - $R_{e_k} > 0.5$: Two modes Gaussian is more fit to a given eigenvector = Given vector is meaningful

Case study3 : Learning based activity clustering

- How to select the number of eigenvectors



(a) Normalised affinity matrix (b) Learned feature relevance (c) BIC before feature selection (d) BIC after feature selection (e) Data distribution in e_2,e_3, and e_4

(f) e_1 (g) e_2 (h) e_3 (i) e_4 (j) e_5 (k) e_{16}

Case study3 :
Learning based activity clustering

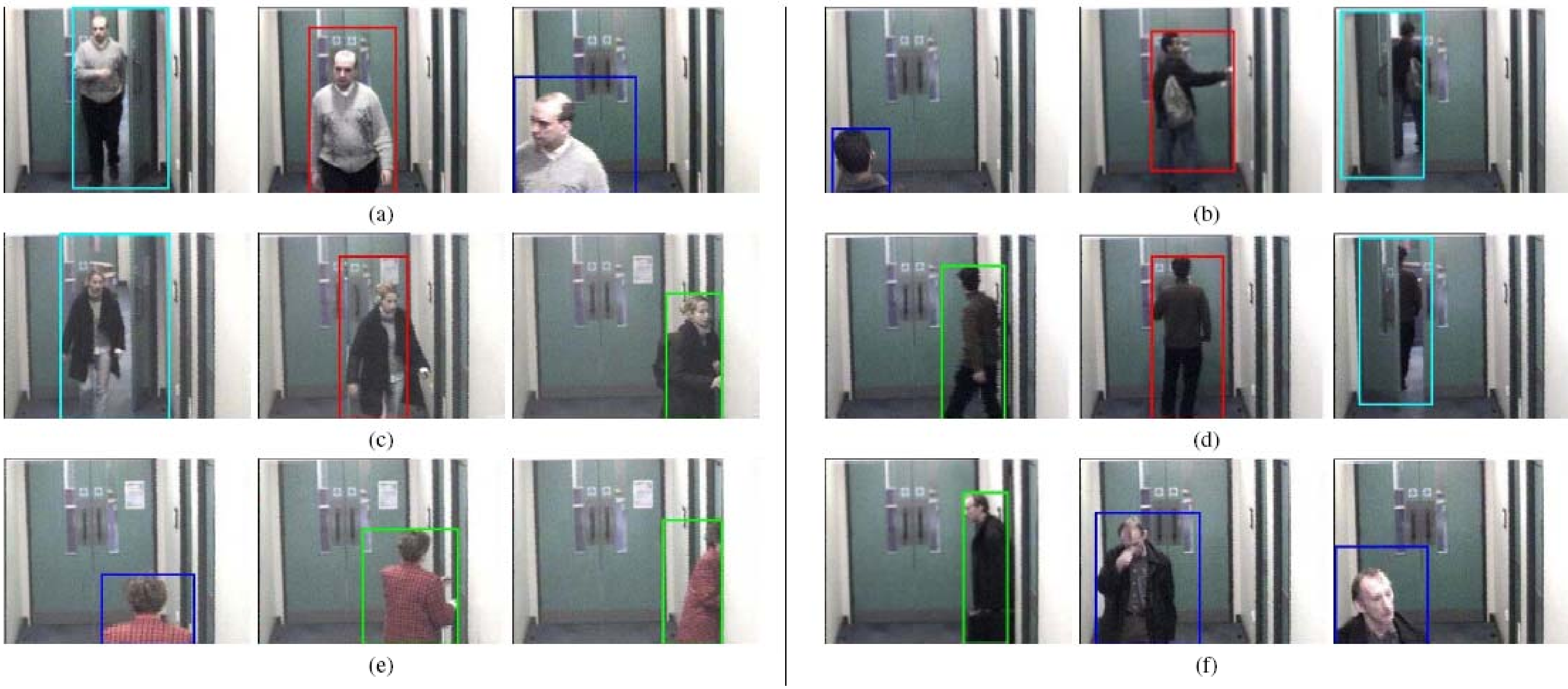
- Step6: Clustering of video segments in k-dimensional space
  - Use a Gaussian Mixture Model with automatic selection of the number of components

Case study3 : Learning based activity clustering

- Step7: Detecting anomaly
  - Re-training of HMMs for each clusters
    - Using all video segments belonging to a given cluster
  - For a new video segment, compute likelihoods for each HMMs $P(\mathbf{P}|\mathbf{M}) = \sum_{k=1}^{K} \frac{N_k}{N} P(\mathbf{P}|\mathbf{B}_k)$
  - If $P(\mathbf{P}|\mathbf{M}) < Th_A$, flag abnormality
  - Otherwise, classify the video segment into a ML cluster $\hat{k} = \arg\max_k \{P(\mathbf{P}|\mathbf{B}_k)\}$

Case study3 : Learning based activity clustering

- Result – Typical activities



Case study3 : Learning based activity clustering

- Result – Abnormal activities

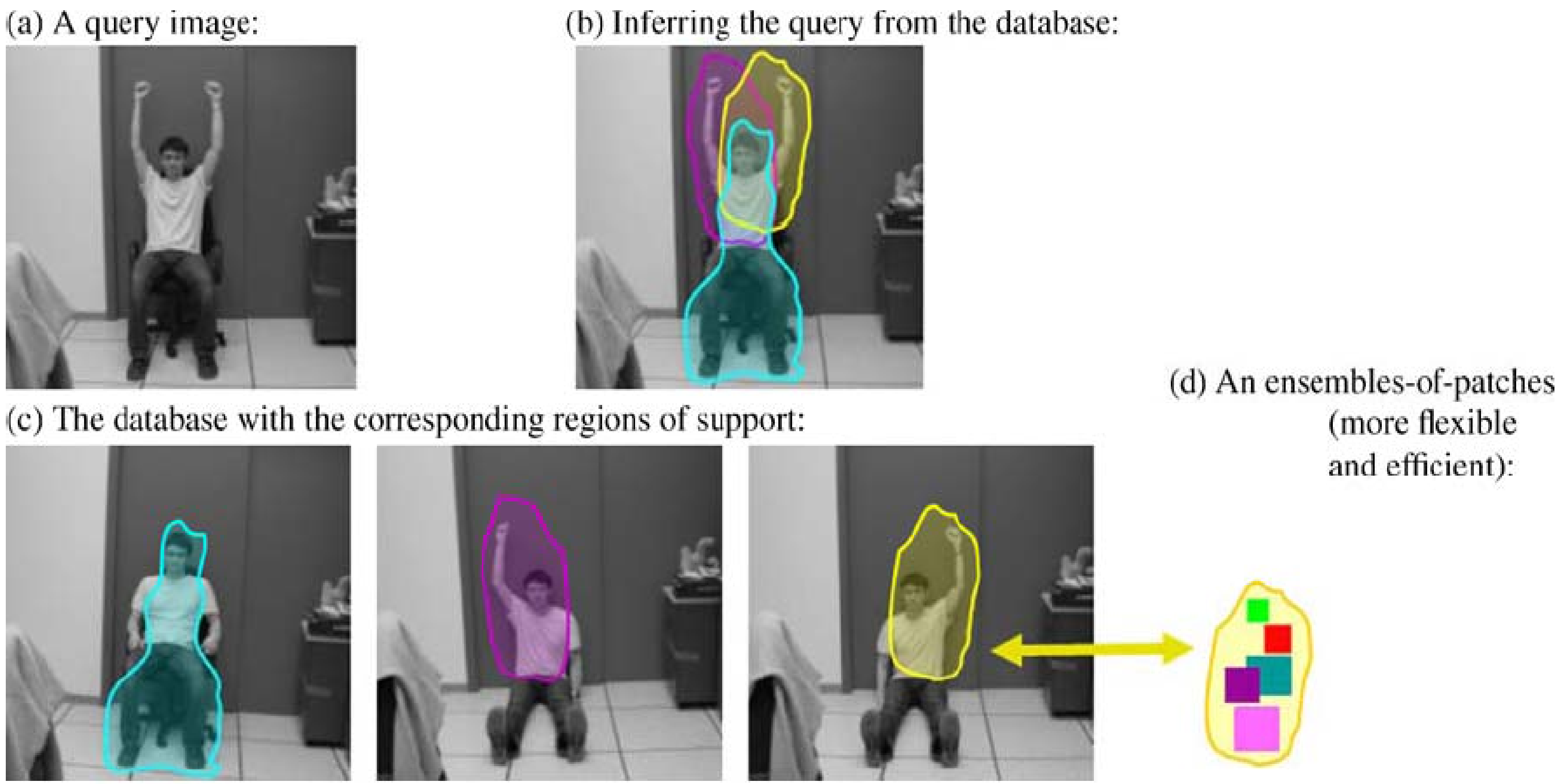
Case study3 : Learning based activity clustering

- Conclusion
  - Propose more advanced technique to cluster activities
    - Automatic selection of the number of clusters
    - Allow variable length of segments by adopting distance measure based on HMM
  - Sensitive to training dataset
    - HMM tends to be over-fitting to the training data
    - Local minimum of estimation of HMM parameters
  - Inadequate to online applications
    - Updating HMMs is computationally expensive
  - Cannot localize the abnormal event
    - Drawback of segment-based approach

Case study 4 : Search-based method

- "Detecting Irregularities in Images and Video," ICCV05, IJCV07
  - For every and each pixel, find a corresponding region in the database

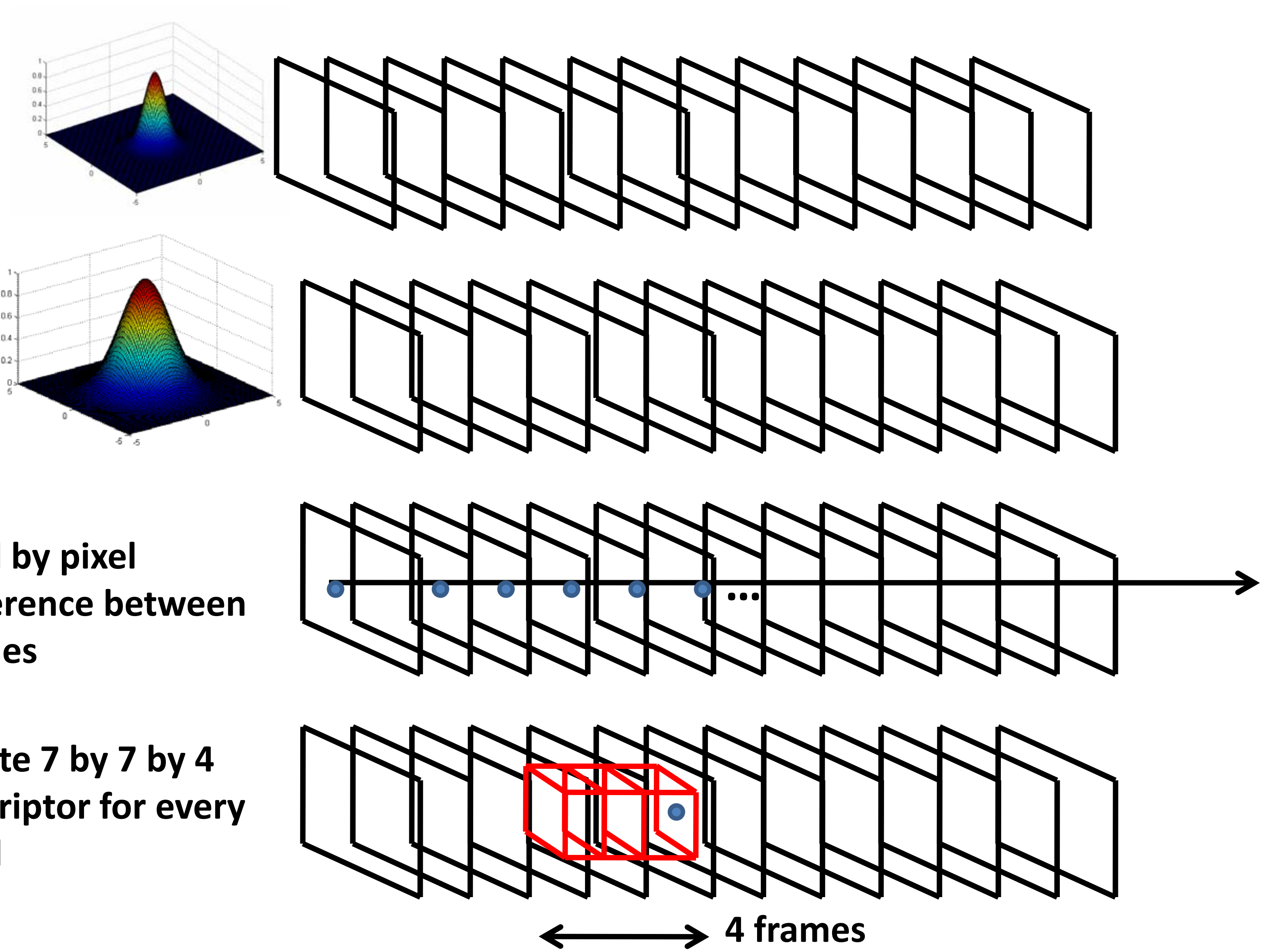
Case study 4 : Search-based method



Case study 4 : Search-based method

- Step1: Create patch descriptor for every pixel in the images
  - Apply Gaussian filter with several scales along the spatial-temporal axis
  - For each scale, compute temporal derivatives
  - For every pixel, 7 by 7 by 4 descriptor is created over multiple scales

Case study 4 : Search-based method



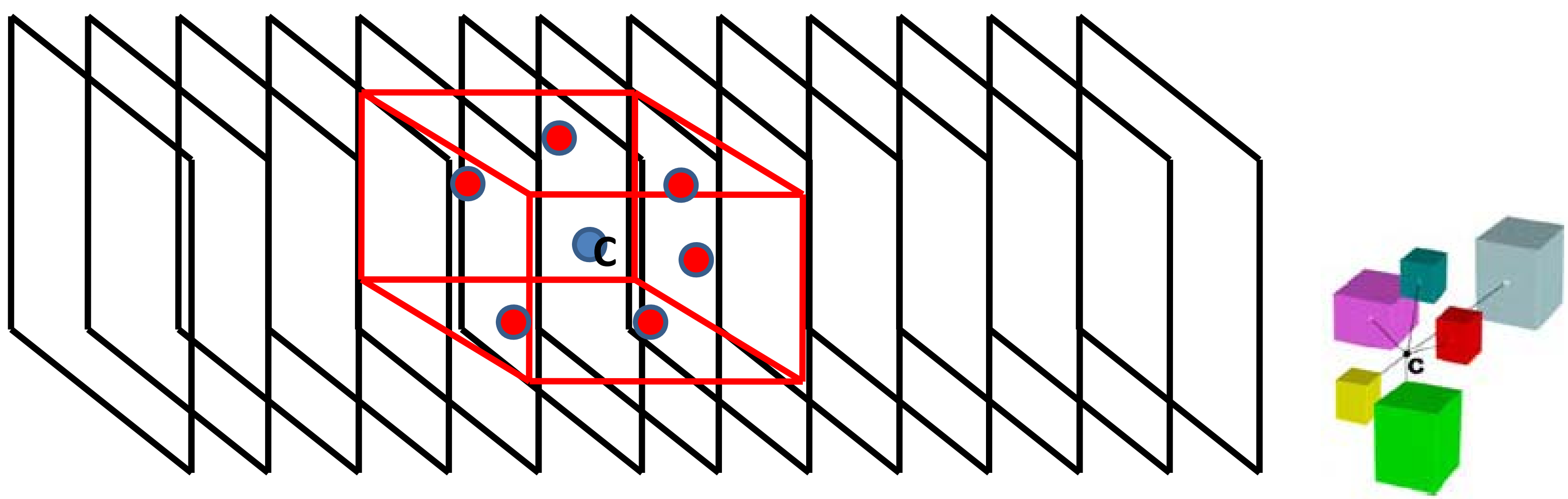
Pixel by pixel Difference between frames

Create 7 by 7 by 4 descriptor for every pixel

Case study 4 : Search-based method

- Step2: Create an ensemble of patches for every pixel
  - Sample hundreds of points in the 50 by 50 by 50 windows surrounding a given pixel
  - Randomly pick a scale of each sampled point
  - An Ensemble of a pixel consists of hundreds of patches of different scales

Case study 4 : Search-based method

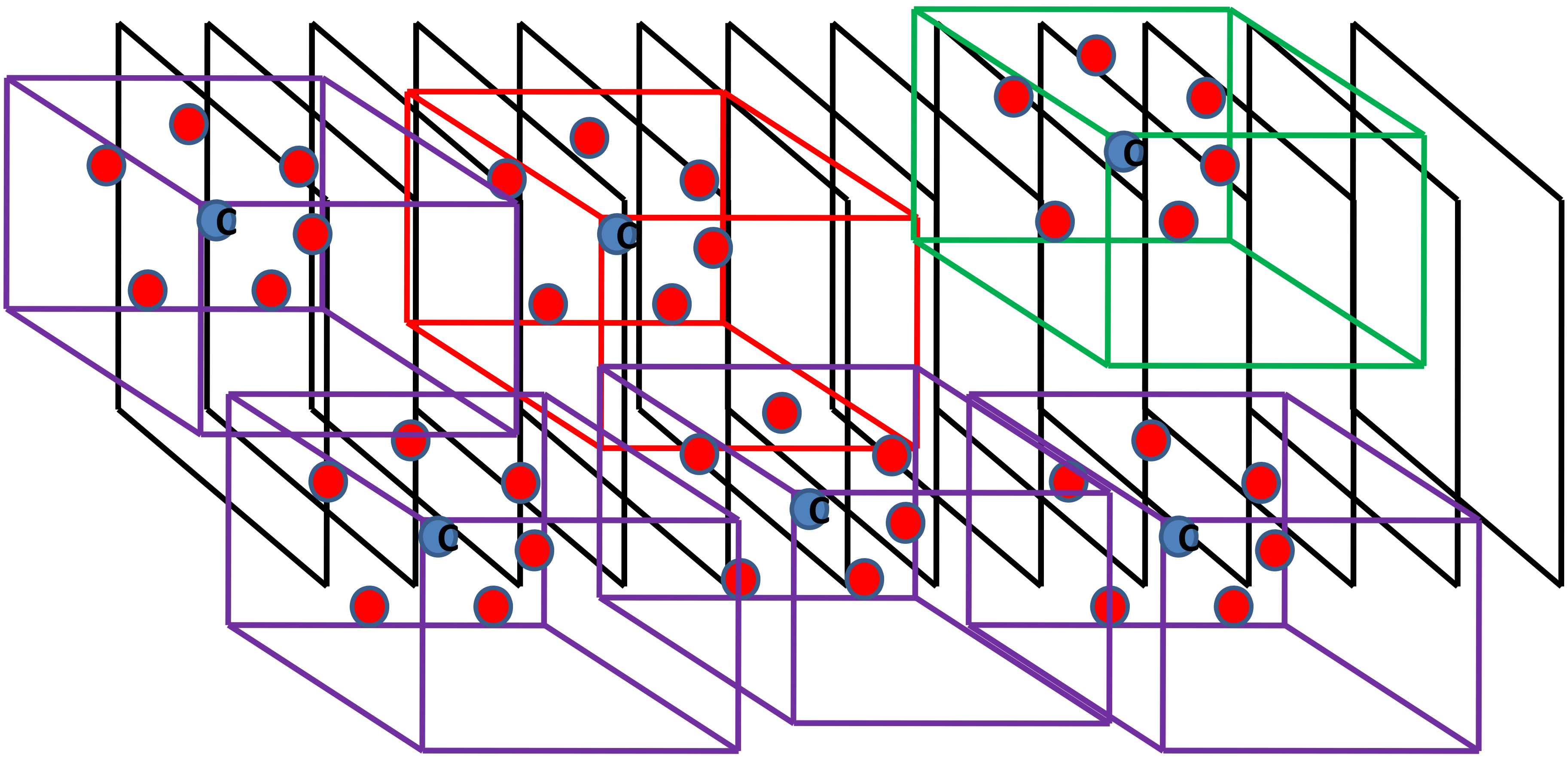


50 by 50 by 50 size of ensemble and sampled points(i.e patches) in an ensemble

Case study 4 : Search-based method

- Step3: Search similar ensembles through the database
  – Based on pre-defined probabilistic model of ensemble variation, find the most similar(most likelihood) ensemble to a given query ensemble

Case study 4 : Search-based method



Full search of database for a given query ensemble

Case study 4 : Search-based method

- Probabilistic Model of ensemble variation
  - Allow some variations of patch locations and patch descriptors in an ensemble

$$P\left(c_x, d_x^1, \ldots, l_x^1, \ldots, c_y, d_y^1, \ldots, l_y^1\right) = \alpha \prod_i P\left(l_y^i \mid l_x^i, c_x, c_y\right) P\left(d_y^i \mid d_x^i\right) P\left(d_x^i \mid l_x^i\right)$$

y: Query
x: Database

$$P\left(d_y^i \,\middle|\, d_x^i\right) = \alpha_1 \exp\left(-\frac{1}{2}(d_y^i - d_x^i)^T S_D^{-1} (d_y^i - d_x^i)\right)$$

Descriptor variation

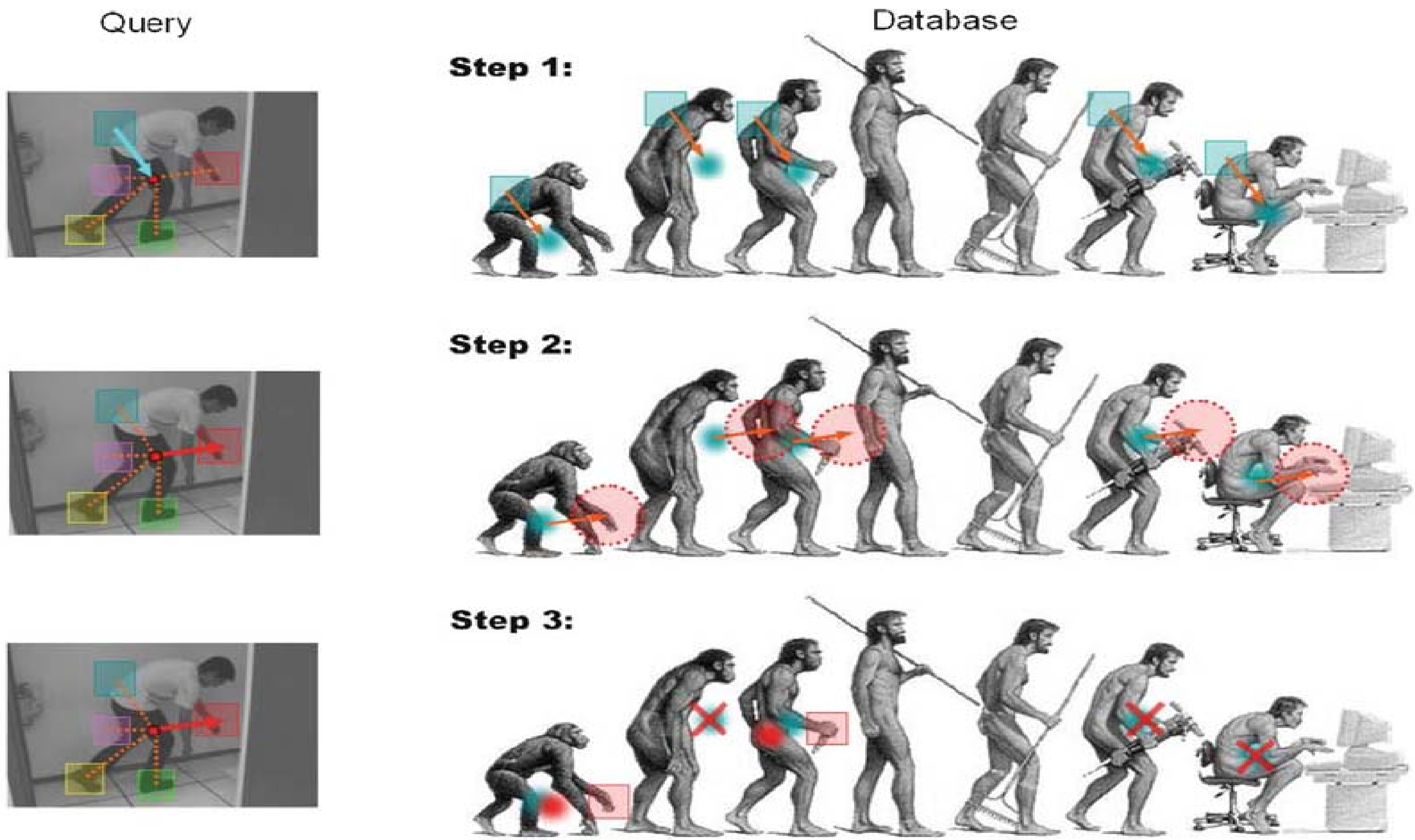
$$P\left(l_y^i \,\middle|\, l_x^i, c_x, c_y\right) = \alpha_2 \cdot \exp\left(-\frac{1}{2}((l_y^i - c_y) - (l_x^i - c_x))^T \times S_L^{-1}((l_y^i - c_y) - (l_x^i - c_x))\right)$$

Relative location variation

Case study 4 : Search-based method

- Speed up the search : Progressive elimination
  - For the first patch, find the best c patches in the database
  - Guess the candidate center locations Cx in the c images that have the best c patches
  - From the guess Cx, determine a region where the second patch can exist
  - Search the similar patches to the second patch in the given region
    - If similarity is below the threshold, stop the search for that image
  - Repeat the guess of Cx location based on the second patch comparison result

Case study 4 : Search-based method



Case study 4 : Search-based method

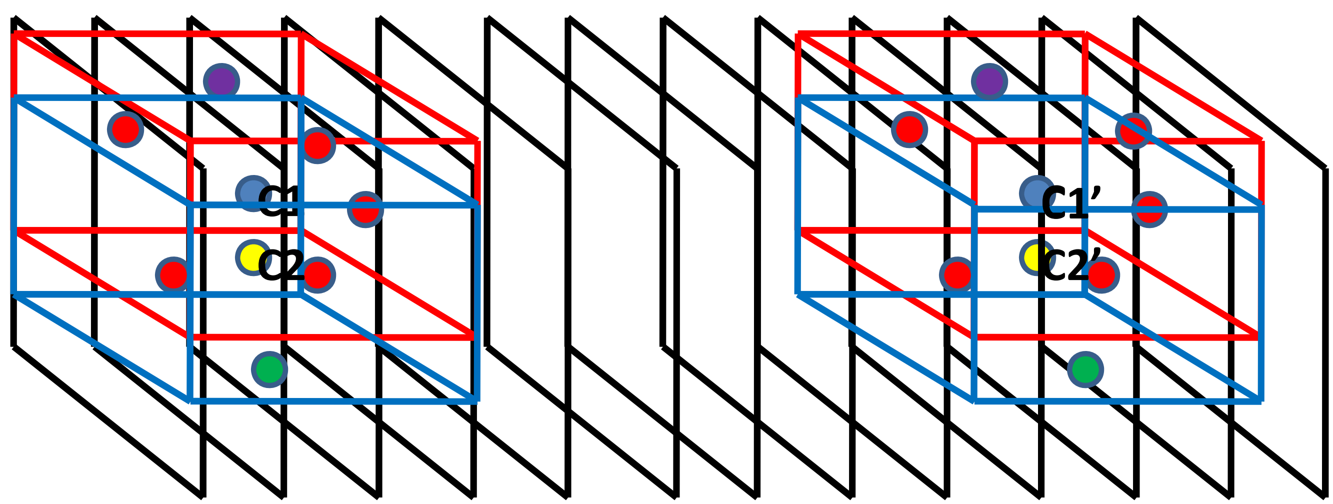
- Speed up the search : Multi-scale search
  - As the first patch to be searched, pick the patch belonging to the largest scale
  - Reduce the risk of early false decision
  - Reduce the number of initial search

Case study 4 : Search-based method

- Speed up the search : Use of hash or KD-tree
  - Vector quantization of descriptors
  - Cluster the descriptors using hash table or KD-tree

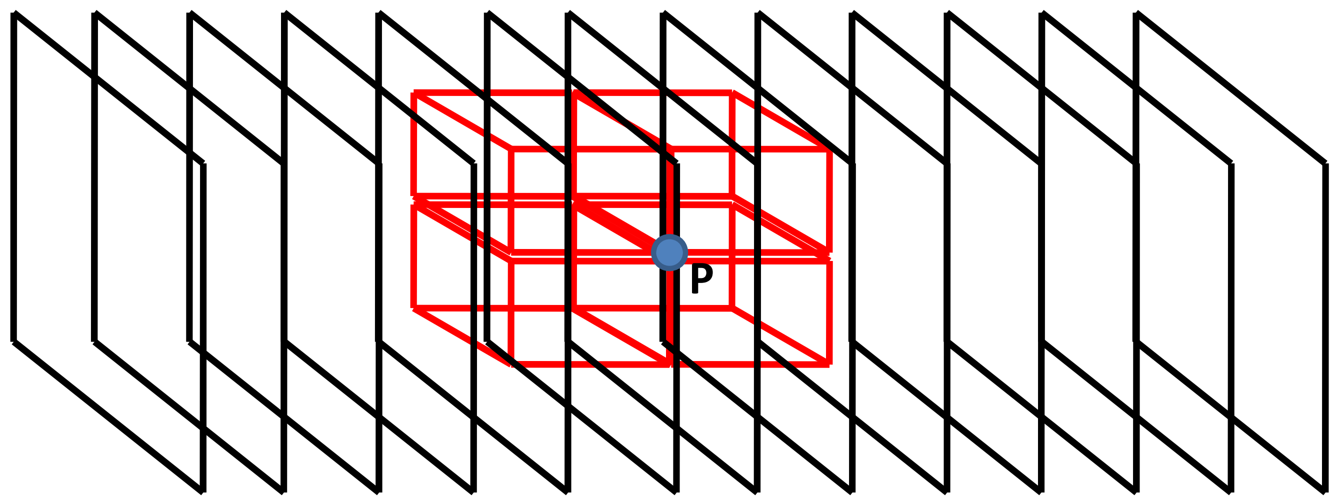
Case study 4 : Search-based method

- Speed up the search: Predictive search
  - For query points in the neighborhood, the matched patch is highly likely to be located in the similar position in the database



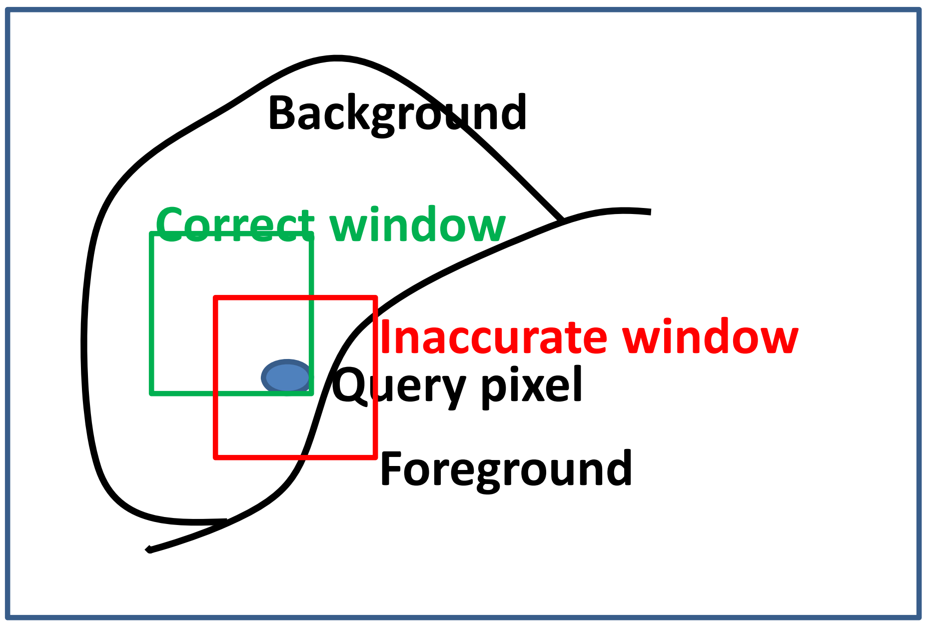
Case study 4 : Search-based method

- Step4: Determining an abnormality – Shifted and variable sized window technique
  – Likelihood of a pixel p $l(p) = \max_{i \in shifted\ neighbor(p)} \Pr(i)$



Case study 4 : Search-based method

- Shifted window
  - Easy way to handle occlusion problem



Case study 4 : Search-based method

- Variable sized windows
  - If low likelihood is obtained at the trial with large size of initial window (e.g. 50 by 50 by 50), retry a search with smaller size of window
  - But, penalty is imposed on the smaller size window
  - Finally, if likelihood is below the threshold, flag an abnormality for that pixel

Case study 4 : Search-based method

- Conclusion
  - Accurate localization of abnormal event
  - Robustly perform independent of the kind of scenes
  - Search time is too long
    - Online application will not be possible
  - Operate in a local manner
    - Cannot deal with co-occurrence of activities or temporal ordering of long sequences of activities
  - Operate in a translation invariant manner
    - Good or bad of this property depends on applications

Conclusion

- Local decision
  - Computationally efficient
  - Easily adaptive to the temporal evolution of activities
  - Many of false alarms : act like a detector of scene change
  - Can be used as pre-processing routine of abnormality detection

Conclusion

- Learning-based decision
  - Based on clustering of normal activities
  - Statistical outliers are regarded as abnormal events
  - Ordering and co-occurrence of actions are handled in a principled way
  - Mainly focused on activities of a single individual
    - Interaction handling could make the number of states in HMM infeasible
  - Hard to adapt to the evolution of observations over a long time
  - Scene sensitive

Conclusion

- Search-based decision
  - Intuitively simple to understand
  - Accurate localization of abnormal event
  - Less false alarms than local decision, but computationally expensive
  - Suffer from occlusion
  - Unclear how to handle co-occurrence of activities
    - Although some activities have been seen in the database, their co-occurrence may be able to be abnormal