

Change Detection

Main Points

- Detect pixels which are changing due to motion of objects.
- Not necessarily measure motion (optical flow), only detect motion.
- A set of connected pixels which are changing may correspond to moving object.

Picture Difference

$$D_i(x, y) = \begin{cases} 1 & \text{if } DP(x, y) > T \\ 0 & \text{.....otherwise} \end{cases}$$

$$DP(x, y) = |f_i(x, y) - f_{i-1}(x, y)|$$

$$DP(x, y) = \sum_{i=-m}^m \sum_{j=-m}^m |f_i(x+i, y+j) - f_{i-1}(x+i, y+j)|$$

$$DP(x, y) = \sum_{i=-m}^m \sum_{j=-m}^m \sum_{k=-m}^m |f_i(x+i, y+j) - f_{i+k}(x+i, y+j)|$$

Background Image

- The first image of a sequence without any moving objects, is background image.
- Median filter

$$B(x, y) = \text{median}(f_1(x, y), \dots, f_n(x, y))$$

PFINDER

Pentland

Pfinder

- Segment a human from an arbitrary complex background.
- It only works for single person situations.
- All approaches based on background modeling work only for fixed cameras.

Algorithm

- **Learn** background model by watching 30 second video
- **Detect** moving object by measuring deviations from background model
- **Segment** moving blob into smaller blobs by minimizing covariance of a blob
- **Predict** position of a blob in the next frame using Kalman filter
- **Assign** each pixel in the new frame to a class with max likelihood.
- **Update** background and blob statistics

Learning Background Image

- Each pixel in the background has associated mean color value and a covariance matrix.
- The color distribution for each pixel is described by Gaussian.
- YUV color space is used.

Detecting Moving Objects

- After background model has been learned, Pfinder watches for large deviations from the model.
- Deviations are measured in terms of Mahalanobis distance in color.
- If the distance is sufficient then the process of building a blob model is started.

Detecting Moving Objects

- For each of k blob in the image, log-likelihood is computed

$$d_k = -.5(y - \mathbf{m}_k)^T K_k^{-1} (y - \mathbf{m}_k) - .5 \ln |K_k| - .5m \ln(2\hat{\lambda})$$

- Log likelihood values are used to classify pixels

$$s(x, y) = \arg \max_k (d_k(x, y))$$

Updating

- The statistical model for the background is updated.

$$K^t = E[(y - \mathbf{m}^t)(y - \mathbf{m}^t)^T]$$

$$\mathbf{m}^t = (1 - \mathbf{a})\mathbf{m}^{t-1} + \mathbf{a}y$$

- The statistics of each blob (mean and covariance) are re-computed.

Mixture of Gaussians

Grimson

Algorithm

- **Learn** background model by watching 30 second video
- **Detect** moving object by measuring deviations from background model, and applying connected component to foreground pixels.
- **Predict** position of a region in the next frame using Kalman filter
- **Update** background and blob statistics

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.

$$\Pr(X_t) = \sum_{j=1}^K \frac{w_j}{(2\mathbf{p})^{\frac{m}{2}} |\Sigma_j|^{\frac{1}{2}}} e^{-\frac{1}{2}(X_t - \mathbf{m}_j)^T \Sigma_j^{-1} (X_t - \mathbf{m}_j)}$$

Mixture of Gaussians

- The K distributions are stored in descending order of the term

$$\frac{\mathbf{w}_j}{\mathbf{S}_j}$$

- Out of “k” distributions, the first B are selected

$$B = \arg \min_b \left[\frac{\sum_{j=1}^b \mathbf{w}_j}{\sum_{j=1}^K \mathbf{w}_j} > T \right]$$

Learning Background Model

- Every new pixel is checked against all existing distributions. The match is the first distribution such that the pixel value lies within 2 standard deviations of mean.
- If no match, introduce new distribution.

Updating

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

$$\mathbf{m}_{j,t} = (1 - \mathbf{r})\mathbf{m}_{j,t-1} + \mathbf{r}X_t$$

$$\mathbf{s}_{j,t} = (1 - \mathbf{r})\mathbf{s}_{j,t-1}^2 + \mathbf{r}(X_t - \mathbf{m}_{j,t})^T (X_t - \mathbf{m}_{j,t})$$

- The weights are adjusted:

$$\mathbf{w}_{j,t} = (1 - \mathbf{a})\mathbf{w}_{j,t-1} + \mathbf{a}(M_{j,t})$$

Segmenting Background

- Any pixel that is more than 2 sd from all the distributions is marked as a part of foreground-moving object.
- Such pixels are then clustered into connected components.

Kanade

Summary

- Very similar to k-Gaussian with following differences:
 - uses only single Gaussian
 - uses gray level images, the mean and variance are scalar values

Algorithm

- **Learn** background model by watching 30 second video
- **Detect** moving object by measuring deviations from background model, and applying connected component to foreground pixels.
- **Update** background and region statistics

Detection

- During detection if intensity value is more than two sigma away from the background it is considered foreground:
 - keep original mean and variance
 - track the object with new mean and variance
 - if new mean and variance persists for sometime, then substitute the new mean and variance as the background model
 - If object is no longer visible, it is incorporated as part of background

W4 (Who, When, Where, What)

Davis

W4

- Compute “minimum”(M(x)), “maximum” (N(x)), and “largest absolute difference” (L(x)).

$$D_i(x, y) = \left\{ \begin{array}{l} 1 \quad \text{if } |M(x, y) - f_i(x, y)| > L(x, y) \text{ or} \\ \quad |N(x, y) - f_i(x, y)| > L(x, y) \\ 0 \quad \dots \text{ otherwise} \end{array} \right\}$$

- Theoretically, the performance of this tracker should be worse than others.
- Even if one value is far away from the mean, then that value will result in an abnormally high value of L .
- Having short training time is better for this tracker.

Limitations

- Multiple people
- Occlusion
- Shadows
- Slow moving people
- Multiple processes (swaying of trees..)

Webpage

- [Http://www.cs.cmu.edu/~vsam](http://www.cs.cmu.edu/~vsam)

Skin Detection

Kjeldsen and Kender

Training

- Crop skin regions in the training images.
- Build histogram of training images.
- Ideally this histogram should be bi-modal, one peak corresponding to the skin pixels, other to the non-skin pixels.
- Practically there may be several peaks corresponding to skin, and non-skin pixels.

Training

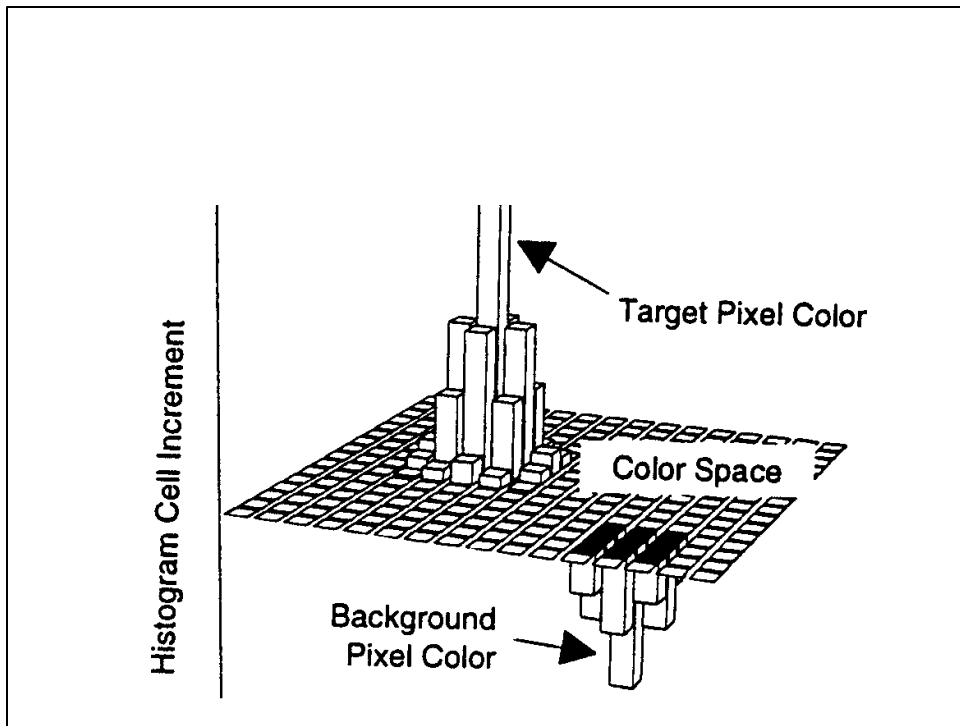
- Apply threshold to skin peaks to remove small peaks.
- Label all gray levels (colors) under skin peaks as “skin”, and the remaining gray levels as “non-skin”.
- Generate a look-up table for all possible colors in the image, and assign “skin” or “non-skin” label.

Detection

- For each pixel in the image, determine its label from the “look-up table” generated during training.

Building Histogram

- Instead of incrementing the pixel counts in a particular histogram bin:
 - for skin pixel increment the bins centered around the given value by a Gaussian function.
 - For non-skin pixels decrement the bins centered around the given value by a smaller Gaussian function.



Tracking People Using Color

Fieguth and Terzopoulos

- Compute mean color vector for each sub region.

$$(r_i, g_i, b_i) = \frac{1}{|R_i|} \sum_{(x,y) \in R_i} (r(x, y), g(x, y), b(x, y))$$

Fieguth and Terzopoulos

- Compute goodness of fit.

$$\Psi_i = \frac{\max \left\{ \frac{r_i}{\bar{r}_i}, \frac{g_i}{\bar{g}_i}, \frac{b_i}{\bar{b}_i} \right\}}{\min \left\{ \frac{r_i}{\bar{r}_i}, \frac{g_i}{\bar{g}_i}, \frac{b_i}{\bar{b}_i} \right\}}$$

Target

Measurement

Fieguth and Terzopoulos

- Tracking

$$\Psi(x_H, y_H) = \sum_{i=1}^N \frac{\Psi_i(x_H + x_i, y_H + y_i)}{N}$$

$$(\hat{x}, \hat{y}) = \arg_{(x_H, y_H)} \min\{\Psi(x_H, y_H)\}$$

Fieguth and Terzopoulos

- Non-linear velocity estimator

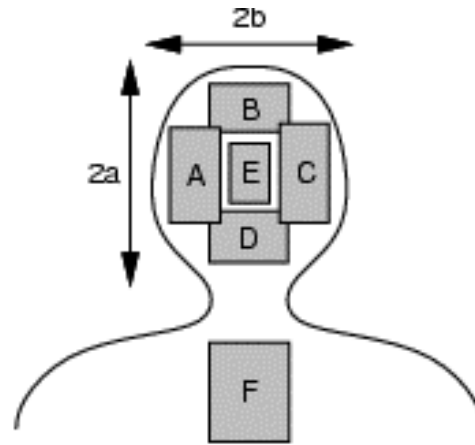
$$v(f) = v(f-1)$$

$$\text{if } (\mathbf{r}(f) \cdot \mathbf{r}(f-1) > 0) \quad v(f) \quad += \quad \mathbf{d} \frac{\text{sgn}(\mathbf{r}(f))}{\Delta t}$$

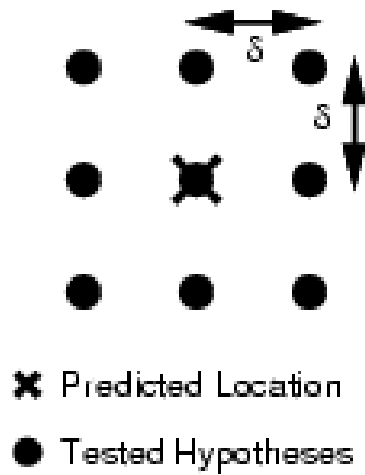
$$\text{if } (\mathbf{r}(f) \cdot v(f-1) < 0) \quad v(f) \quad += \quad \mathbf{d} \frac{\text{sgn}(\mathbf{r}(f))}{\Delta t}$$

$$\text{if } (\mathbf{r}(f) = 0) \quad v(f) \quad -= \quad \mathbf{d} \frac{\text{sgn}(v(f))}{2\Delta t}$$

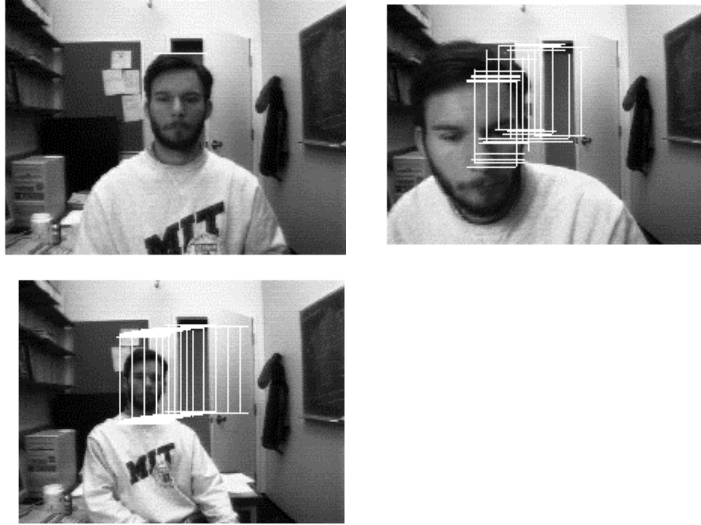
Fieguth and Terzopoulos



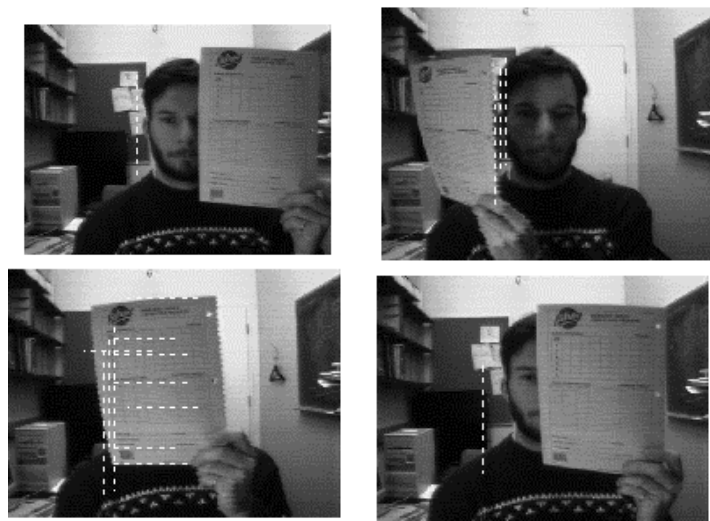
Fieguth and Terzopoulos



Fieguth and Terzopoulos



Fieguth and Terzopoulos



Bibliography

- J. K. Aggarwal and Q. Cai, “Human Motion Analysis: A Review”, *Computer Vision and Image Understanding*, Vol. 73, No. 3, March, pp. 428-440, 1999
- .Azarbayejani, C. Wren and A. Pentland, “Real-Time 3D Tracking of the Human Body”, MIT Media Laboratory, Perceptual Computing Section, TR No. 374, May 1996
- .W.E.L. Grimson *et. al.*, “Using Adaptive Tracking to Classify and Monitor Activities in a Site”, *Proceedings of Computer Vision and Pattern Recognition*, Santa Barbara, June 23-25, 1998, pp. 22-29

Bibliography

- .Takeo Kanade *et. al.* “Advances in Cooperative Multi-Sensor Video Surveillance”, *Proceedings of Image Understanding workshop*, Monterey California, Nov 20-23, 1998, pp. 3-24
- .Haritaoglu I., Harwood D, Davis L, “W⁴ - Who, Where, When, What: A Real Time System for Detecting and Tracking People”, *International Face and Gesture Recognition Conference*, 1998
- .Paul Fieguth, Demetri Terzopoulos, “Color-Based Tracking of Heads and Other Mobile Objects at Video Frame Rates”, *CVPR 1997*, pp. 21-27

Monitoring Human Behavior In an Office Environment

Goals of the System

- Recognize human actions in a room for which **prior knowledge** is available.
- Handle multiple people
- Provide a textual description of each action
- Extract “key frames” for each action

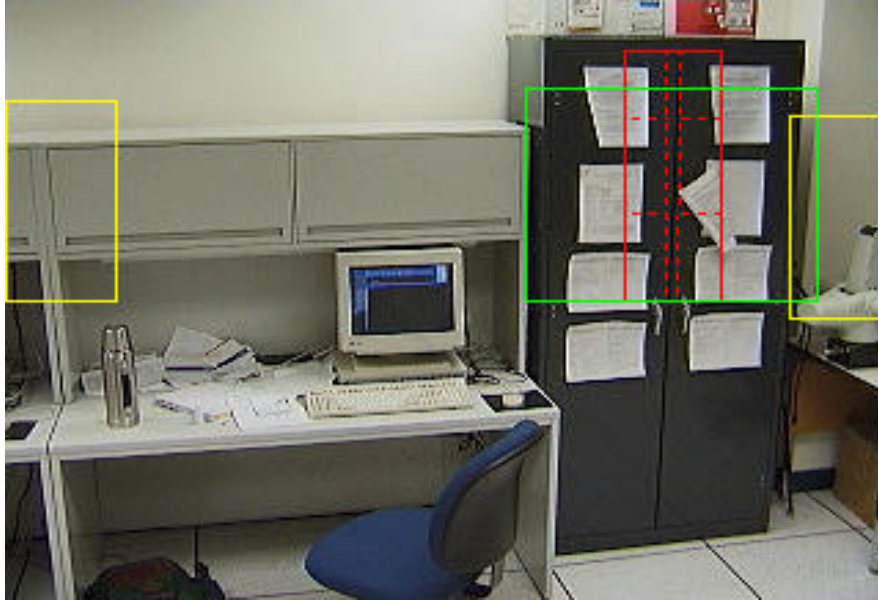
Possible Actions

- **Enter**
- **Leave**
- **Sitting** or **Standing**
- **Picking Up Object**
- **Put Down Object**
-

Prior Knowledge

- Spatial layout of the scene:
 - Location of **entrances** and **exits**
 - Location of **objects** and some information about how they are use
- Context can then be used to improve recognition and save computation

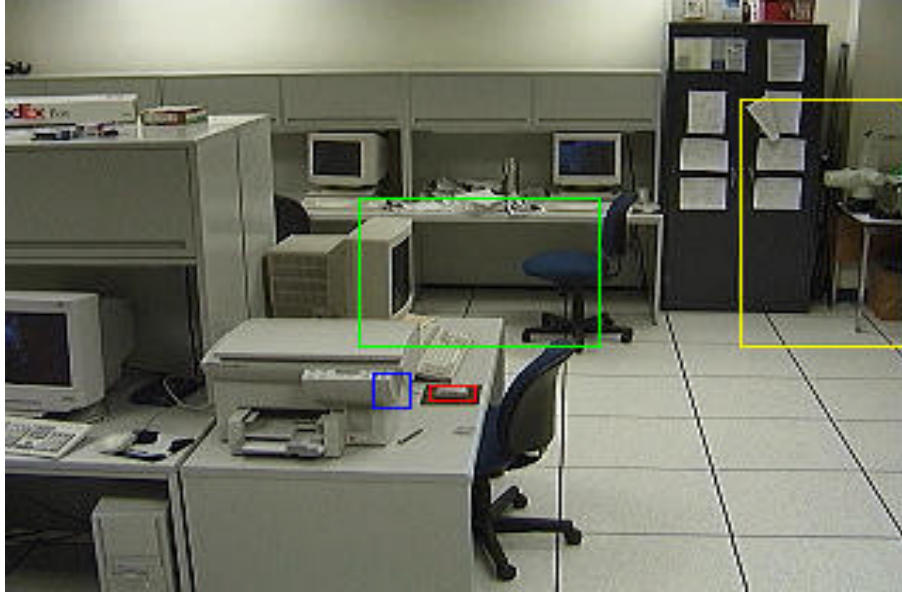
Layout of Scene 1



Layout of Scene 2



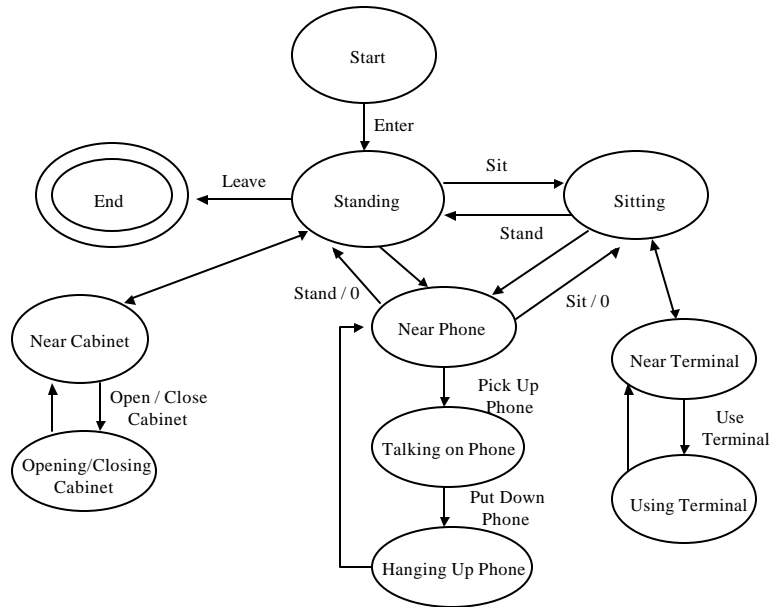
Layout of Scene 4



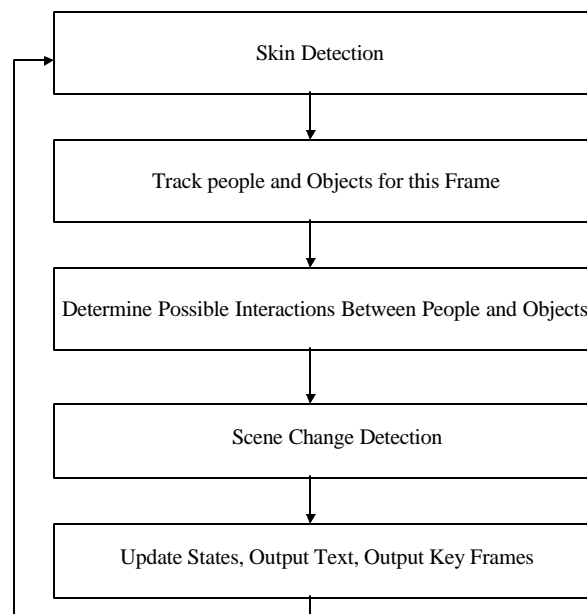
Major Components

- Skin Detection
- Tracking
- Scene Change Detection
- Action Recognition

State Model For Action Recognition



Flow of the System



Key Frames

- Why get key frames?
 - Key frames take less space to store
 - Key frames take less time to transmit
 - Key frames can be viewed more quickly
- We use heuristics to determine when key frames are taken
 - Some are taken before the action occurs
 - Some are taken after the action occurs

Key Frames

- “Enter” key frames: as the person leaves the entrance/exit area
- “Leave” key frames: as the person enters the entrance/exit area
- “Standing/Sitting” key frames: after the tracking box has stopped moving up or down respectively
- “Open/Close” key frames: when the % of changed pixels stabilizes

Results



Key Frames Sequence 1 (350 frames), Part 1



Key Frames Sequence 1 (350 frames), Part 2





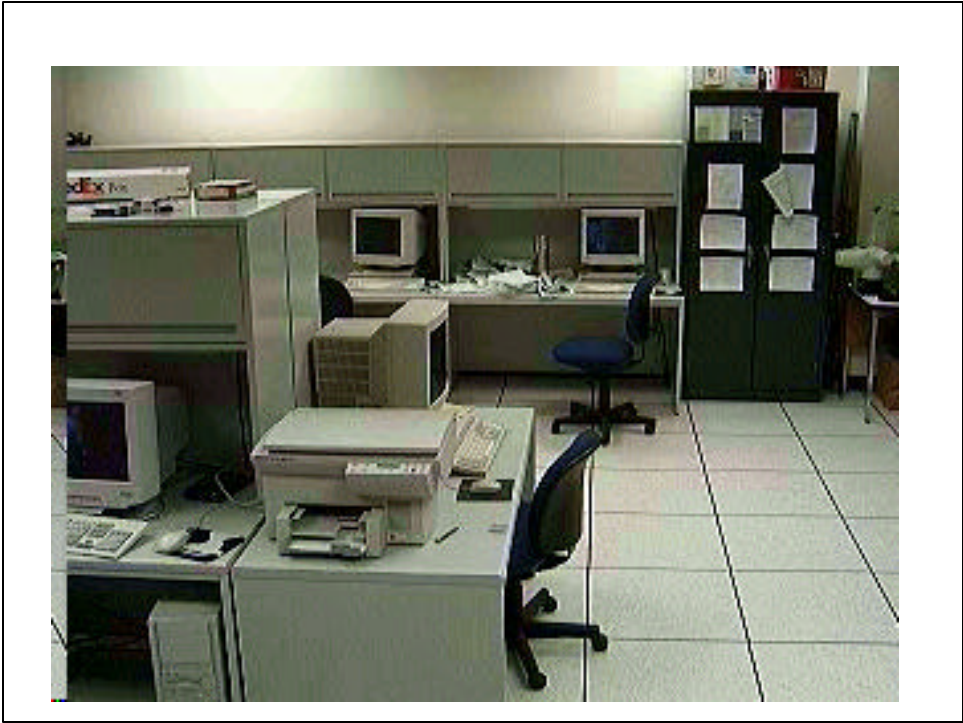
Key Frames Sequence 2 (200 frames)





Key Frames Sequence 3 (200 frames)





Key Frames Sequence 4 (399 frames), Part 1



Key Frames Sequence 4 (399 frames), Part 2



