JAIST
JAPAN
ADVANCED INSTITUTE OF
SCIENCE AND TECHNOLOGY
1990

Pedestrian Tracking by Learning of Multi-view Human Color Appearance

Huynh Loc Huu

JAIST, 2012/08

Content

1. Introduction

- Locating the accurate position of all persons in the observed area.
- Different people are enclosed by different color rectangles.
- Each people is enclosed by the same color in all camera-views.

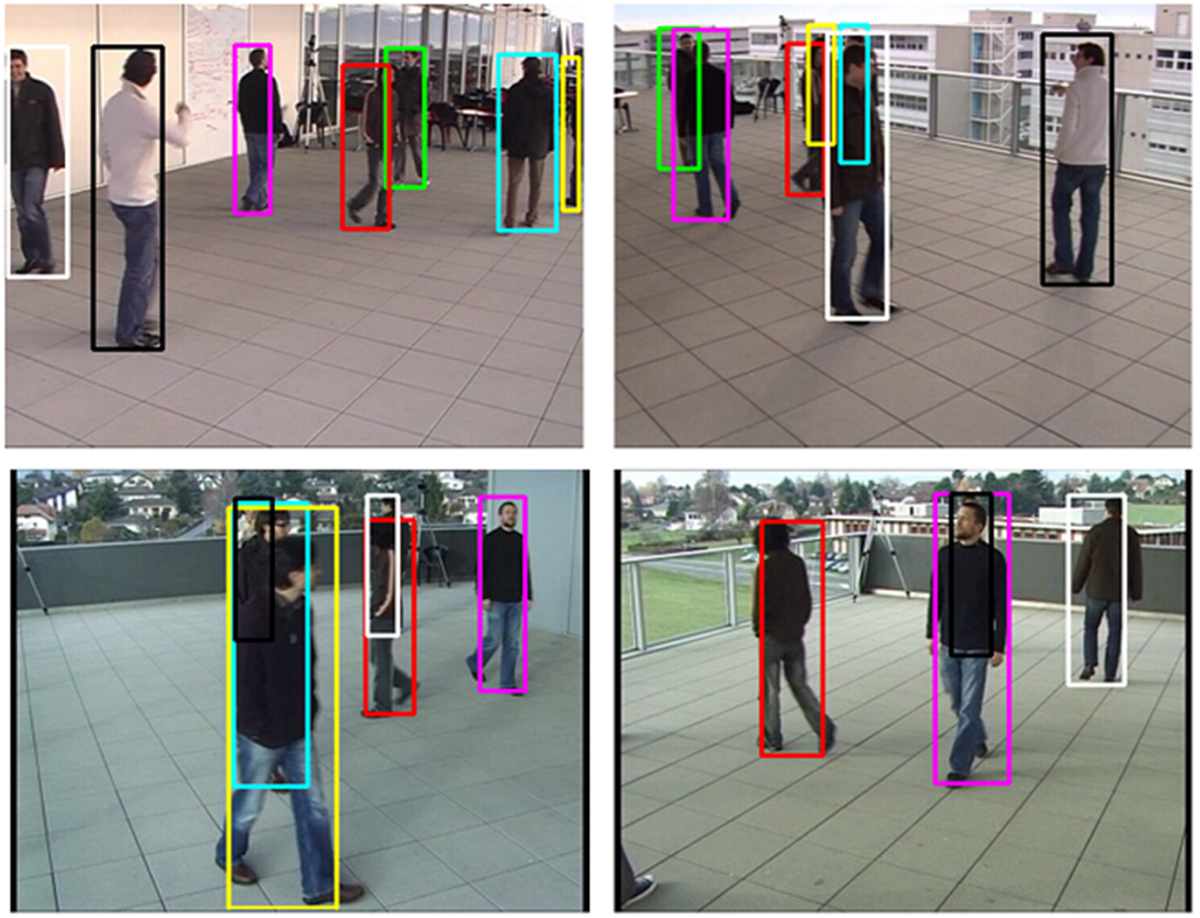


Figure 1.1: Result of Pedestrian Tracking by Using Multiple Cameras

1.1 Applicable Service

- Pedestrian surveillance for security purpose:
  - Detecting and identify tragic events, potential accidents.
  - Automatically reporting to the security center.
  - Reduce the human workload of security authority.
- The research of analysis customer behavior in supermarket by navigating the interesting location of the customers.
- Analyzing team sports' tactical information.

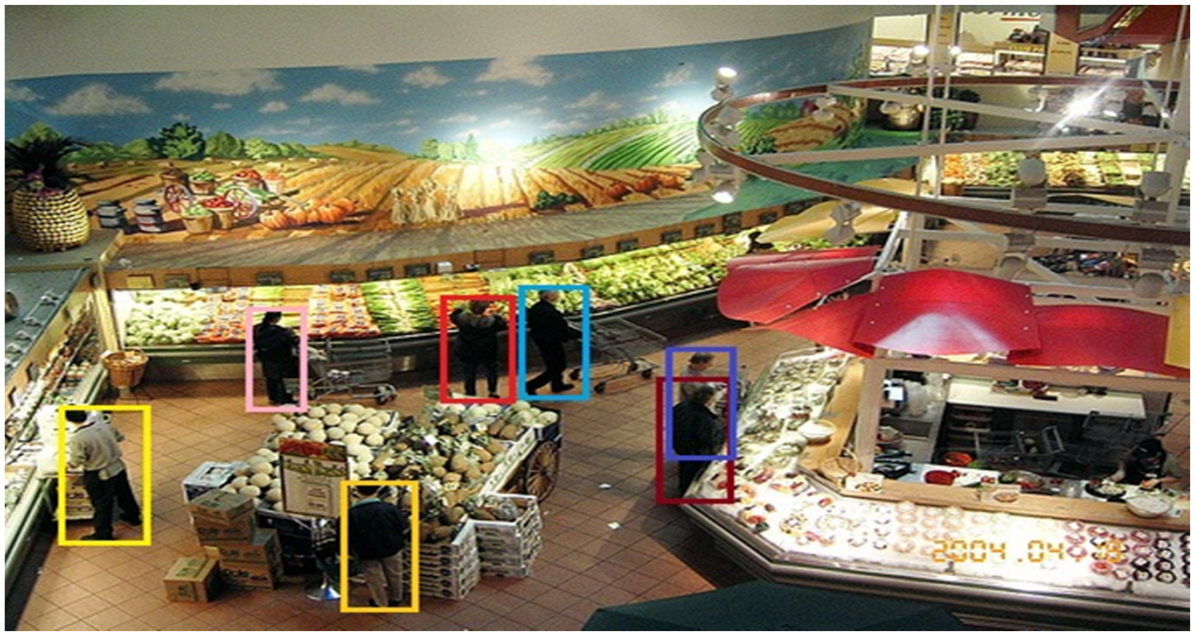


Figure 1.2: Pedestrian Surveillance in Supermarket

1.2 Challenges

Figure 1.3: Obstacles

Figure 1.4: People is occluded

Figure 1.5: Identities switching

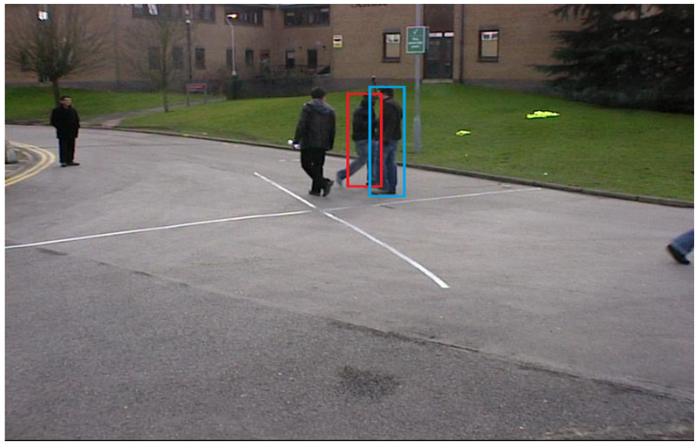


Figure 1.6: Interacting between people

1.3 Motivation

Figure 1.7: Noisy made it very hard to detect people

Figure 1.8: Accurate detection by using the information from color

- We are motivated from the problems of previous methods which used only the information from foreground mask. By the using of color appearance, we could detect people more accurately and help to overcome the identities switching problem.

- The tendency of moving such as spatial location and velocity also can be a very important cue to smooth the tracking trajectories.

1.4 Proposed method

- Handling occlusion by using multiple cameras.
- We applied color appearance learning into the tracking system.
  - The advantages is fusing color appearance information from all views without directly reconstructing 3D shape.
  - The appearance of people is learned and updated automatically.
- Human model is the mixing of ellipse and rectangle.
- The tracking model is comprehensively formulated as the modified Bayesian tracking model.
- The tendency of moving such as spatial location and velocity are used to smooth the tracking trajectories.

2.1 Overview of the System

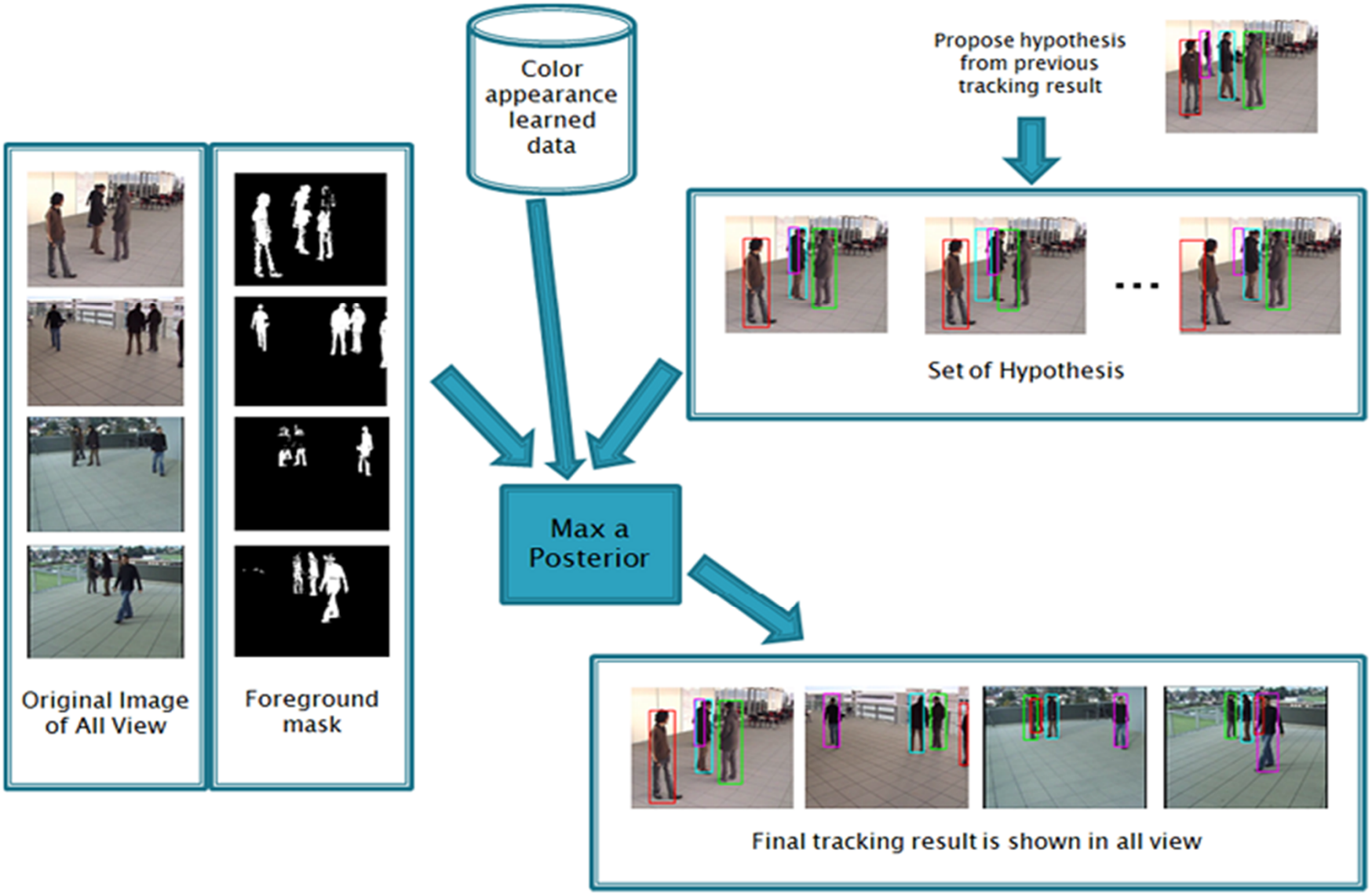


Figure 2.1: Overview of the system

2.2 Formulation a tracking system as MAP

- At time *t*, the state of multi targets is defined as:

$$X^{*(t)} = \{ x_1^{(t)}, \ x_2^{(t)}, \dots \ x_n^{(t)} \}, \mathrm{n} > 0$$

- At target *i*, the state includes the position *c* and the height of person *h*:

$$x_i^{(t)} = \{ \ c_i, \ h_i \ \}$$

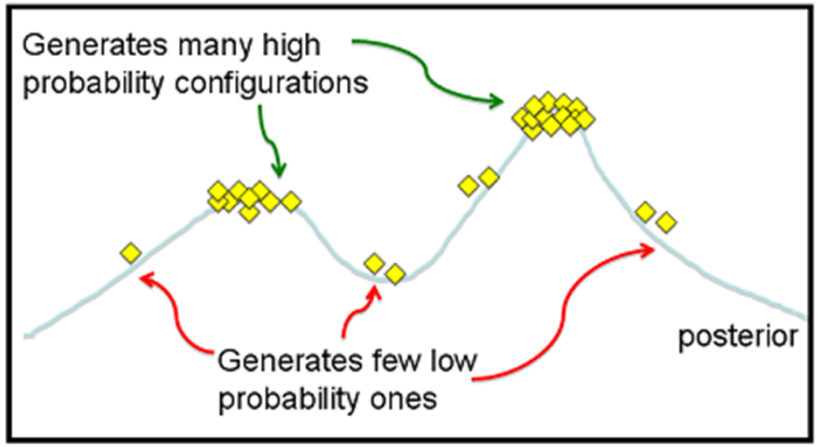
- Formulating the tracking problem as computing the maximum a posterior (MAP) estimation

$$X^{*(t)} = \ argmax_{X^{(t)} \in \omega} \ \mathrm{P}\left(X^{(t)} \mid I^{(t)}, X^{*(t-1)}\right) \qquad (2.1)$$

2.3 Reversible jump Markov chain Monte Carlo (RJMCMC)

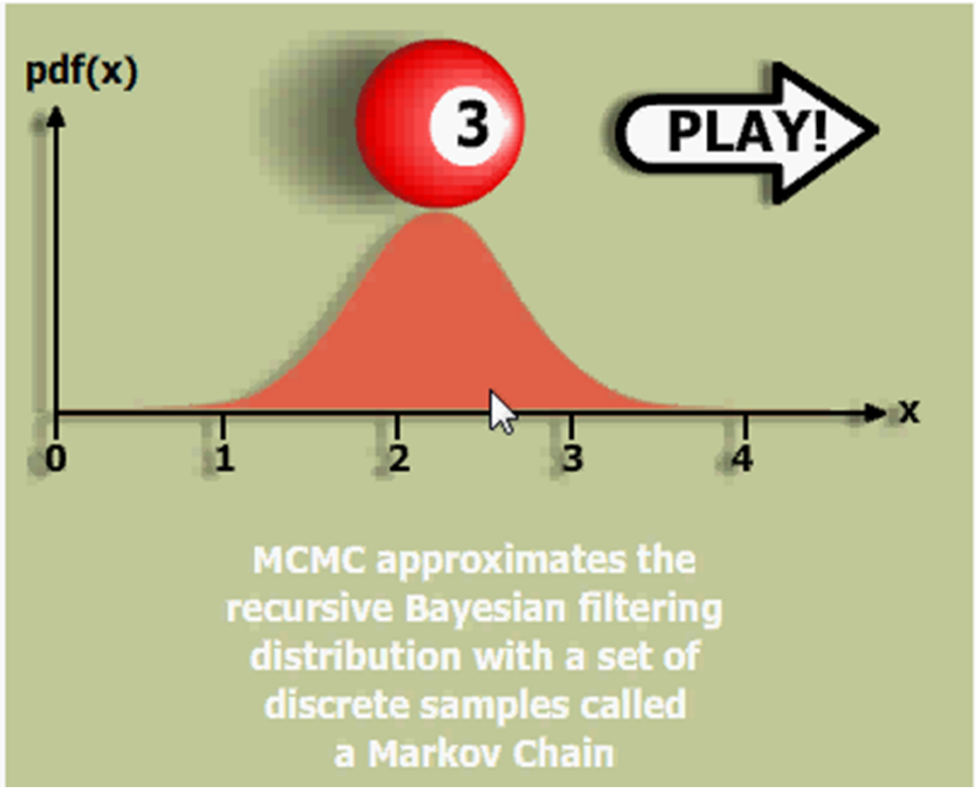
- The growing of number of people also increases extremely number of hypothesis. Therefore, direct computation of the posterior for all hypothesis is intractable.

→ To solve this problem, we used RJMCMC approximate the posterior distribution.



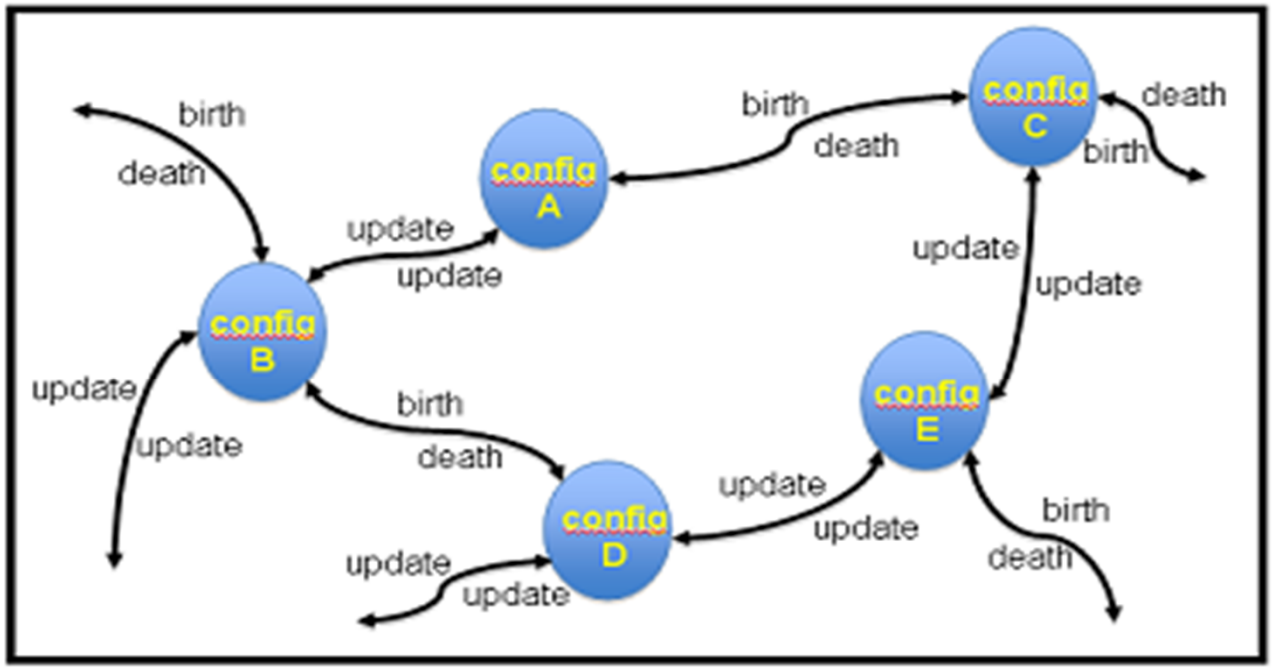
Markov Chain Monte Carlo

Sampling the unknown posterior function with the finite number of sample



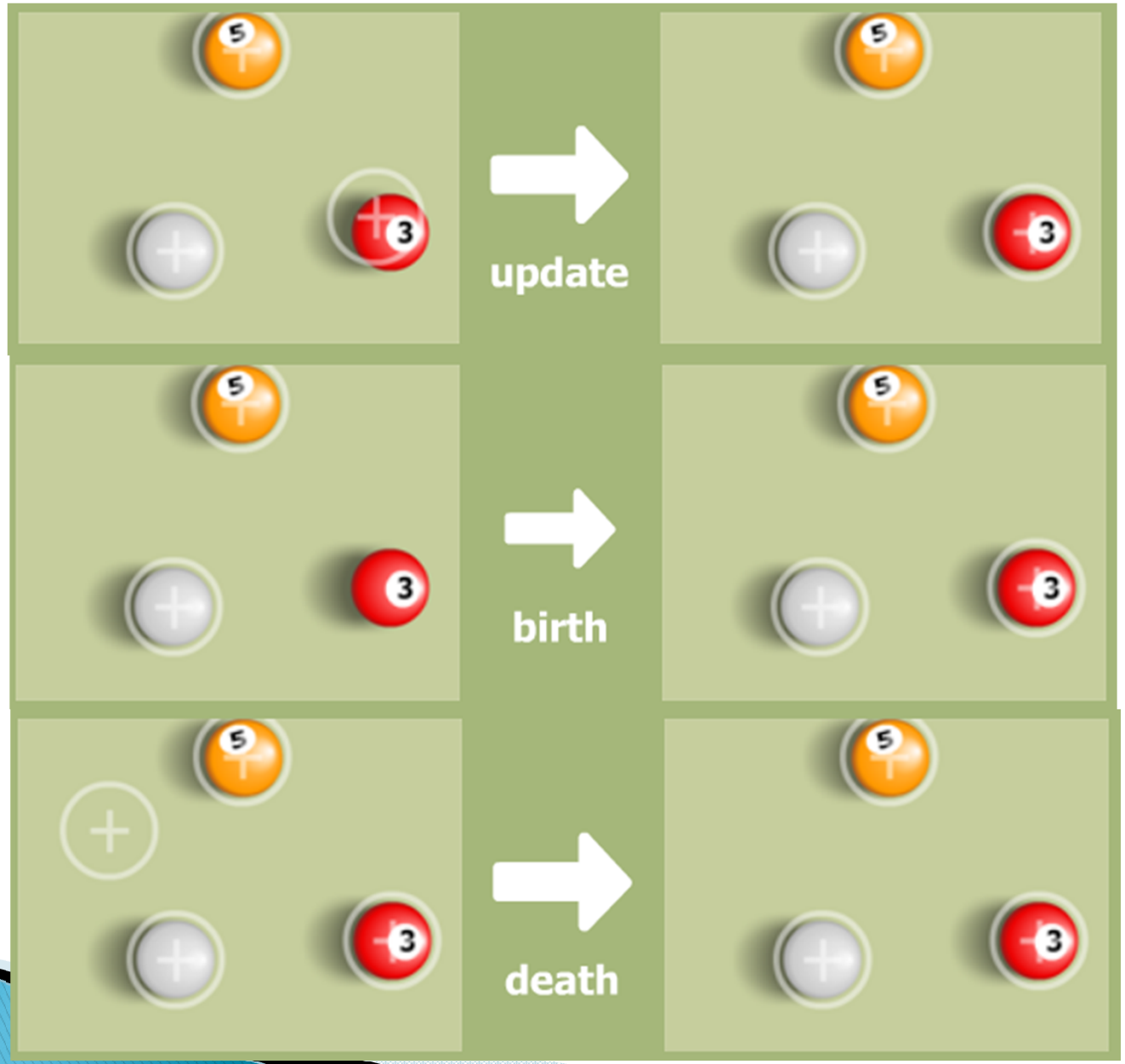
2.3 Reversible jump Markov chain Monte Carlo (RJMCMC)

RJMCMC extends the classical MCMC to handle the unknown number of objects



Exploring the configuration space

2.3 Reversible jump Markov chain Monte Carlo (RJMCMC)



2.4 Posterior probability as modified Bayesian model

- The posterior probability of each hypothesis is decomposed into a likelihood term and a prior term

$$\mathrm{P}\left(X^{(t)} \mid I^{(t)}, X^{*(t-1)}\right) = \mathrm{P}\left(X^{(t)} \mid X^{*(t-1)}\right) * \mathrm{P}\left(I^{(t)} \mid X^{(t)}\right) \qquad (2.2)$$

The prior term

The likelihood term

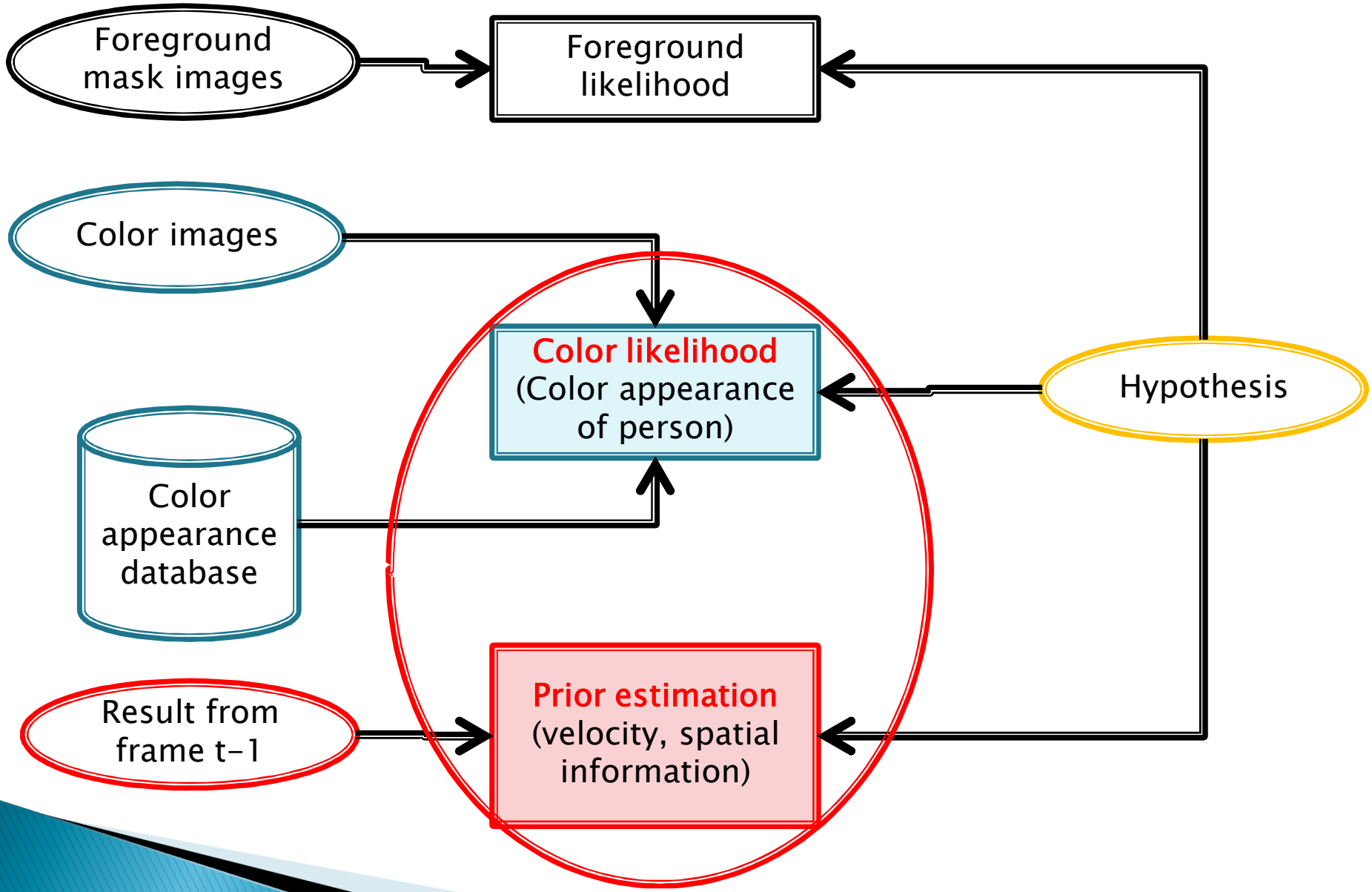
- Our likelihood probability is decomposed into the summarization of the background likelihood and the color likelihood:

$$\mathrm{P}\left(I^{(t)} \mid X^{(t)}\right) = \alpha * \mathrm{P}\left(BI^{(t)} \mid X^{(t)}\right) + \beta * \mathrm{P}\left(CI^{(t)} \mid X^{(t)}, color\_database\right) \qquad (2.3)$$

The background likelihood term

The color likelihood term

2.4 The components of posterior computation



3.1 The Background Likelihood

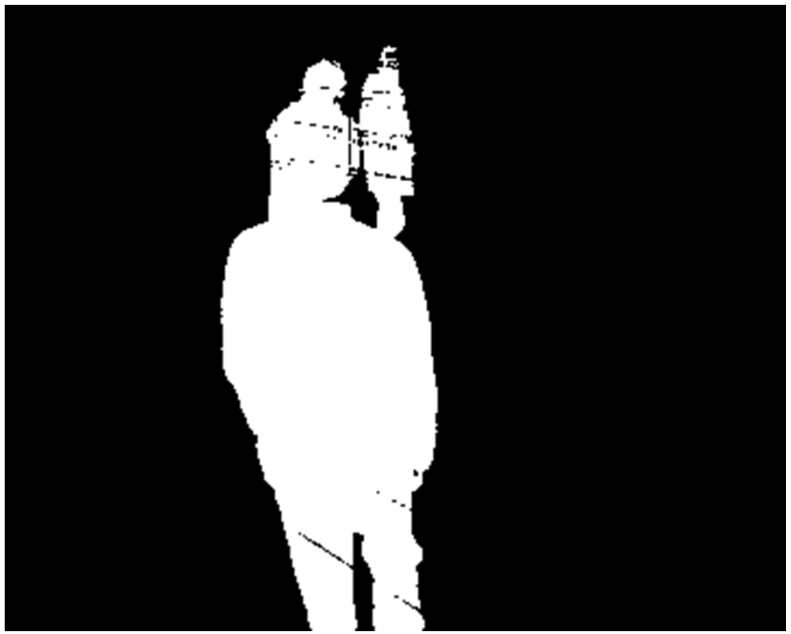


Figure 3.1:
Foreground mask

Figure 3.2:
Human model

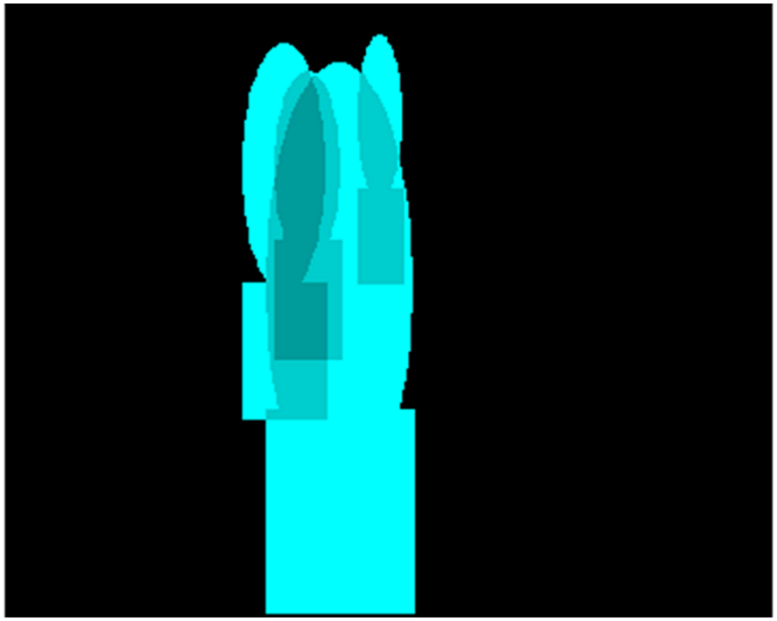
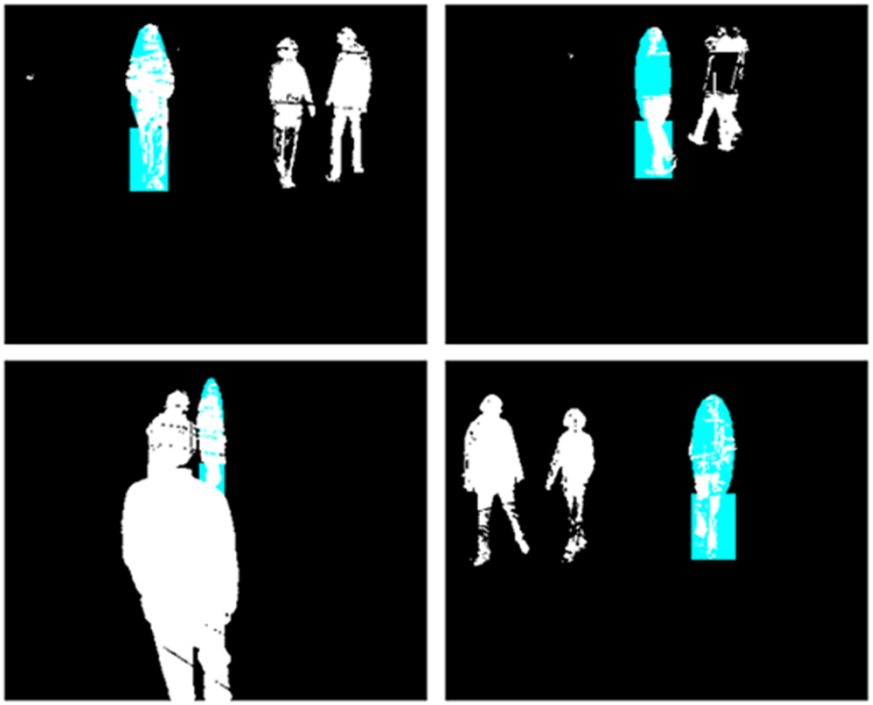


Figure 3.2: Synthesis image of the hypothesis by projecting human model into the image

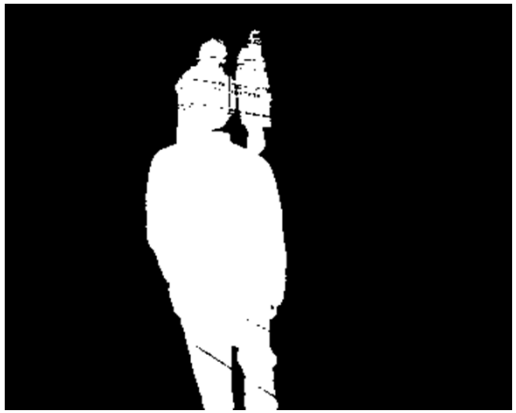
3.1 The Background Likelihood



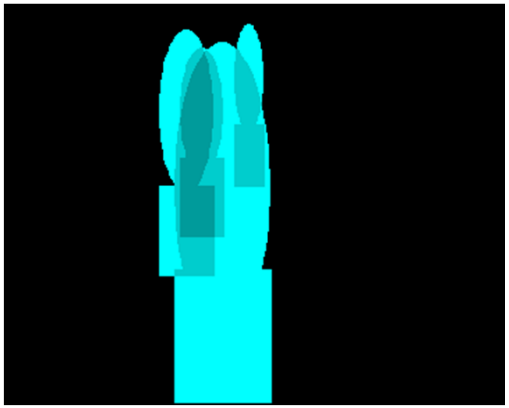
- For fusing all camera views, the general background likelihood probability is computed by summarizing the background likelihood probability of all camera views v :

$$\mathrm{P}\left(BI^{(t)} \mid X^{(t)}\right) = \sum_{v=1}^{V} e^{-\rho\left(BI_v^{(t)}, SBI_v^{(t)}\right)} \qquad (3.1)$$

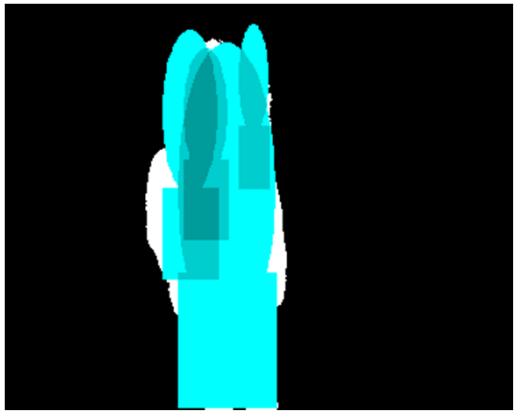
3.1 The Background Likelihood



Foreground image



Synthesis image of hypothesis



Distance between Foreground and Synthesis

- For each view, the normalized pseudo-distance between the foreground mask BI and the synthesis foreground image SBI is computed by counting the number of foreground pixel in the area of synthesis image

$$\rho(BI, SBI) = \frac{\sum_{i=1,j=1}^{Width,Height} p_{ij}^{BI} * (1 - p_{ij}^{SBI}) + p_{ij}^{SBI} * (1 - p_{ij}^{BI})}{\sum_{i=1,j=1}^{Width,Height} p_{ij}^{SBI}} \quad (3.2)$$

3.2 The Color Likelihood

Human model

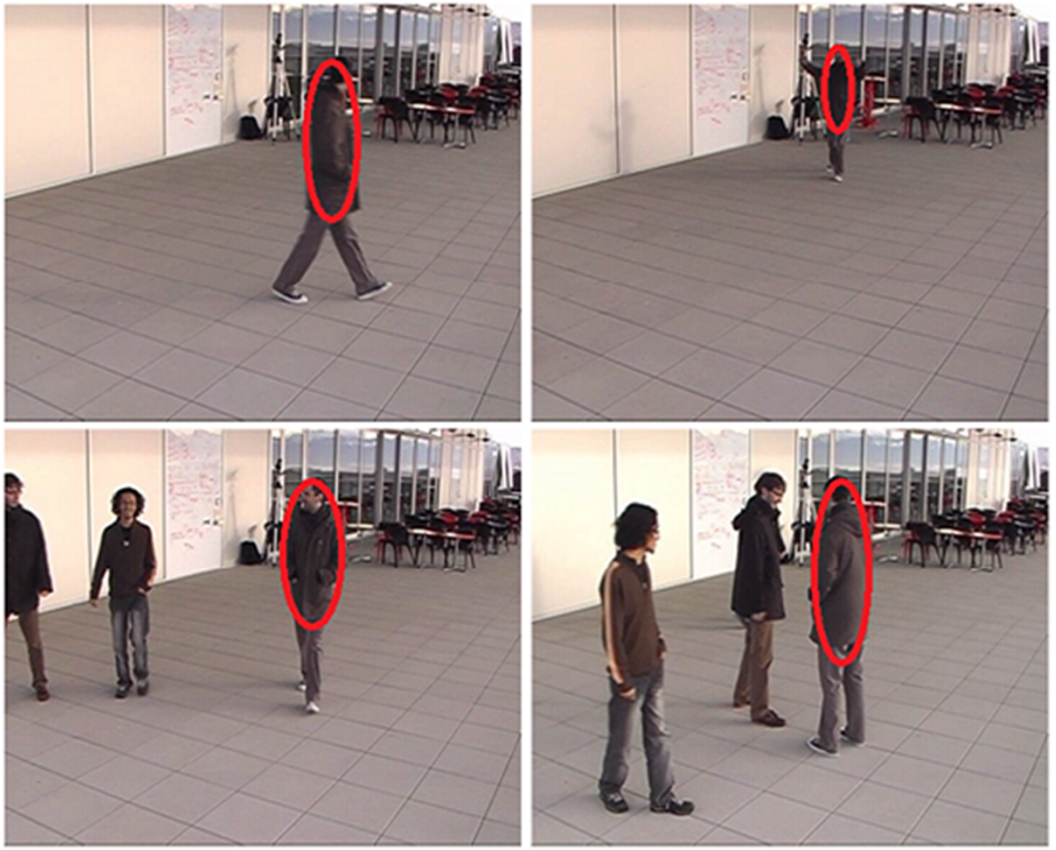


Figure 3.4: Ellipse shape for color matching

3.2 The Color Likelihood

- The color matching step has been done separately each view.
- Result of all views are summarized to contribute the final color likelihood probability following the below formulation:

$$\mathrm{P}(CI^{(t)} \mid X^{(t)}, color\_database) = \sum_{v=1}^{V} e^{-\sigma(CI_v^{(t)} \mid X^{(t)},\ color\_database_v^{(t)})} \quad (3.3)$$

Figure 3.5: The difference in color distribution between different views

3.2 The Color Likelihood

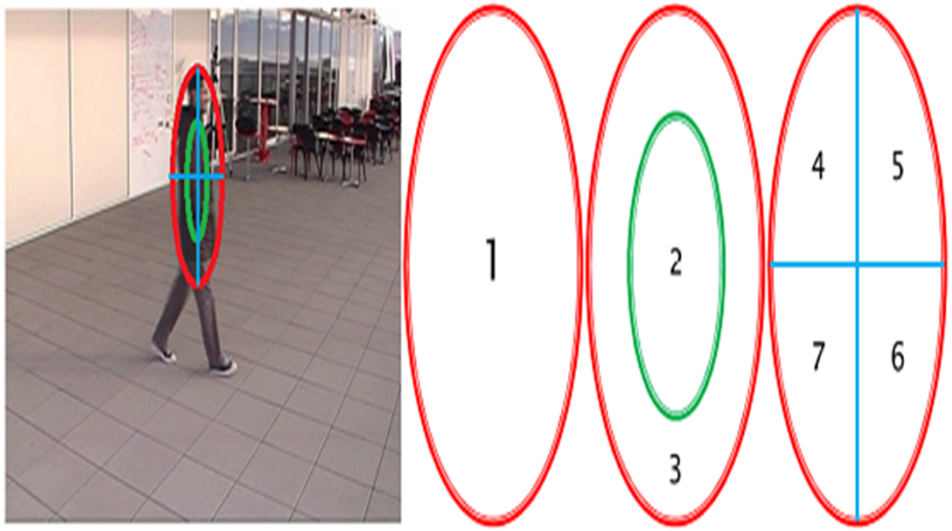
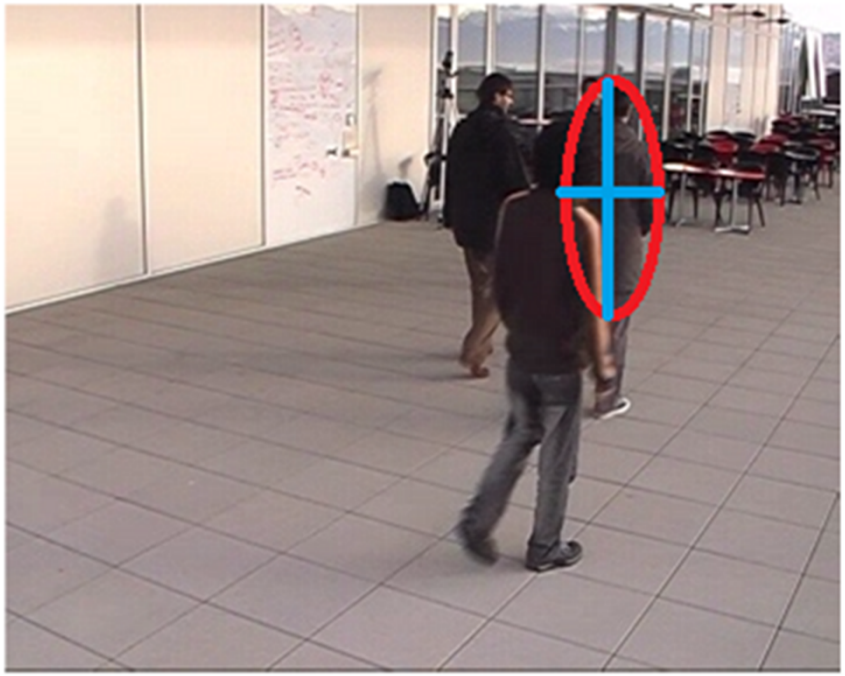


Figure 3.6: Seven parts of ellipse model

Figure 3.7: Visible and Invisible parts of ellipse



3.2 The Color Likelihood

- The color matching step for each person available in the camera view is computed by the summarization of all 7 parts using Bhattachayya distance.

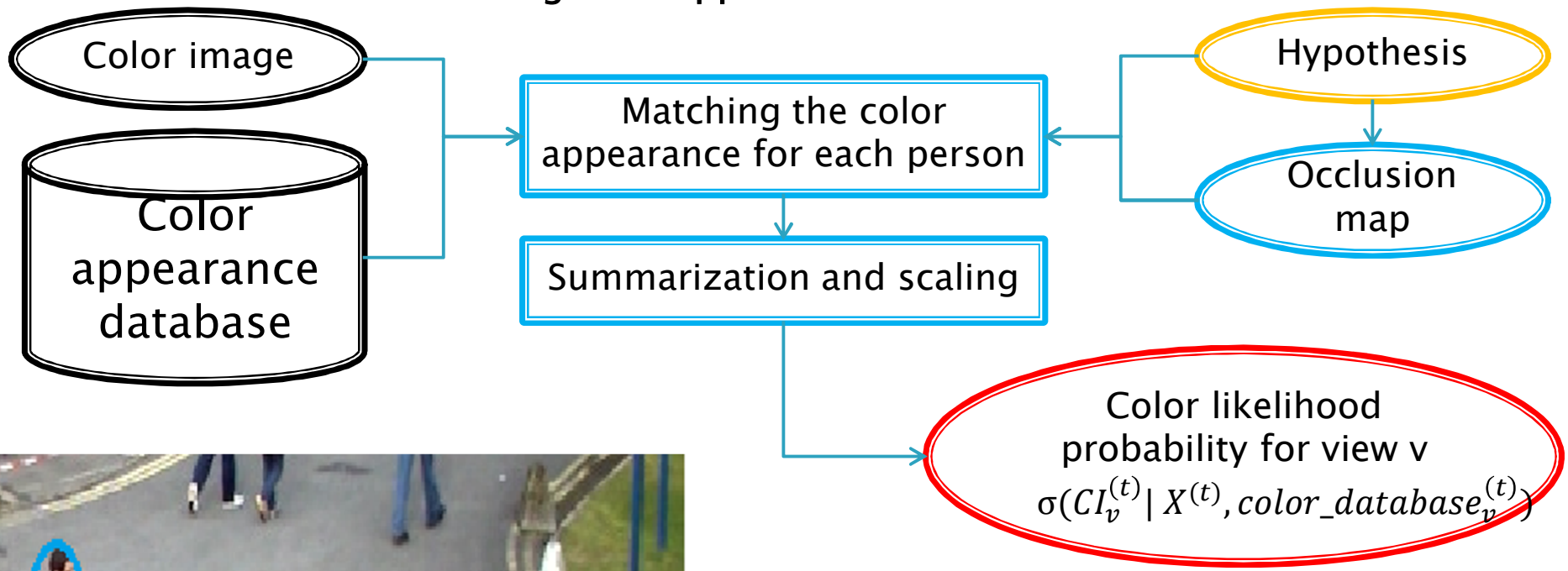
$$\tau\left(\mathrm{CI}\middle|x^{(t)}\right) = \sum_{p=1}^{7} \delta_p BH\left(image\_his_p^{(t)}, database\_his_p^{(t)}\right) \quad (3.4)$$

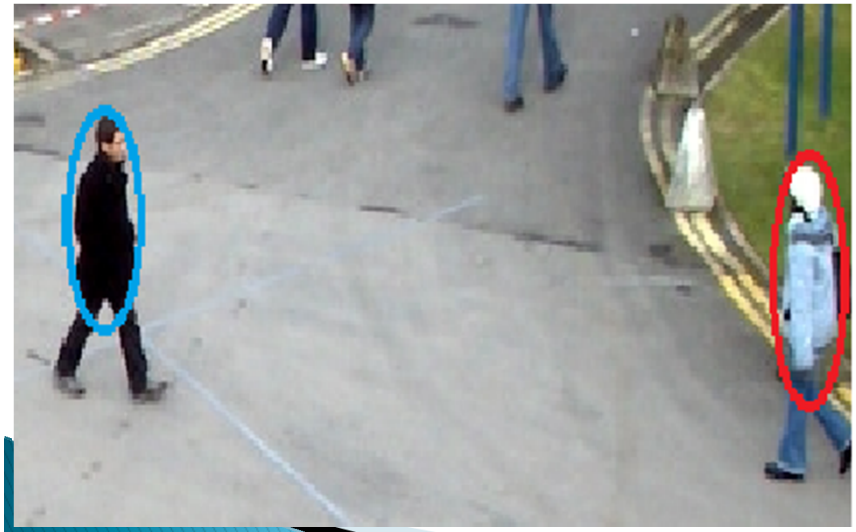
- The color matching for each view *v* is the summarization and scaling of the matching of all targets.

$$\sigma\left(CI_v^{(t)} \middle| X^{(t)},\ color\_database_v^{(t)}\right) = \frac{\sum_{i=1}^{n} \tau\left(CI_v^{(t)} \middle| x_i^{(t)}\right) * visible_i}{\sum_{i=1}^{n} visible_i} \quad (3.5)$$

3.2 The Color Likelihood

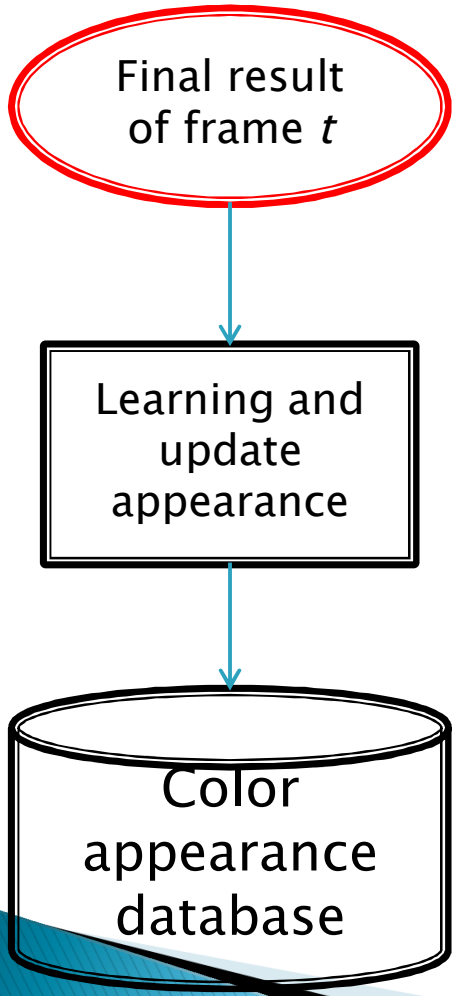
Matching color appearance for each view





3.2 The Color Likelihood

Learning color



Color appearance from previous old frames

Result of tracking at frame *t*

- Store the black coat person and update him with the new color at frame *t.*
- Don't update the blue one because she is invisible in the new frame.

3.3 The Prior

- We realize that tendency of moving is also the important information in pedestrian tracking.
- Our assumptions are related to three kinds of penalty for this prior information.
  - Dynamic model
  - Mutual exclusion model
  - Ghost detection

$$P(X^{(t)} \mid X^{*(t-1)}) = e^{-\mathrm{dyn}(X^{(t)} \mid X^{*(t-1)}, X^{*(t-2)}, \ldots)} * e^{-\mathrm{exc}(X^{(t)})} * e^{-\mathrm{gho}(BI^{(t)} \mid X^{(t)})} \quad (3.6)$$

3.3 The Prior

- People do not change their velocity in the short time. So the velocity *v* of people in the batch of 15 frames should be equal. The dynamic model is:

$$\mathrm{dyn}(X^{(t)}|X^{*(t-1)}, X^{*(t-2)}, \dots) = \sum_{i=1}^{n}\sum_{f=t-15}^{t}\left|v_i^f - v_i^{f+1}\right|^2$$

$$v_i^f = c_i^{f+1} - c_i^f . \qquad (3.7)$$

- Two objects cannot occupy the same space simultaneously. The mutual exclusion model is:

$$\mathrm{exc}(X^{(t)}) = \sum_{i \neq j} \frac{close\_dist^2}{|c_i^t - c_j^t|^2} \qquad (3.8)$$

- If the number of foreground pixels covered by the human model of each human state is too small, the ghost detection will determine a human hypothesis of human is not valid.

$$\mathrm{gho}(BI^{(t)}|\, x_i^{(t)}) = \begin{cases} 1 & \exists v, s.t.\ \frac{|SBI^v(x_i^{(t)}) \cap BI^v|}{|SBI^v(x_i^{(t)})|} \leq ghost\_const \\ 0 & otherwise \end{cases} \qquad (3.9)$$

4.1 Evaluation–Multiple Object Detection Precision

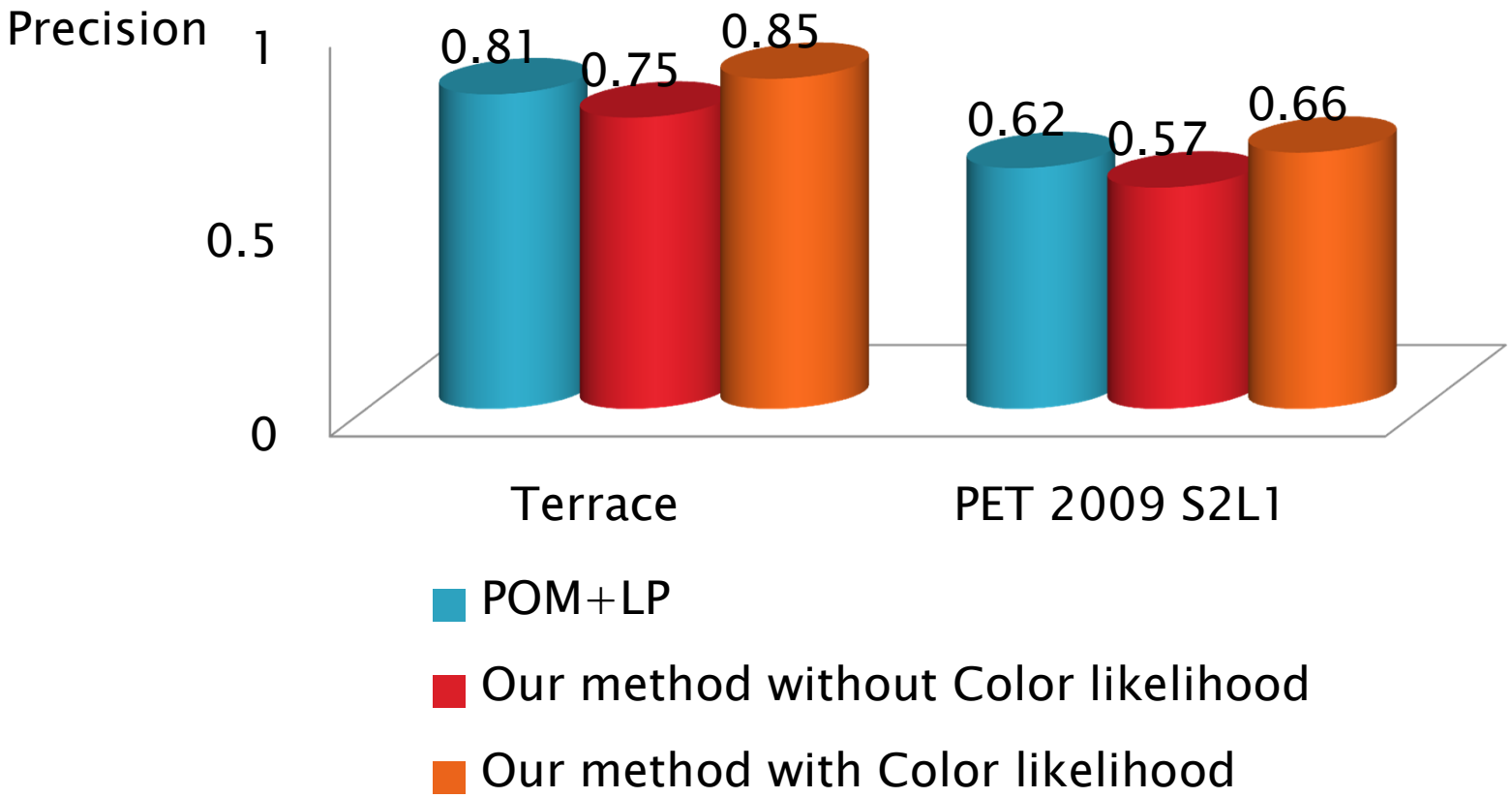


Figure 4.1: The Multiple Object Detection Precision (MODP) evaluation between POM+LP and our proposed methods

4.2 Evaluation–Determining α and β

The background likelihood

The color likelihood

$$P(I^{(t)} \mid X^{(t)}) = \alpha * P(BI^{(t)} \mid X^{(t)}) + \beta * P(CI^{(t)} \mid X^{(t)}, color\_database)$$

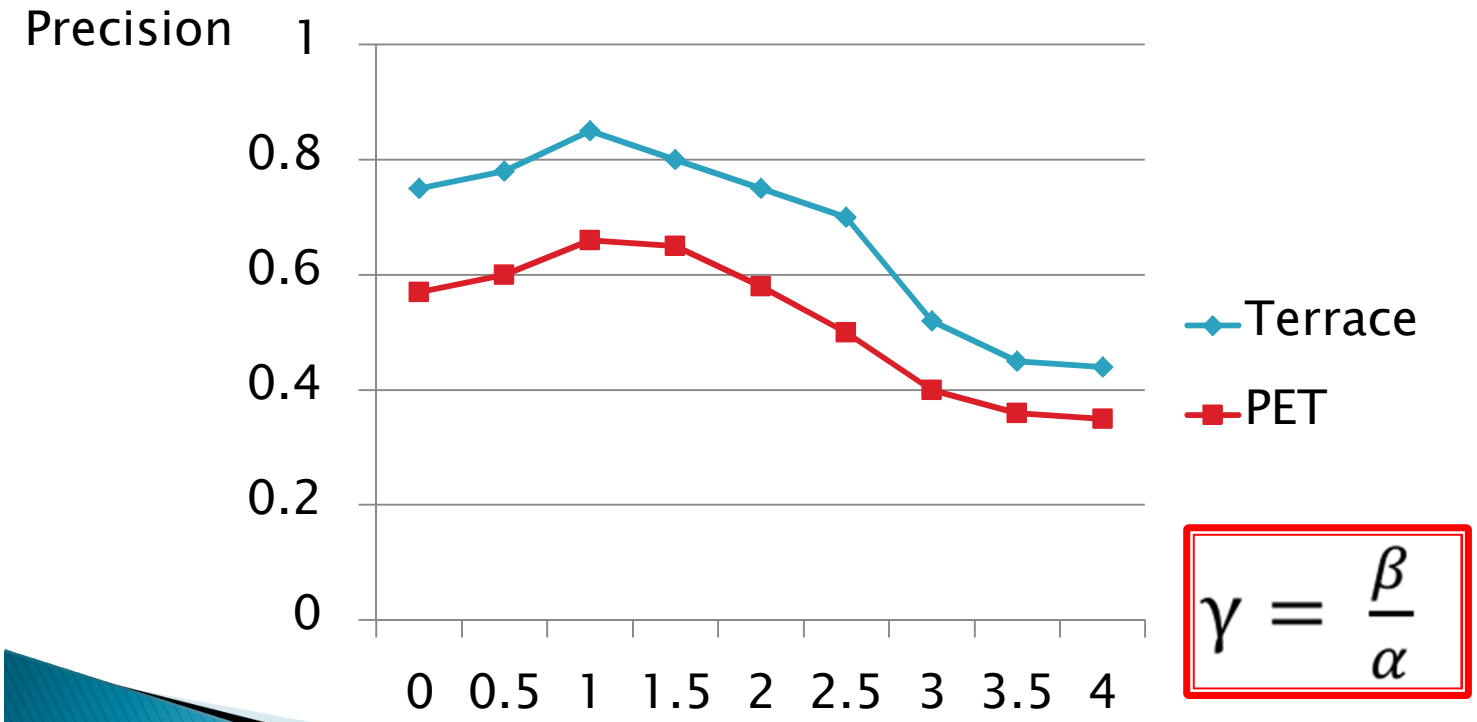


Figure 4.2: Illustration for changing the ratio

5.1 Conclusion

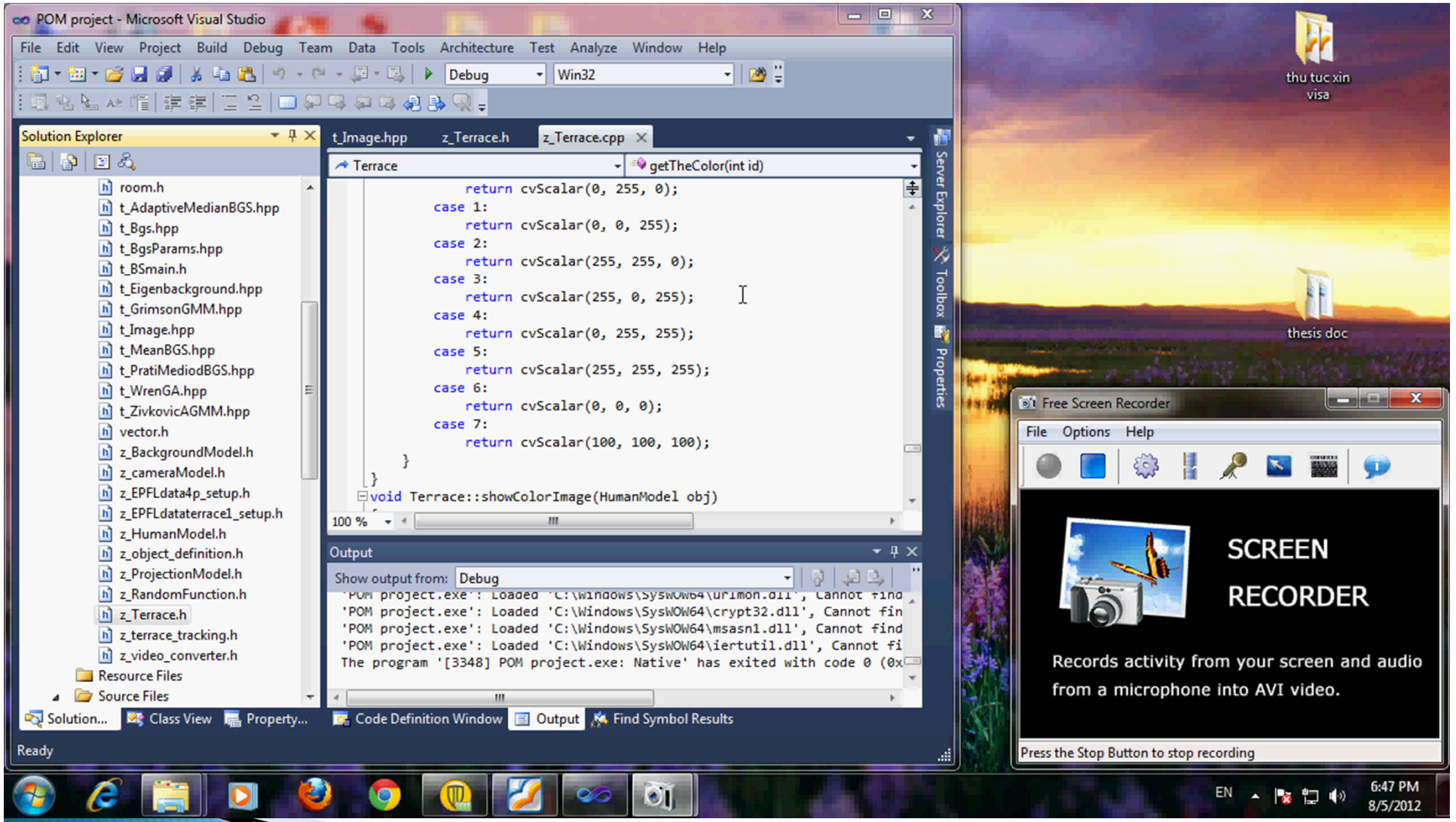
- A vision system capable of detecting multiple people, tracking them and automatically learning their color appearance.
- The color appearance was very powerful not only in improving the precision of locating the people, but also overcoming the identities switching problem.
- The tendency of moving such as spatial location and velocity was smoothing the tracking trajectories.
- The system also has some weakness:
  - The system could not recovery when the system lost its targets because the detection step is not separate from the tracking step.
  - The system is depended on the accuracy of calibration parameters. The error of calibration decreased the precision of the system

5.2 Future works

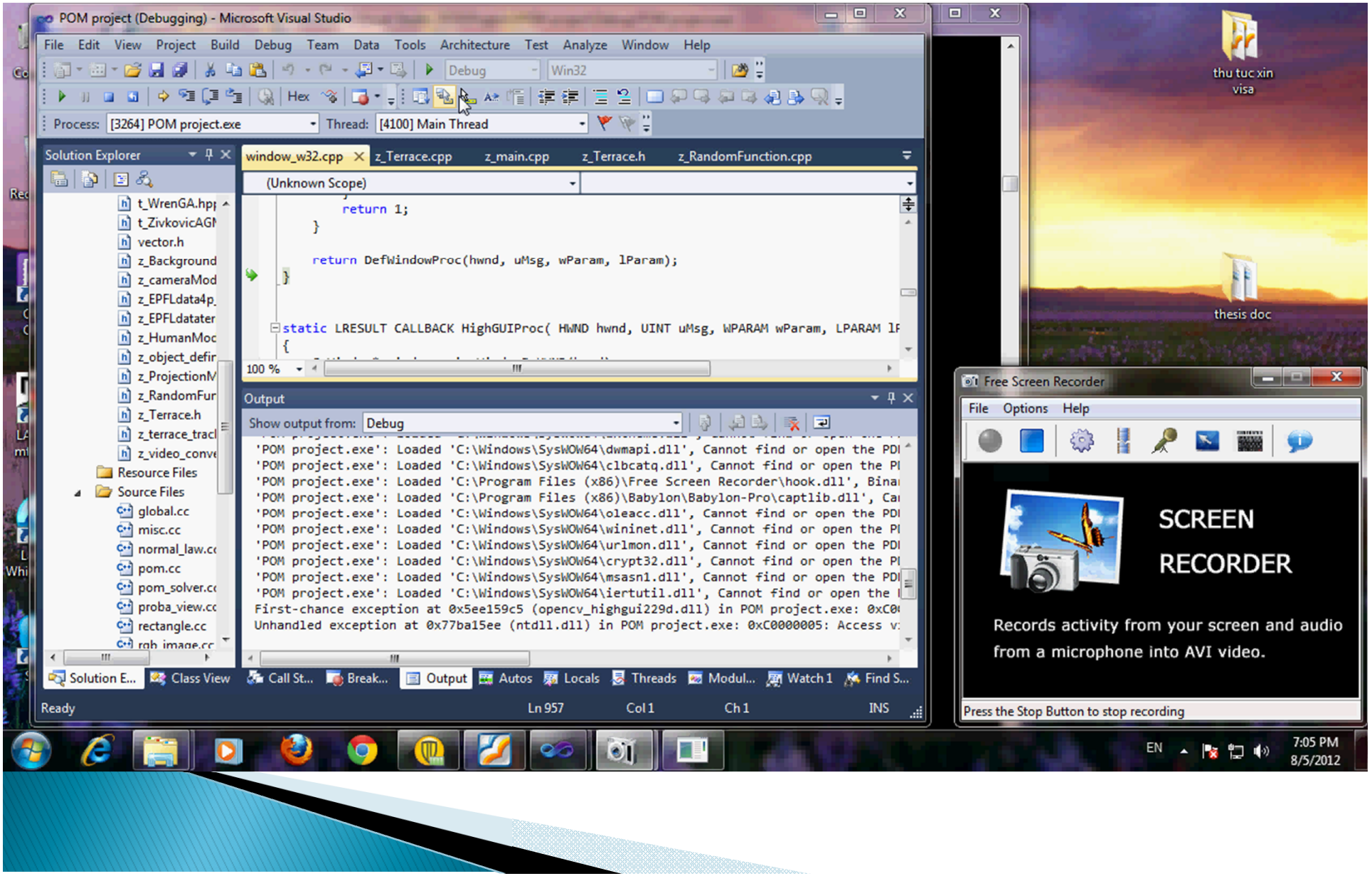
- Using more sophisticated appearance, detecting the people falling down, or jumping up.
- Using more powerful features.
- Using faster processor with more efficient implementation to reach real-time performance.
- Combining MCMC with faster deterministic local search.

Thank you
ご静聴ありがとうございます

Demo

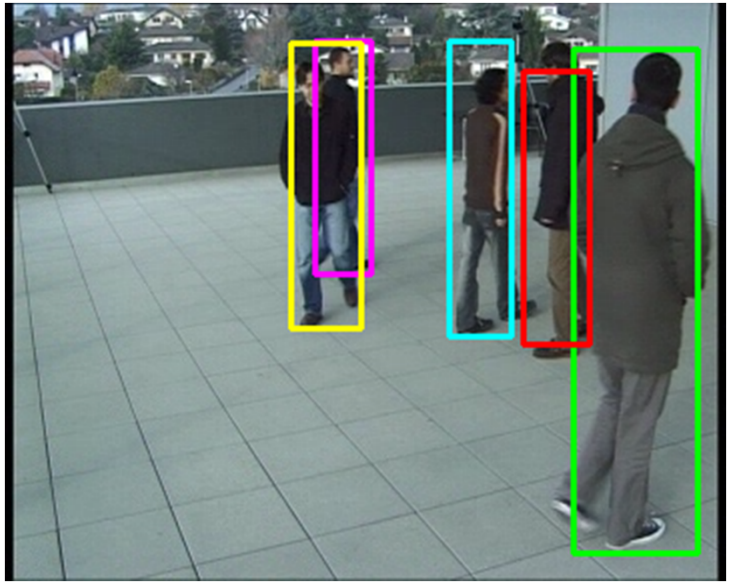

POM project - Microsoft Visual Studio
File Edit View Project Build Debug Team Data Tools Architecture Test Analyze Window Help
Debug
Win32
Solution Explorer
room.h
t_AdaptiveMedianBGS.hpp
t_Bgs.hpp
t_BgsParams.hpp
t_BSmain.h
t_Eigenbackground.hpp
t_GrimsonGMM.hpp
t_Image.hpp
t_MeanBGS.hpp
t_PratiMediodBGS.hpp
t_WrenGA.hpp
t_ZivkovicAGMM.hpp
vector.h
z_BackgroundModel.h
z_cameraModel.h
z_EPFLdata4p_setup.h
z_EPFLdataterrace1_setup.h
z_HumanModel.h
z_object_definition.h
z_ProjectionModel.h
z_RandomFunction.h
z_Terrace.h
z_terrace_tracking.h
z_video_converter.h
Resource Files
Source Files
t_Image.hpp
z_Terrace.h
z_Terrace.cpp
Terrace
getTheColor(int id)
return cvScalar(0, 255, 0);
case 1:
return cvScalar(0, 0, 255);
case 2:
return cvScalar(255, 255, 0);
case 3:
return cvScalar(255, 0, 255);
case 4:
return cvScalar(0, 255, 255);
case 5:
return cvScalar(255, 255, 255);
case 6:
return cvScalar(0, 0, 0);
case 7:
return cvScalar(100, 100, 100);
}
}
void Terrace::showColorImage(HumanModel obj)
100 %
Output
Show output from: Debug
'POM project.exe': Loaded 'C:\Windows\SysWOW64\crypt32.dll', Cannot fin
'POM project.exe': Loaded 'C:\Windows\SysWOW64\msasn1.dll', Cannot find
'POM project.exe': Loaded 'C:\Windows\SysWOW64\iertutil.dll', Cannot fi
The program '[3348] POM project.exe: Native' has exited with code 0 (0x
Solution...
Class View
Property...
Code Definition Window
Output
Find Symbol Results
Ready
Server Explorer
Toolbox
Properties
thu tuc xin visa
thesis doc
Free Screen Recorder
File Options Help
SCREEN RECORDER
Records activity from your screen and audio from a microphone into AVI video.
Press the Stop Button to stop recording
EN
6:47 PM
8/5/2012

Demo


POM project (Debugging) - Microsoft Visual Studio
File Edit View Project Build Debug Team Data Tools Architecture Test Analyze Window Help
Process: [3264] POM project.exe
Thread: [4100] Main Thread
Solution Explorer
window_w32.cpp
z_Terrace.cpp
z_main.cpp
z_Terrace.h
z_RandomFunction.cpp
(Unknown Scope)
return 1;
return DefWindowProc(hwnd, uMsg, wParam, lParam);
static LRESULT CALLBACK HighGUIProc(HWND hwnd, UINT uMsg, WPARAM wParam, LPARAM lP
Output
Show output from: Debug
'POM project.exe': Loaded 'C:\Windows\SysWOW64\dwmapi.dll', Cannot find or open the PDI
'POM project.exe': Loaded 'C:\Windows\SysWOW64\clbcatq.dll', Cannot find or open the PI
'POM project.exe': Loaded 'C:\Program Files (x86)\Free Screen Recorder\hook.dll', Binar
'POM project.exe': Loaded 'C:\Program Files (x86)\Babylon\Babylon-Pro\captlib.dll', Ca
'POM project.exe': Loaded 'C:\Windows\SysWOW64\oleacc.dll', Cannot find or open the PDI
'POM project.exe': Loaded 'C:\Windows\SysWOW64\wininet.dll', Cannot find or open the PI
'POM project.exe': Loaded 'C:\Windows\SysWOW64\urlmon.dll', Cannot find or open the PDI
'POM project.exe': Loaded 'C:\Windows\SysWOW64\crypt32.dll', Cannot find or open the PI
'POM project.exe': Loaded 'C:\Windows\SysWOW64\msasn1.dll', Cannot find or open the PDI
'POM project.exe': Loaded 'C:\Windows\SysWOW64\iertutil.dll', Cannot find or open the
First-chance exception at 0x5ee159c5 (opencv_highgui229d.dll) in POM project.exe: 0xC0
Unhandled exception at 0x77ba15ee (ntdll.dll) in POM project.exe: 0xC0000005: Access v
Ready
Ln 957
Col 1
Ch 1
INS
thu tuc xin visa
thesis doc
Free Screen Recorder
File Options Help
SCREEN RECORDER
Records activity from your screen and audio from a microphone into AVI video.
Press the Stop Button to stop recording
EN
7:05 PM
8/5/2012

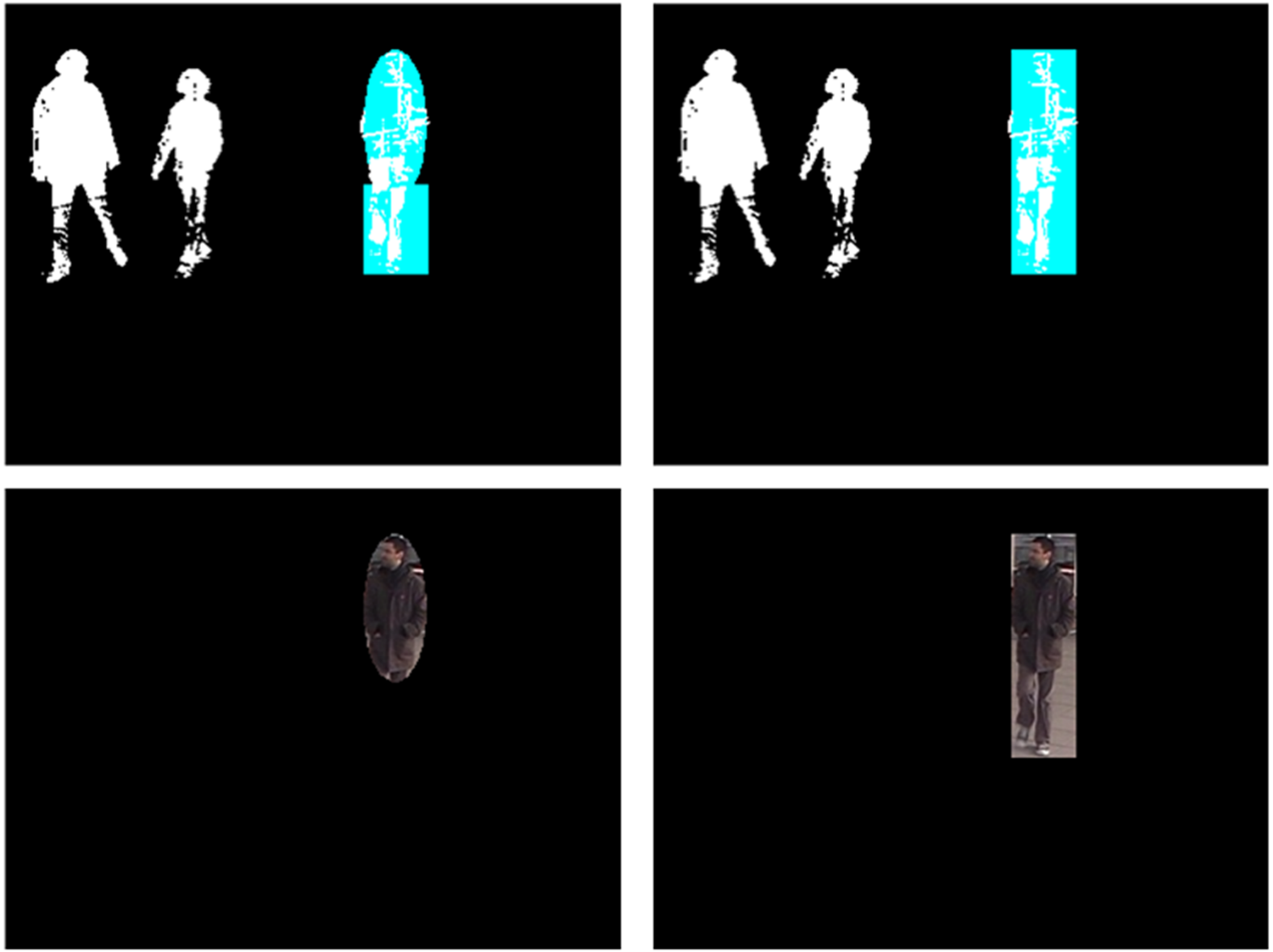
Identities switching without using color likelihood

Tracking result at frame *t−1*

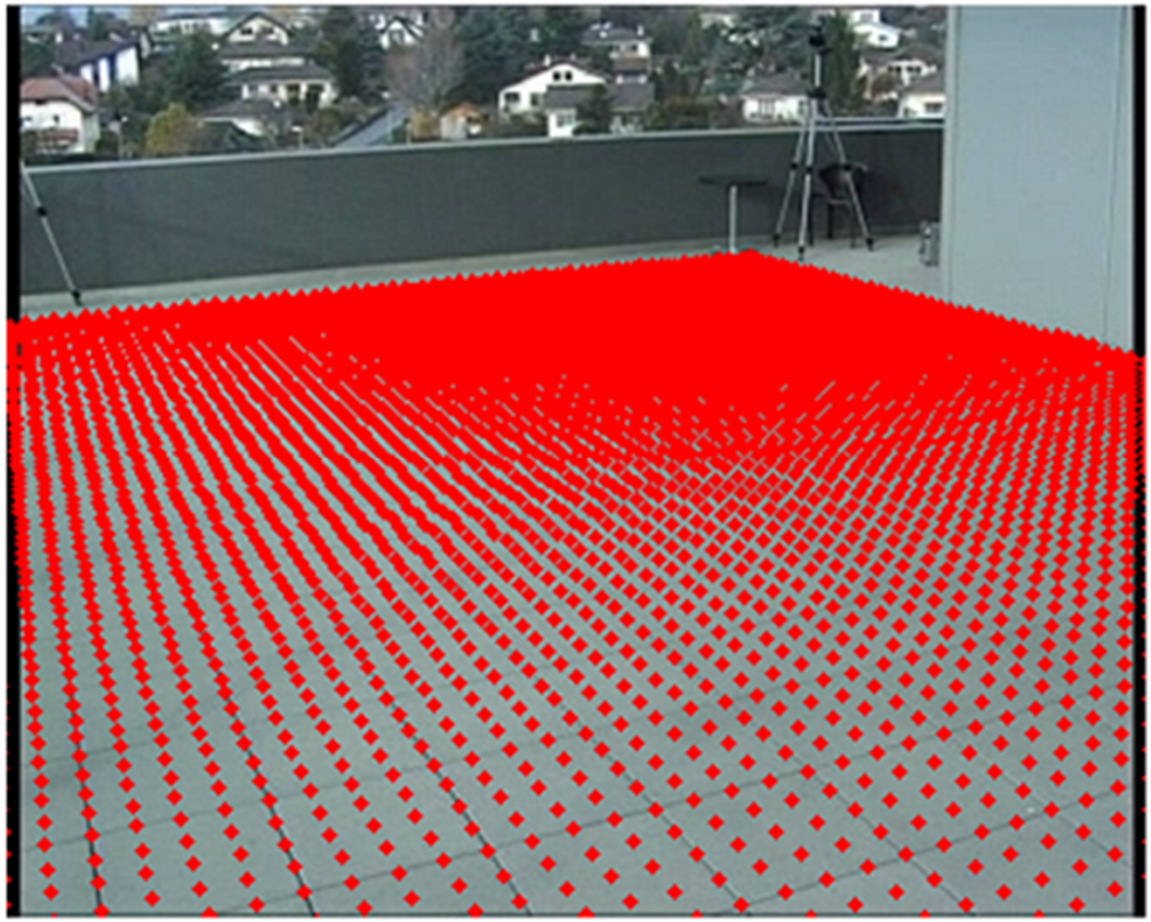


Tracking result at frame *t*

Human model approximate better



Grid Discretization and Camera Calibration



Ground plane is discretized into a finite number of locations. The red points are the locations

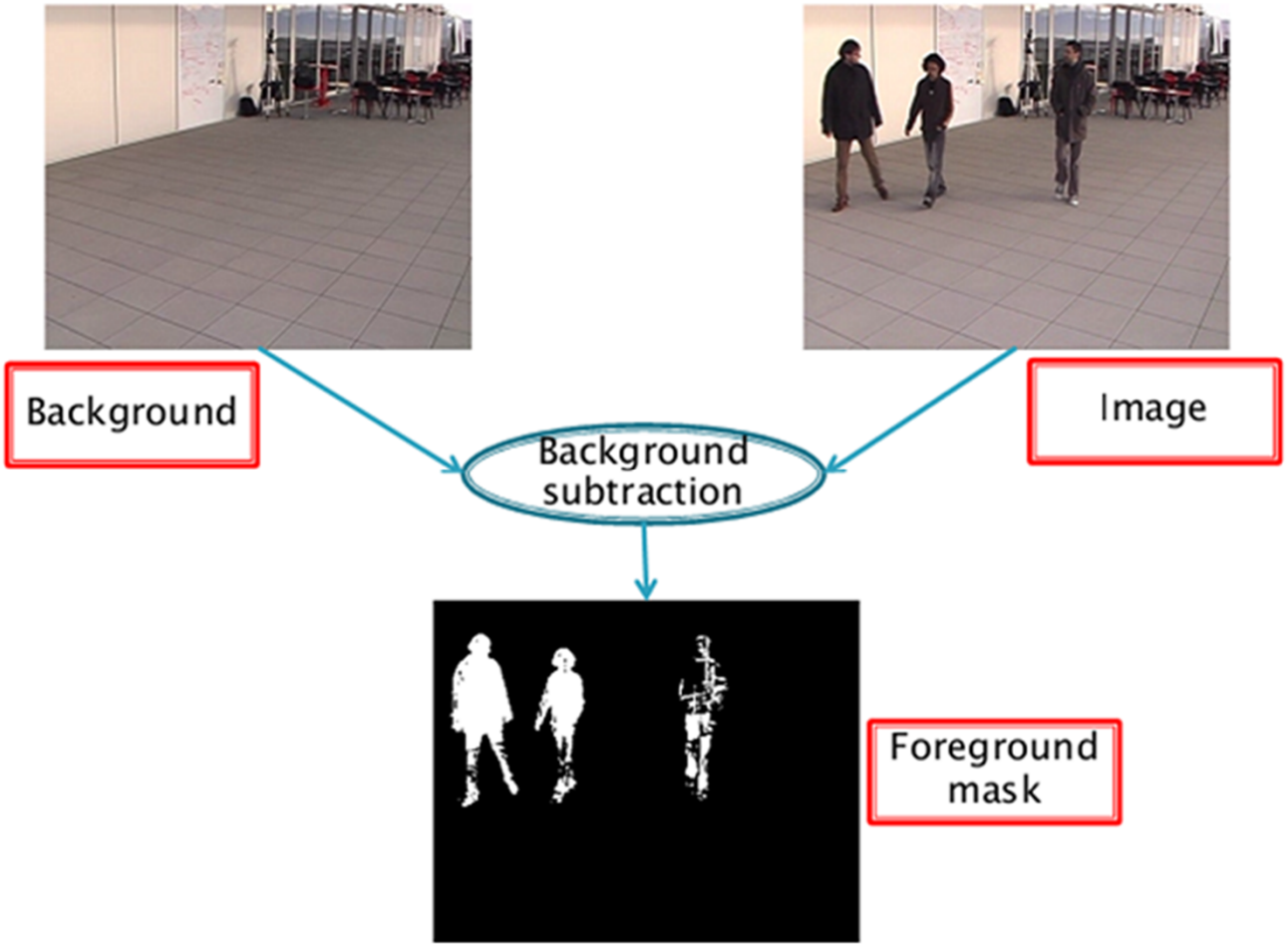
- The ground is discretized in to grid of location.
- The advantage of this discretization is that we could implement our method as a discrete problem.
- The speed of the system is also increased by preventing the complexity of continuous computation.

Grid Discretization and Camera Calibration

- Calibration is the method which finding the corresponding points between different views.
- We used the calibration parameters provided by the dataset.
- There are many methods to calibrate multiple cameras such as Tsai or auto calibration.

Corresponding points in all views show the result of calibration

Background Subtraction



The background subtraction