# Simulação Gráfica
# e
# Visão Computacional

Soraia Raupp Musse

03/12/2013

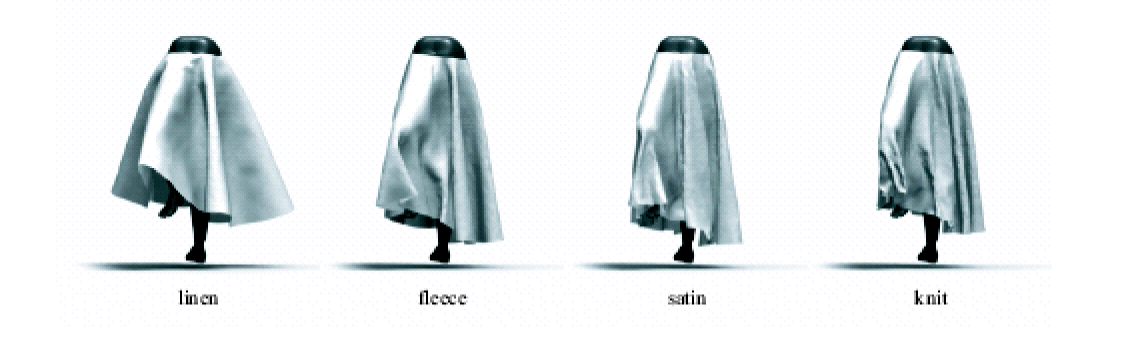
# Objetivo

- Analisar exemplos comerciais e do estado-da-arte científicos que utilizam dados reais para aprimorar a qualidade de simulações e animações.

# Cinema: King Kong

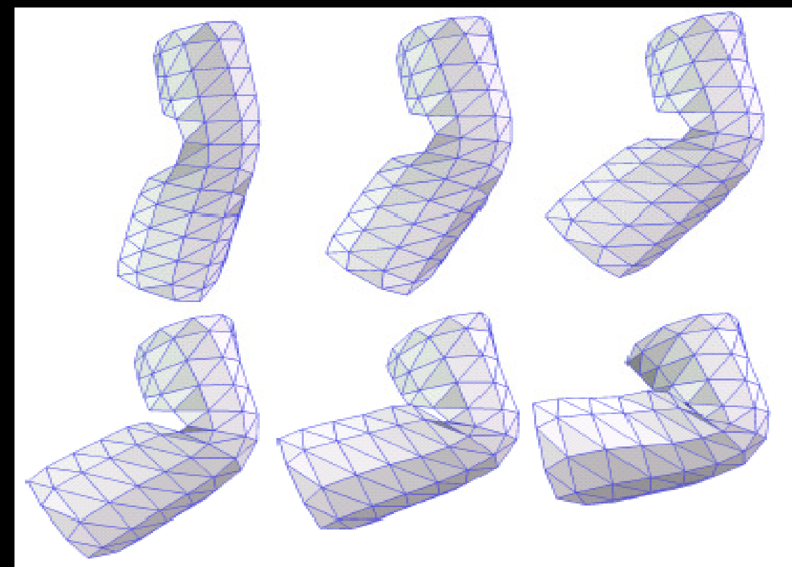
- O estúdio foi equipado com 52 câmeras para gravar todos os movimentos de Andy. Foram usados 60 marcadores na roupa, que mapeados pelo computador, informaram a posição do ator no espaço.
- Posteriormente estes pontos foram ligados em um modelo 3D humano para então criar uma proporção com o modelo do King Kong e repassar todos os movimentos para este modelo.

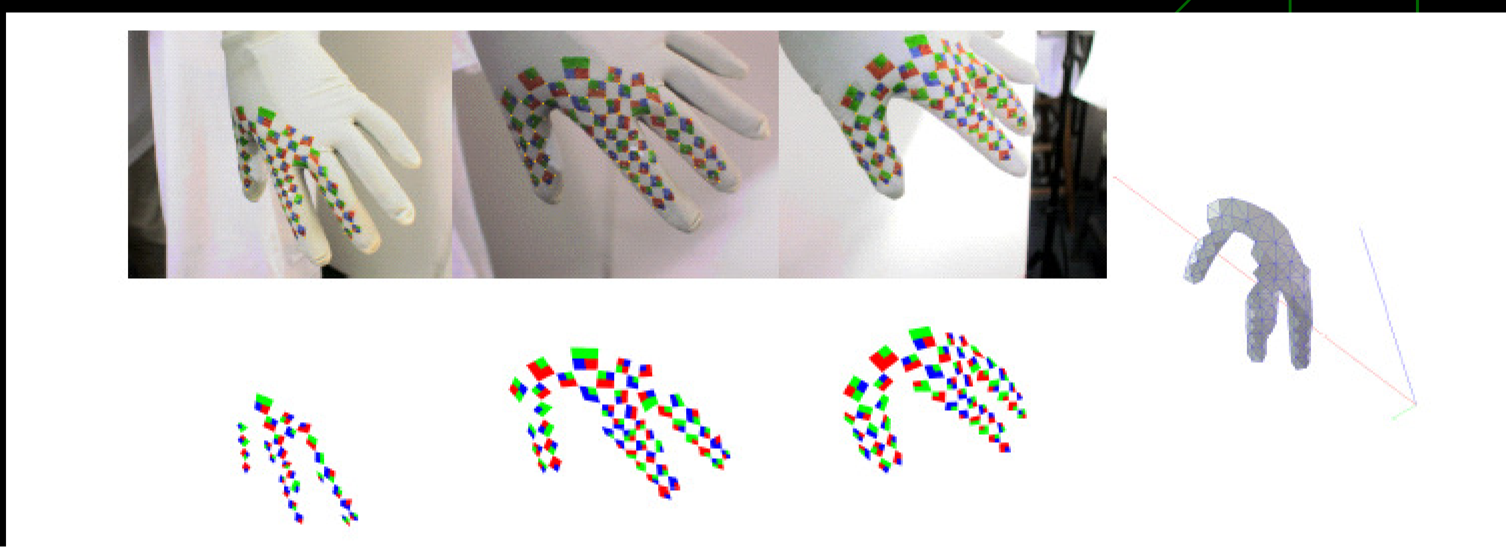
# Cloth Simulation



# Trackable Surfaces

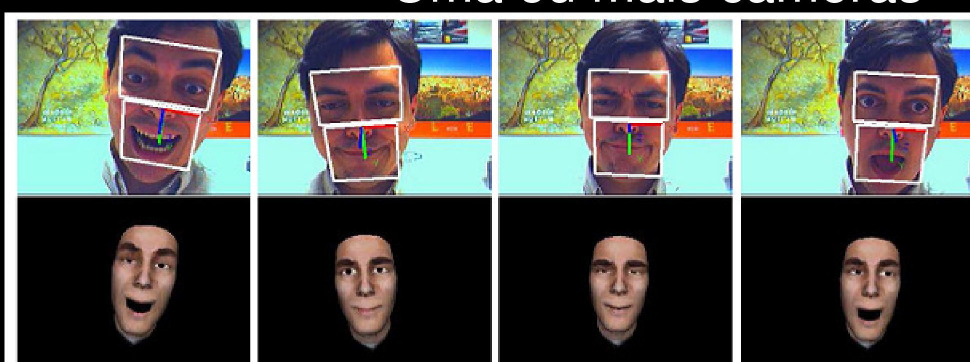
Vídeo: Guskov





# Técnicas para Animação Facial

- Performance-driven
  - Captura de pessoas reais
    - MOCAP
    - Visão Computacional
      - Com ou sem marcadores
      - Uma ou mais câmeras
      - essam

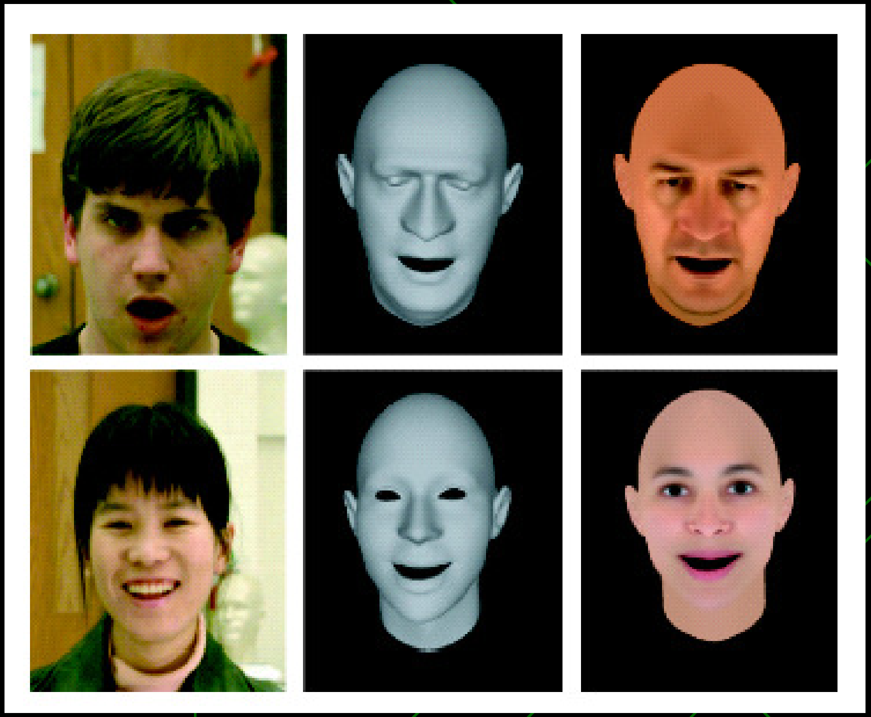


# Kara (Quantic Dreams)

- Usando marcadores...

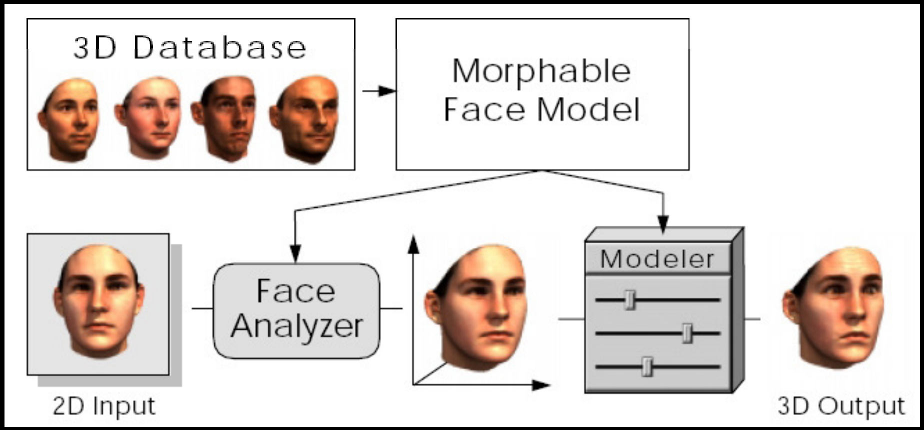
# Vision-based Control

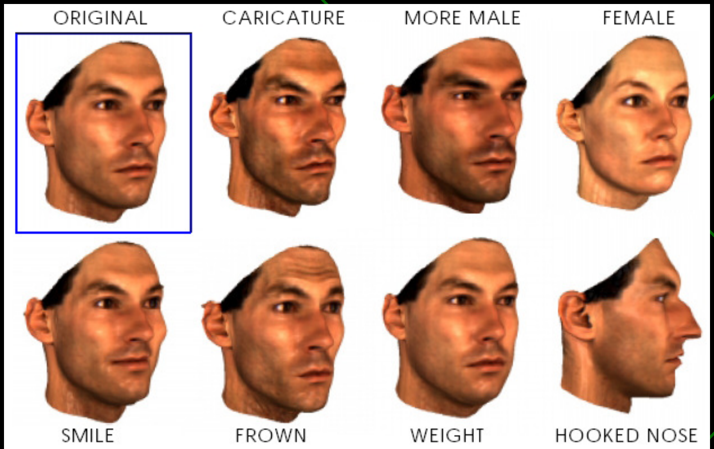
- Real-time tracking facial expressions



# Animação Facial

- Principais técnicas
  - Paramétrica/Modelos "Transformáveis"



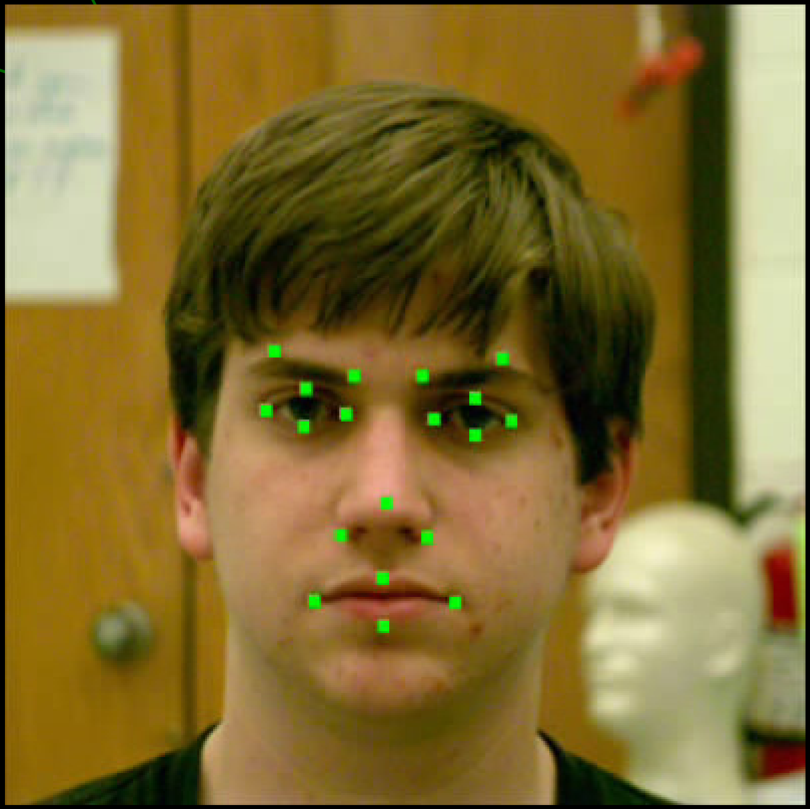

[Blanz 99]

# Com marcação, mapeamento direto

1/597

RossanaReflecting

# Sem marcação, combinando dados de MOCAP



Face.avi

# Sistema Óptico com Maquiagem

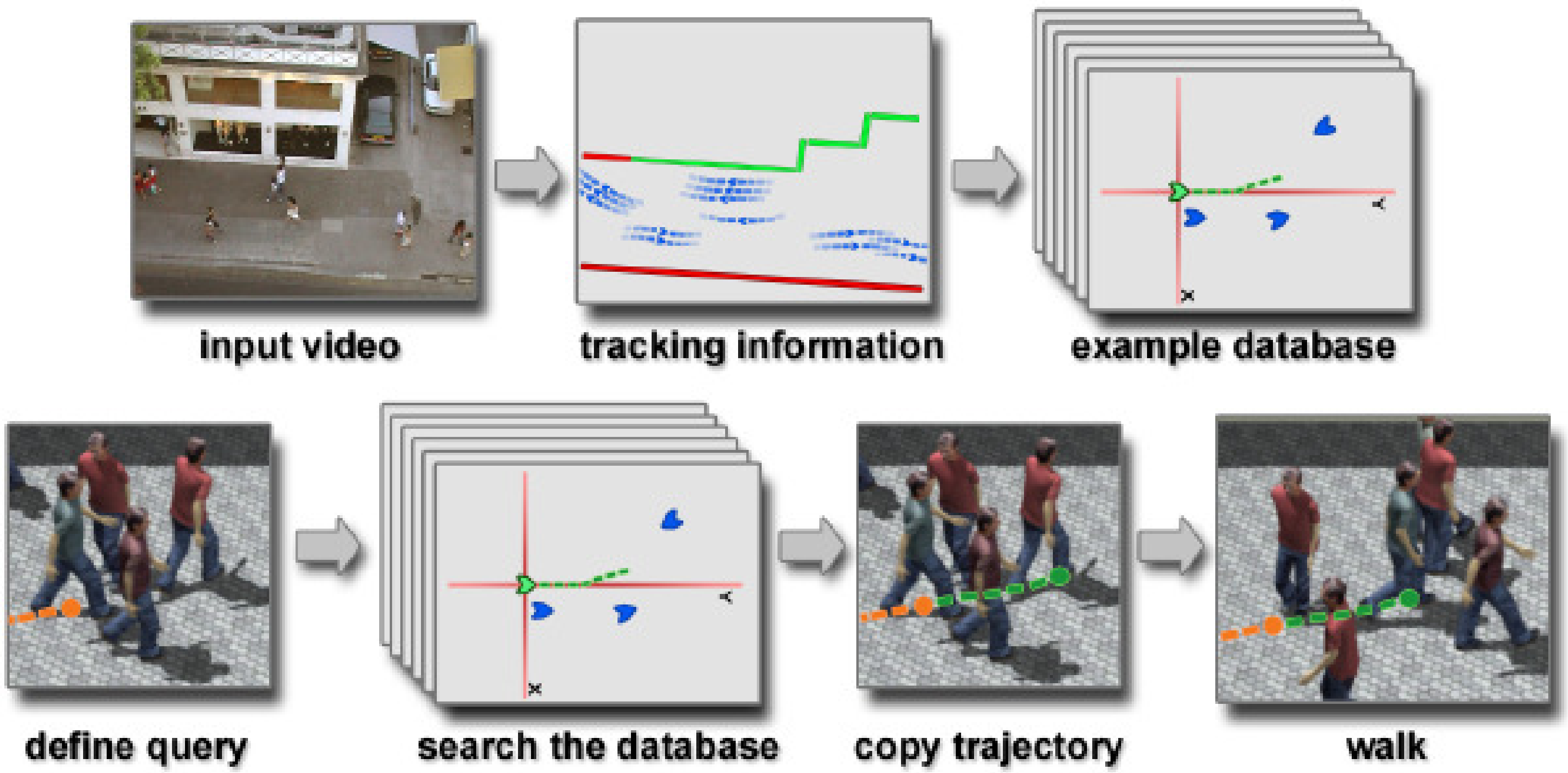

www.mova.com

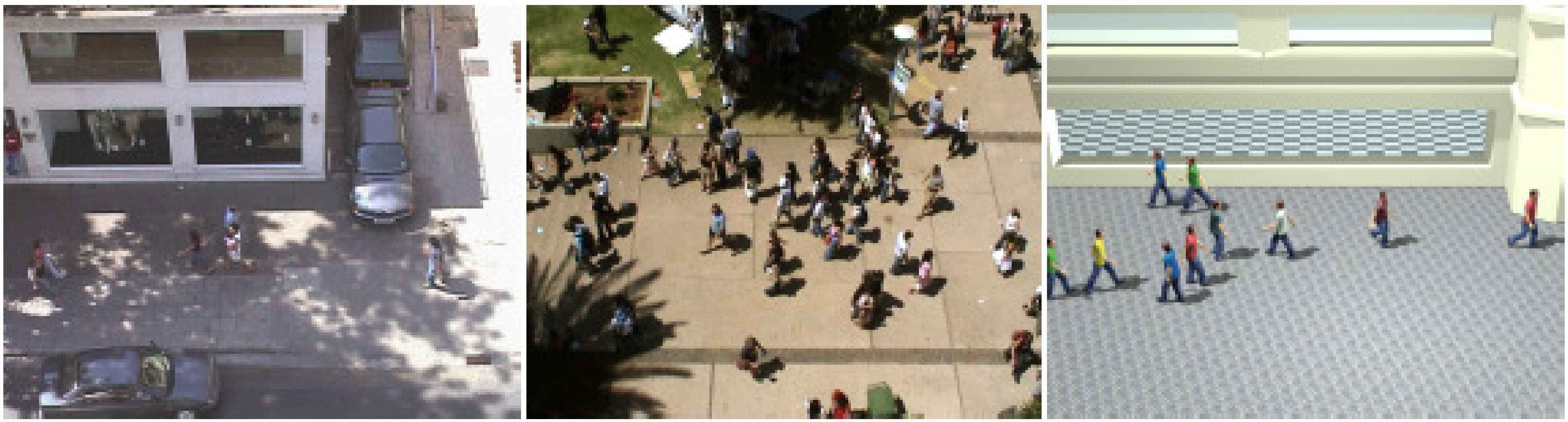
# Image Metrics

# Falando da movimentação de pessoas

- Que tal usar dados da vida real para modelar movimentos de pessoas?

# EG 2007


input video
tracking information
example database
define query
search the database
copy trajectory
walk



- Challenges:
  - Computer vision algorithms..
  - How to validate?
  - How to compare with real life?

# Outline

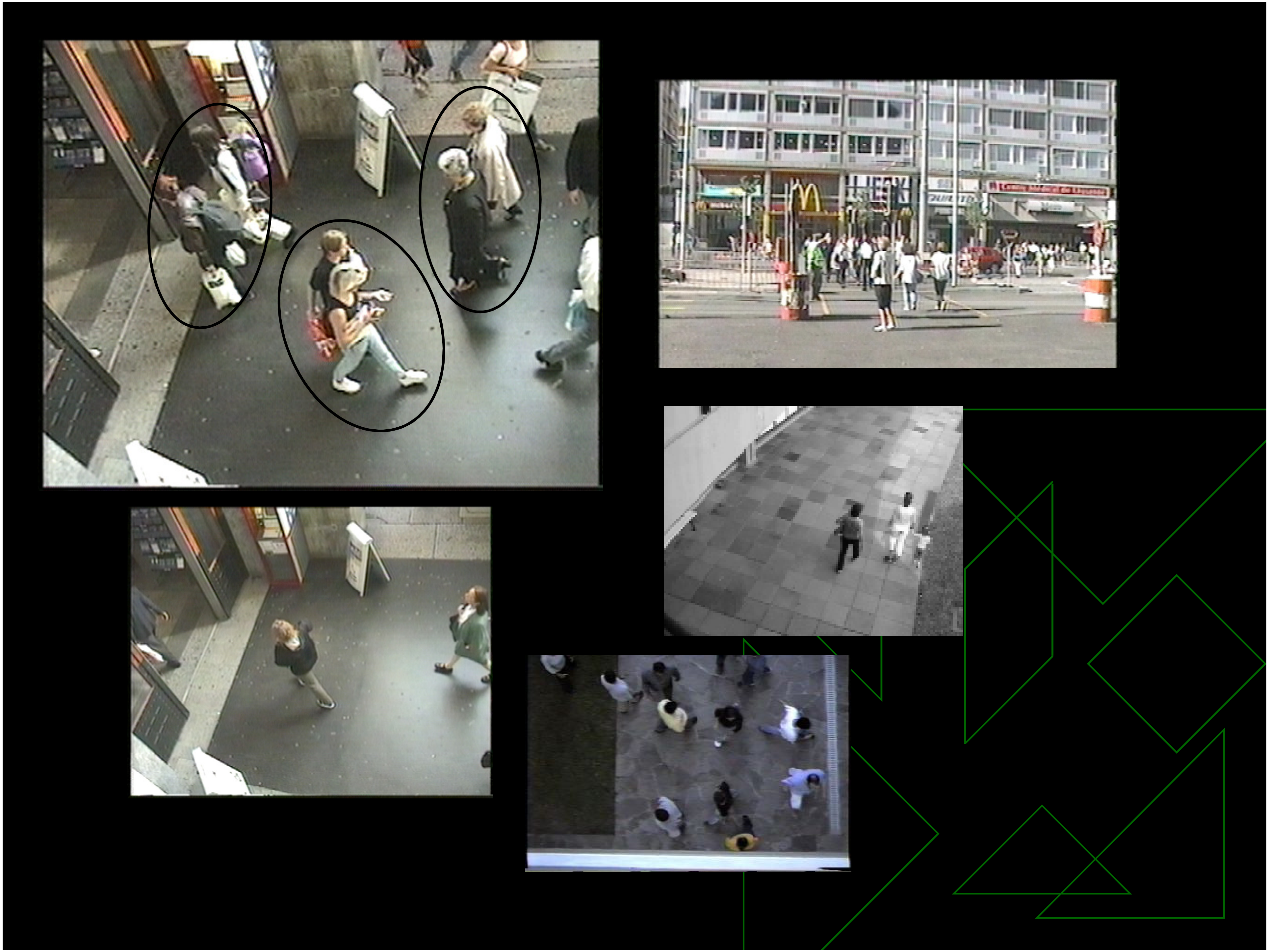
- Introduction
- Patterns of real people behaviour
- Using Computer Vision for simulating and validating crowds
- Crowd Simulation in Security Applications


# Introduction

- Important challenge is to include characteristics of real crowds into computer simulation
- ***How to characterize real crowds?***
- ***How to annotate crowd behaviors?***

# Introduction

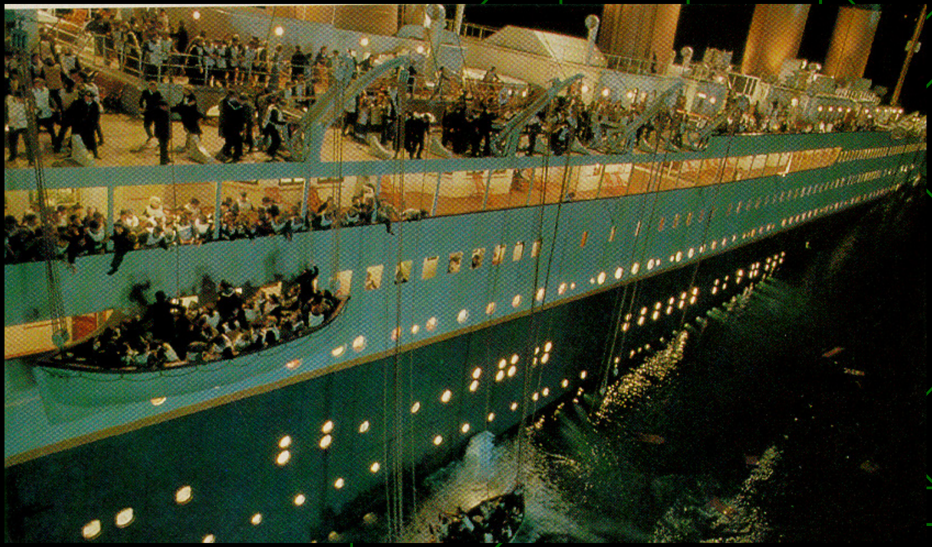
- ***Crowd Characteristics***
  - *Crowd space* (occupied space, proximity among individuals, regions where people walk),
  - *Crowd size* (number of groups and individuals inside each group),
  - *Crowd density* (relation between space and crowd sizes) – also related with crowd structure (crowds, groups and individuals)
  - *Crowd activity,*
  - *Crowd basic behaviours* (walk, grasp, look at some location, apply a posture),
  - Others…



# One example...

```
begin
    CROWD STRUCTURE
    NUMBER_PEOPLE: 100
    Density = NO-CROWDED
    GOALS_CROWD:
    LEFT LOCATION ( X Y Z )RIGHT LOCATION ( X Y Z ) (Related to the crowd space)
    ACTION_LOCATION SIT ( X Y Z )
    REGION GATE_3 (X Y Z) (X Y Z)
    NUMBER_GROUPS: 3
    BASIC BEHAVIORS
    GROUP_1NB_PEOPLE: [3,6] (Group contains from 3 to 6 individuals)
    BASIC_BEHAVIOUR: WALK from LEFT to RIGHT
    GROUP_2NB_PEOPLE: [3,6]
    BASIC_BEHAVIOUR: SITED
    GROUP_3NB_PEOPLE: [3,6]
    BASIC_BEHAVIOUR: WALK from LEFT to RIGHT
    CROWD EVENTS
    Event_1:
    WHEN: Time = 5,2 min
    WHO: ALL PEOPLE IN REGION OF GATE 3
    Reaction Event_1:
    ACTION: ENTER THE TRAIN THROUGH THE CLOSEST DOOR
end
```

# Crowd notation can work…

# How about a complex situation?

# So, we need People Tracking...

- One or more cameras?
- Color or monochromatic?
- Static or moving camera?

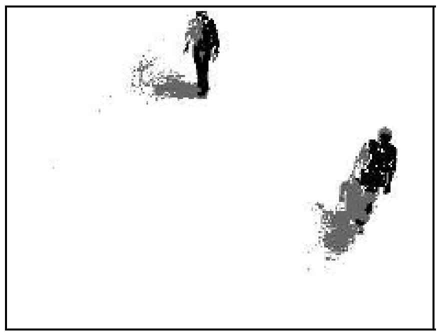
The most common approach is to use a single static camera (color or monochromatic), and the first step of tracking algorithms is typically background subtraction

# Background Subtraction

- In a few words, it consists of obtaining a mathematical model of the background, which is compared to each frame of the video sequence. Then, pixels with sufficient discrepancy are considered foreground pixels, and sets of connected pixels are usually called blobs.

# Background Subtraction

- One problem inherent to background subtraction is the undesired detection of shadows (or highlights) as foreground objects. Indeed, shadows may connect isolated people in a scene, generating a single blob and probably compromising the performance of the tracking algorithm.



# Shadows and background adaptation

- So, we need algorithms for shadow detection
- Another desired characteristic for background removal is adaptation to changes in the background.

# An Approach for Crowd Simulation Using Computer Vision (CAVW 2007)

- Overview of the method:
  - Use computer vision algorithms to track the trajectory of each filmed individual
  - Group coherent trajectories into “motion clusters”, based on the main direction of each trajectory
  - Compute an extrapolated velocity field for each motion cluster
  - Apply a crowd simulator that uses the extrapolated velocity fields to guide virtual humans

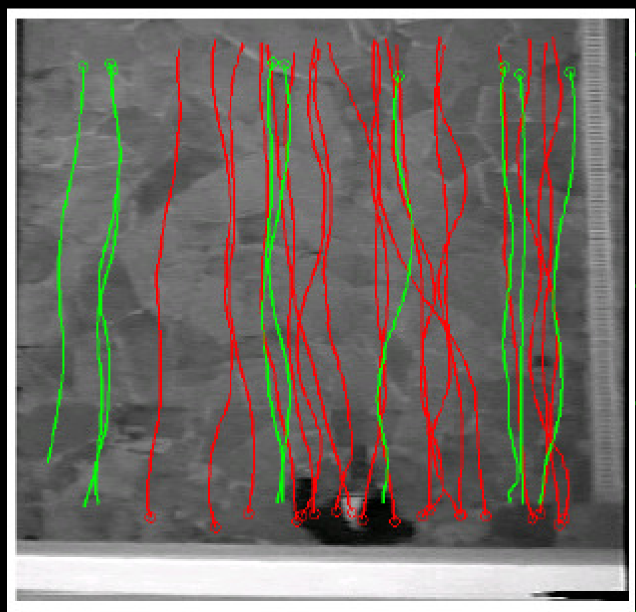
# Clustering Approach

$$\mathbf{d}_i = (x(t_{i+1}) - x(t_i), y(t_{i+1}) - y(t_i)),$$

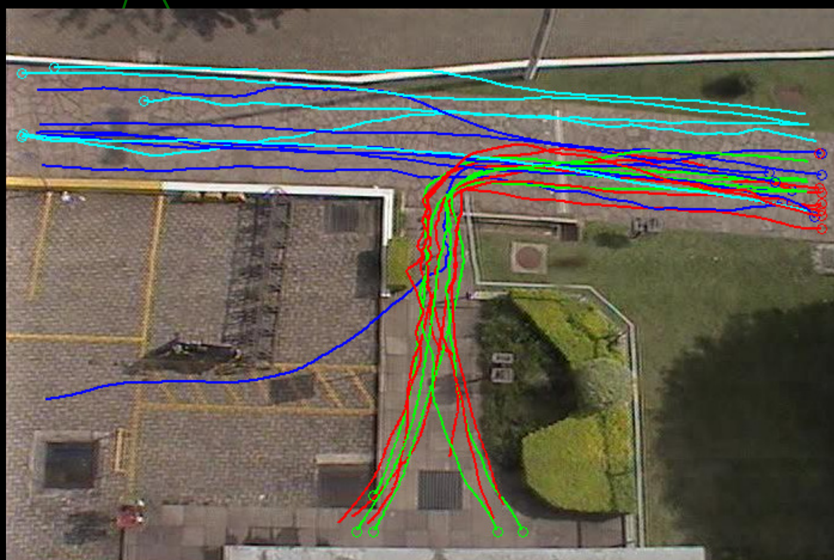
Displacement vector

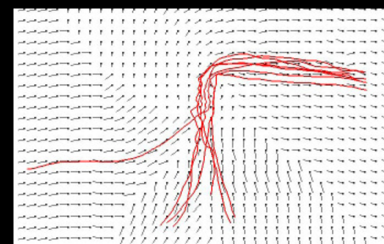
$$\mathbf{f}_j = (\Delta x_0, \Delta y_0, \Delta x_1, \Delta y_1, \cdots, \Delta x_{N-1}, \Delta y_{N-1}).$$

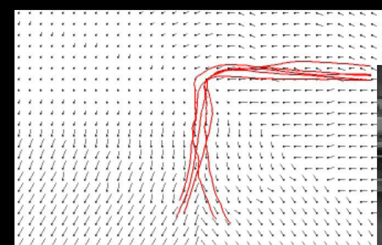
Feature vector

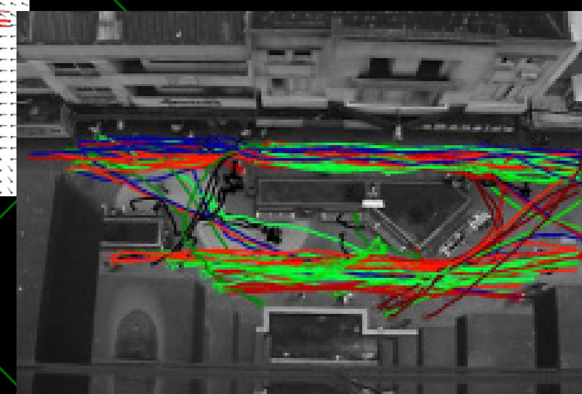


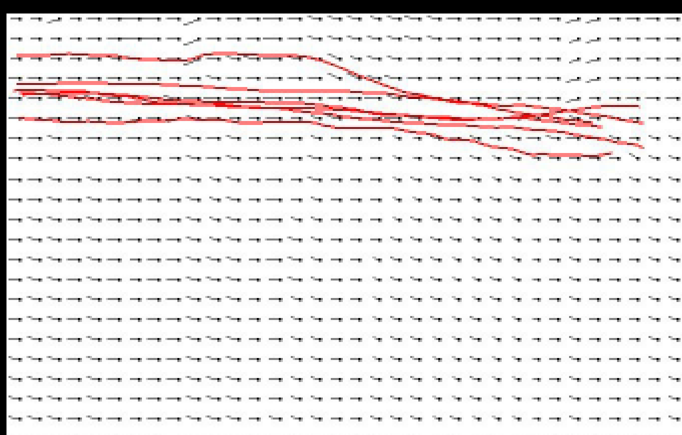
# Clustering Approach

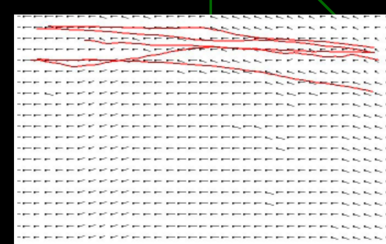




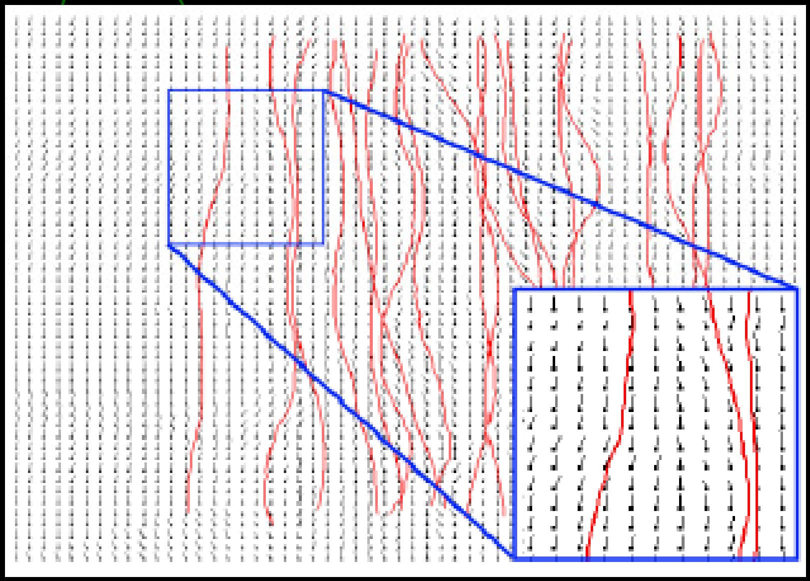


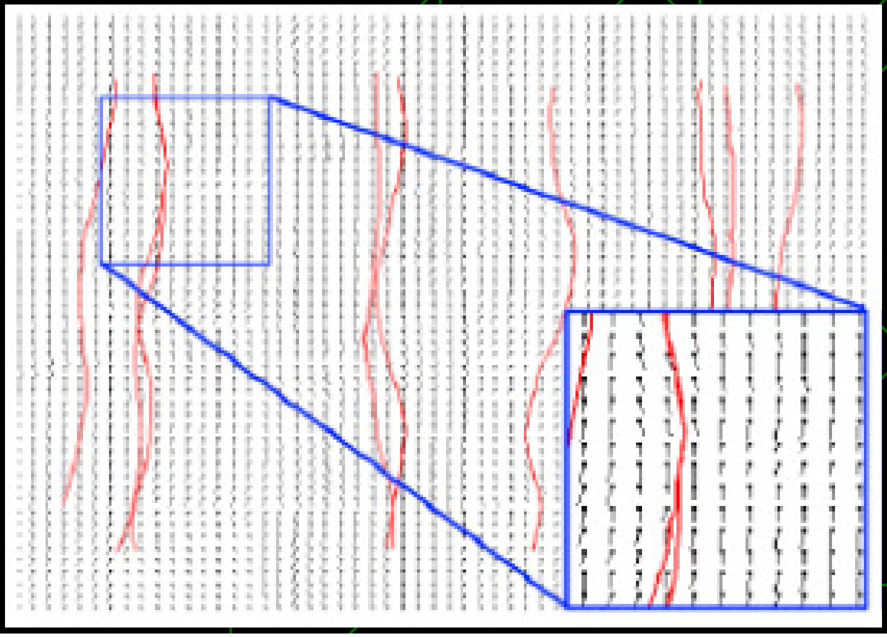






# Computing an extrapolated velocity field for each cluster





# Experimental Results

- Integration with Crowd Simulator

$$m_i \frac{d\mathbf{v}_i}{dt} = m_i \frac{\mathbf{v}_i^g - \mathbf{v}_i(t)}{\tau_i} + \sum_{j \neq i} \mathbf{f}_{ij} + \sum_{w} \mathbf{f}_{iw}$$

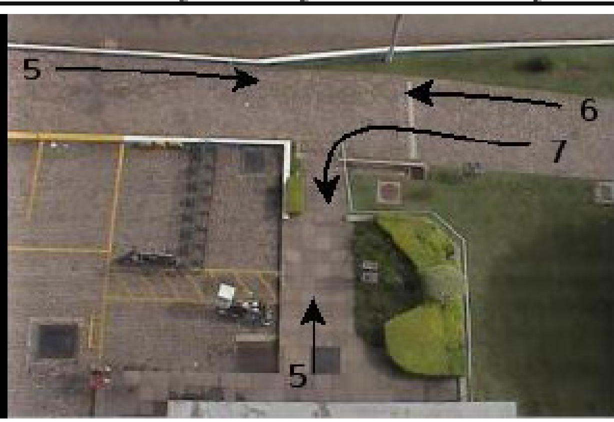
Where $v_i^g$ comes from extrapolated vector field

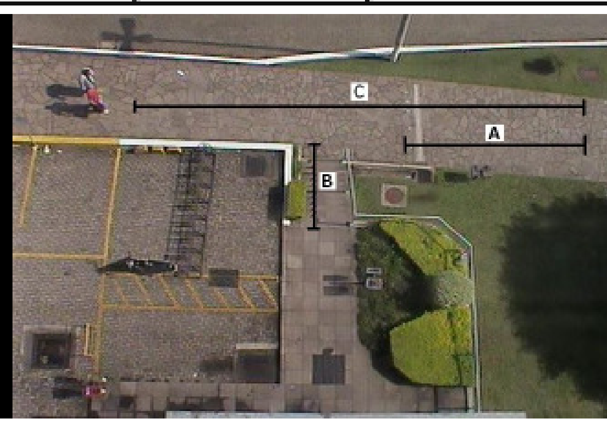
If it's an emergency situation, then it points to exits

# Experimental Results

- Simulating using 23 virtual agents

| Region | Dir. | Video (speed) | | Simulation (speed) | |
|---|---|---|---|---|---|
| | | mean | std | mean | std |
| A | → | 0.96m/s | 0.17m/s | 0.84m/s | 0.26m/s |
| | ← | 1.00m/s | 0.19m/s | 0.91m/s | 0.21m/s |
| B | ↓ | 0.52m/s | 0.33m/s | 0.48m/s | 0.30m/s |
| | ↑ | 0.53m/s | 0.29m/s | 0.58m/s | 0.29m/s |
| C | → | 1.03m/s | 0.20m/s | 0.89m/s | 0.27m/s |
| | ← | 1.06m/s | 0.20m/s | 0.99m/s | 0.23m/s |



# Experimental Results

- Simulating using 70 virtual agents

| Região | Direção | Vídeo (vel.) | | Simulação (vel.) | |
|---|---|---|---|---|---|
| | | média | desvio | média | desvio |
| A | $\rightarrow$ | 0.95998m/s | 0.17493m/s | 0.86385/s | 0.31848m/s |
| | $\leftarrow$ | 1.0024m/s | 0.19033m/s | 0.86917m/s | 0.2945m/s |
| B | $\downarrow$ | 0.40749m/s | 0.23703m/s | 0.41637m/s | 0.34696m/s |
| | $\uparrow$ | 0.41951m/s | 0.18849m/s | 0.42659m/s | 0.28291m/s |
| C | $\rightarrow$ | 1.028m/s | 0.20061m/s | 0.89804m/s | 0.33736m/s |
| | $\leftarrow$ | 1.0625m/s | 0.20418m/s | 0.94548m/s | 0.31595m/s |