Tracking II

2014-01-20

Overview

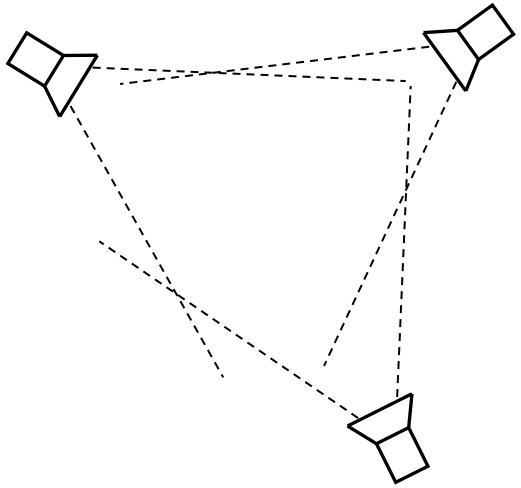
- Multi-camera systems
- Examples:
  - 3D Multi-object Tracking
  - Audio-Visual Tracking of a Lecturer
  - Multi-person tracking on a robot
- Articulated Body Tracking

Multi-Camera Systems

Multi-Camera Topologies

Wide-baseline multi-camera system

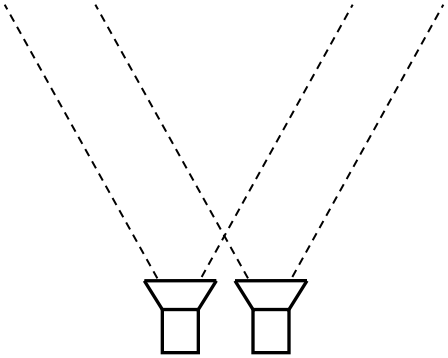
- Arbitrary distance and orientation, overlapping field of view
- An object's appearance is different in each of the cameras
- Allows for 3D localization of objects in the joint field of view

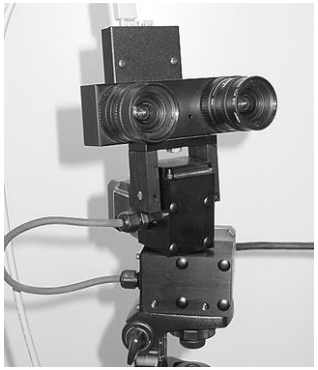


Multi-Camera Topologies

Stereo-camera system (narrow baseline)

- Close distance and equal orientation
- An object's appearance is almost the same in both cameras
- Allows for calculation of a dense disparity map

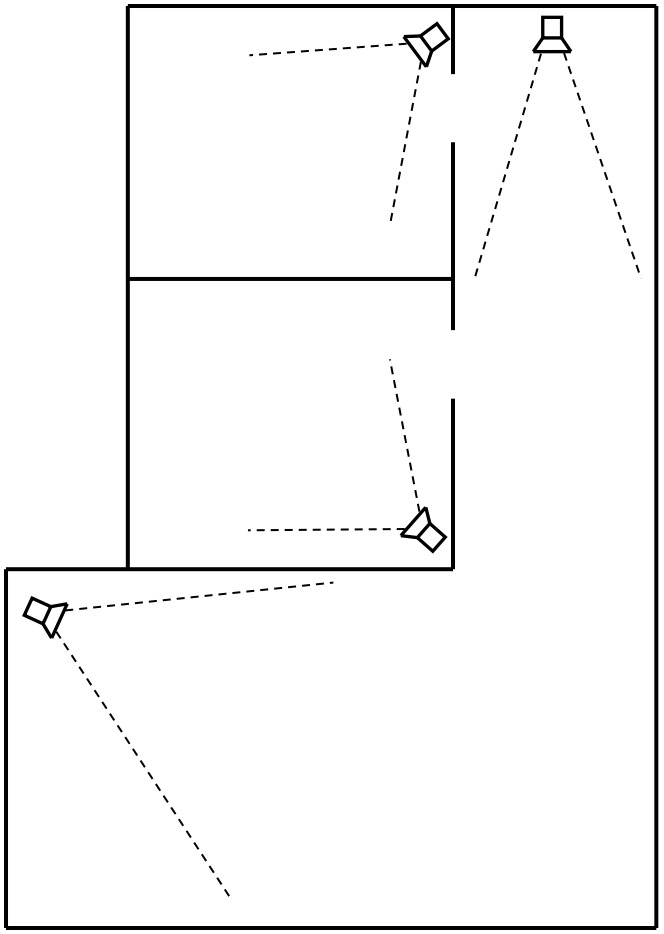




Multi-Camera Topologies

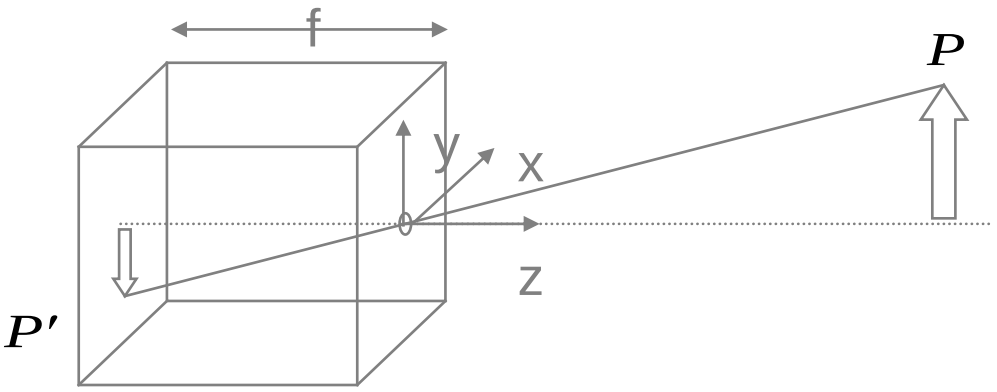
Multi-camera network

- Non-overlapping field of view
- An object's appearance differs strongly from one camera to another



3D to 2D projection: Pinhole Camera Model

$$P' = \begin{pmatrix} x' \\ y' \\ z' \end{pmatrix}$$



$$P = \begin{pmatrix} x \\ y \\ z \end{pmatrix}$$

- due to similar triangles and the intercept theorems:

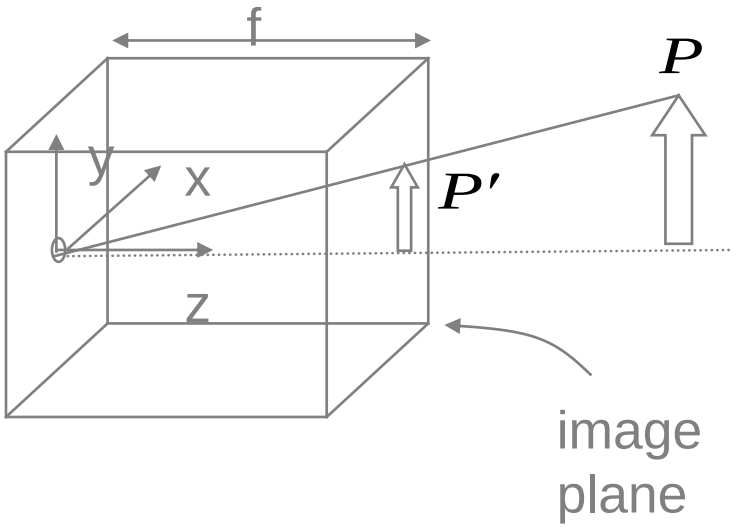
$$z' = -f$$

$$\frac{y'}{-f} = \frac{y}{z} \quad \Rightarrow \quad y' = \frac{-fy}{z}$$

$$\frac{x'}{-f} = \frac{x}{z} \quad \Rightarrow \quad x' = \frac{-fx}{z}$$

Mathematical Simplification

- translation of the image plane in front of the focal point



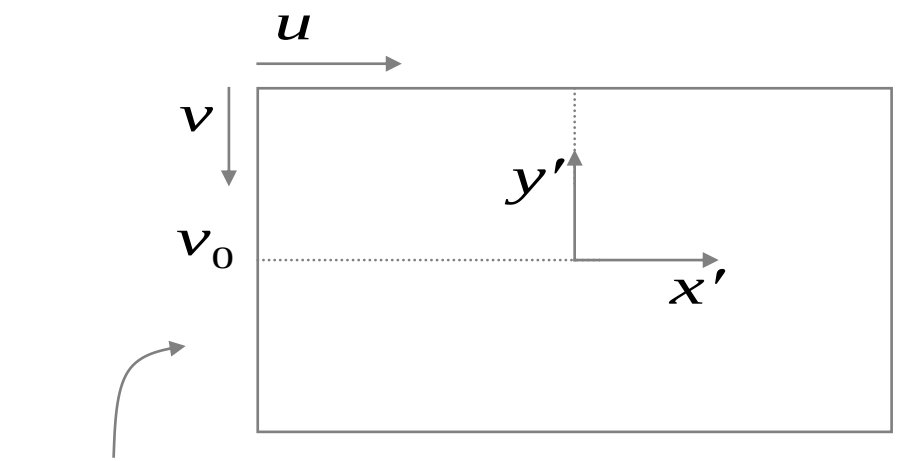
$$z' = f$$

$$x' = \frac{fx}{z}$$

$$y' = \frac{fy}{z}$$

Completion of the Projection

- so far:
  - projection of a point P onto the image plane
- what's missing:
  - pixel-coordinates (u,v) of the projected points



$$u = k_u x' + u_0$$
$$v = -k_v y' + v_0$$

With k_u and k_v scaling factors which denote the ratio between world and pixel coordinates

In matrix formulation:

$$\begin{pmatrix} u \\ v \end{pmatrix} = \begin{pmatrix} k_u & 0 \\ 0 & -k_v \end{pmatrix} \begin{pmatrix} x' \\ y' \end{pmatrix} + \begin{pmatrix} u_0 \\ v_0 \end{pmatrix}$$

Perspective Projection

- 4 parameters:

$$\alpha_u = k_u f$$
$$\alpha_v = -k_v f$$
$$u_0$$
$$v_0$$

  - have to be known to perform the projection
  - called ‘internal camera parameters’ since they depend on the camera only
  - perform calibration to estimate those parameters

  - (in addition, distortion parameters k_1, … k_n needed to model lens distortions)

Calibration - Intrinsics

The intrinsic parameters describe the optical properties of each camera ("the camera model"). Typically, this includes:

- f — the focal length
- c_x, c_y — the principal point ("optical center") (*)
- K_1.. K_n — distortion parameters (radial and tangential)

- (*) denoted as u_0, v_0 in previous slides

Calibration - Extrinsics

The extrinsic parameters describe the location of each camera with respect to a global coordinate system:

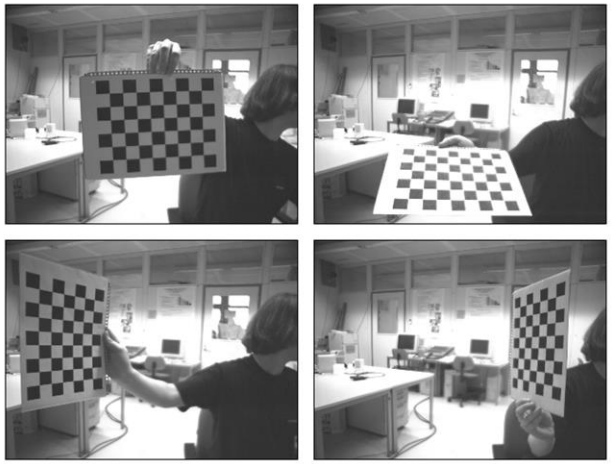
- T the translation vector
- R the 3x3 rotation matrix

Transformation of world coordinates of point p* = (x,y,z) to camera coordinates p:

- $p = R\,(x\ y\ z)^T + T$

Camera Calibration

1. For each camera: A calibration target with a known geometry is captured from multiple views
2. The corner points are extracted (semi-)automatically
3. The locations of the corner points are used to estimate the *intrinsics* iteratively



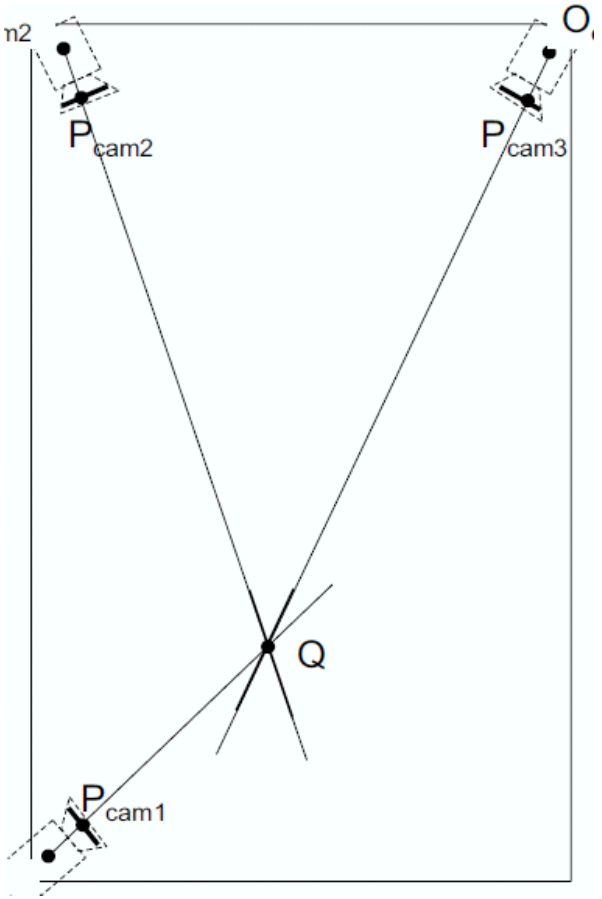
4. Once the intrinsics are known, a fixed calibration target is captured from all of the cameras → *extrinsics*

A camera calibration toolbox for Matlab

→ http://www.vision.caltech.edu/bouguetj/calib_doc/

Triangulation

- Assumption: the object location is known in multiple views
- Ideally: The intersection of the lines-of-view determines the 3D location
- Practically: least-squares approximation



Sensor Fusion

How to integrate data from multiple cameras?

- Run a dedicated tracker on each camera's data and combine the trackers' output.
- Fuse the data from all cameras and run one tracker on the joint data representation.
- Run one tracker and evaluate it's hypotheses on each camera's data separately.

Multi-Object Tracking

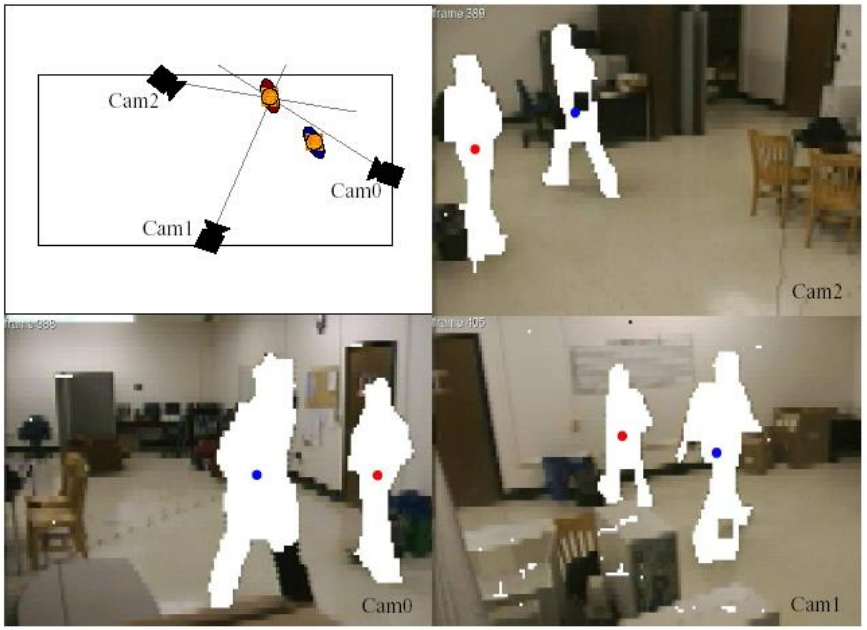
Multi-object Tracking

Two different approaches

- A dedicated tracker for each of the objects
  - Information has to be shared across trackers to find a good assignment
  - Typically fast and well parallelizable
- A single tracker in a joint state space
  - Easier to find optimal assignment
  - More complex due to the high dimensionality of the state space

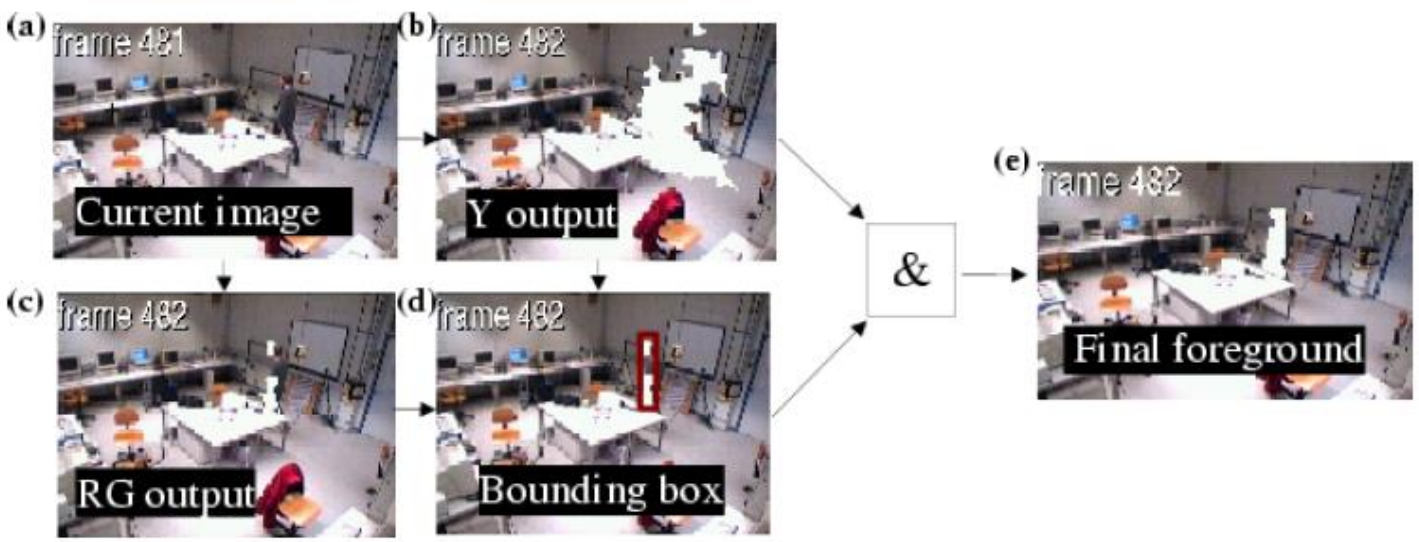
3D Multi-Person Tracking [Focken02]

- Target:
  multiple persons
- Sensors:
  multiple fixed cameras
- Feature:
  foreground segmentation
- Tracking scheme:
  Kalman-Filter



Foreground Segmentation

Mixed foreground modeling in Y and rg to reduce effects of shadow



Triangulation

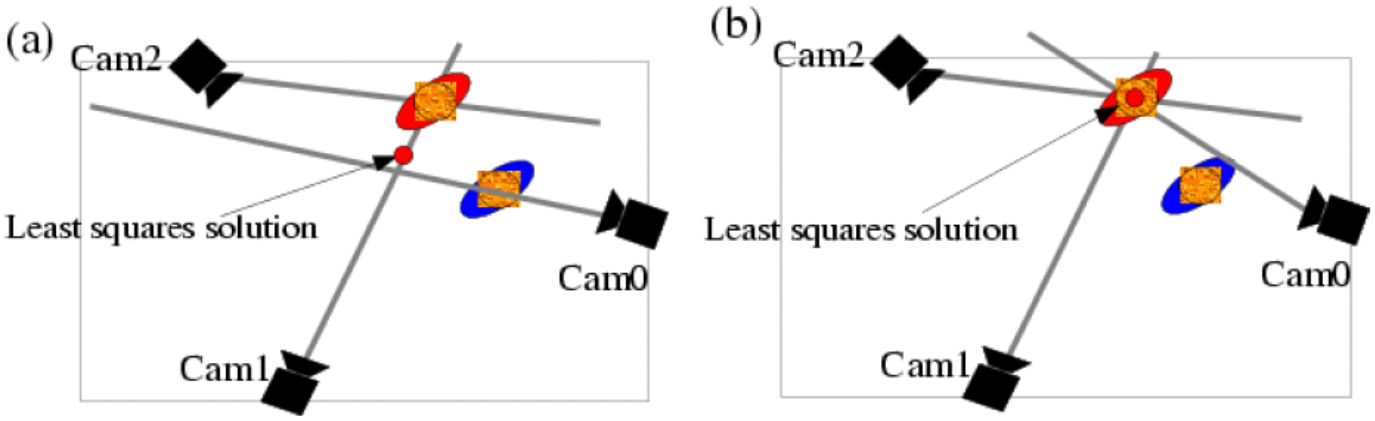
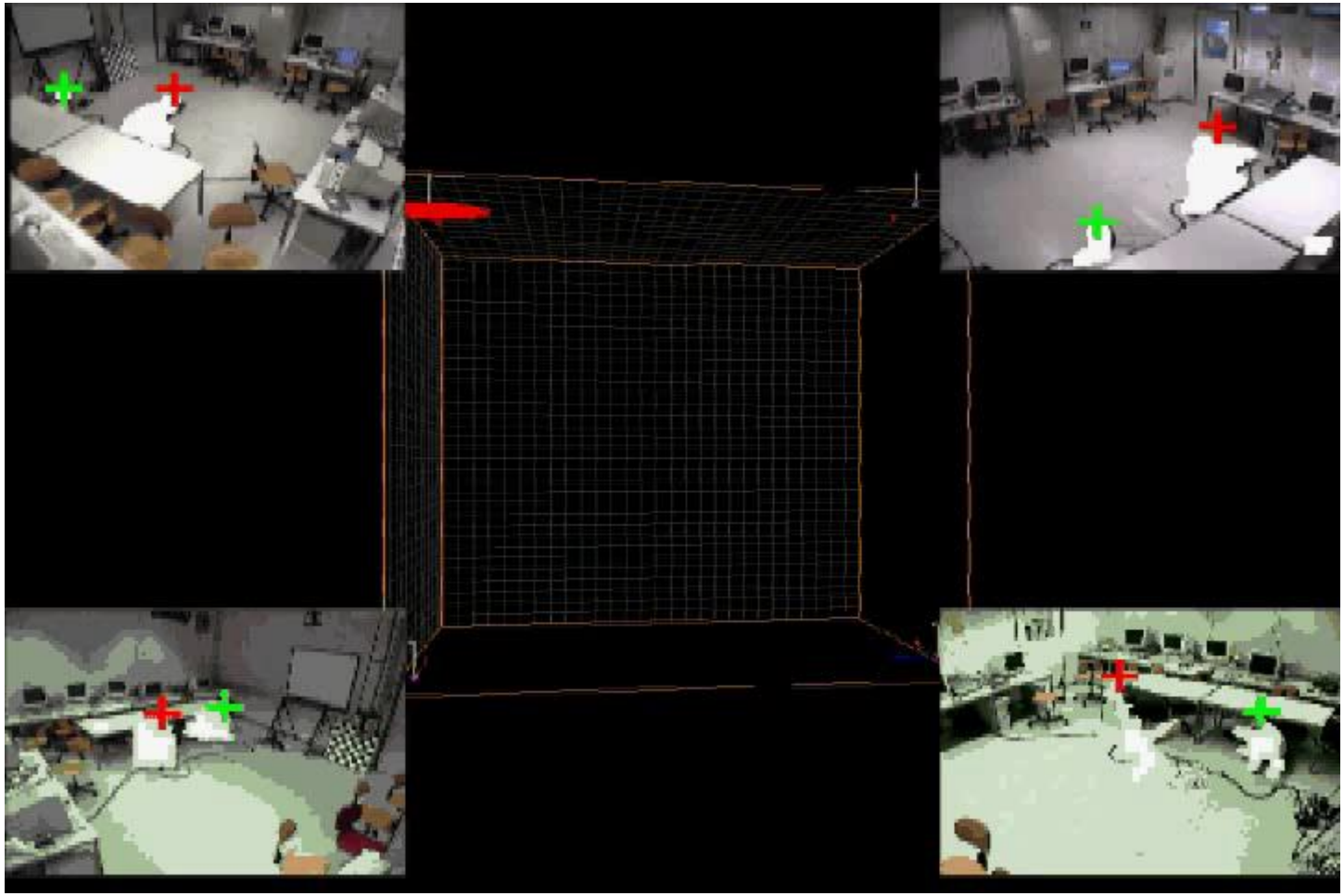


Figure 3. (a) A wrong correspondence (b) A correct correspondence

- In order to generate 3D measurements, all possible correspondences between foreground regions from different cameras are searched
- The triangulation results are sorted by the number of foreground regions supporting it, and by the residual error

Tracking Scheme

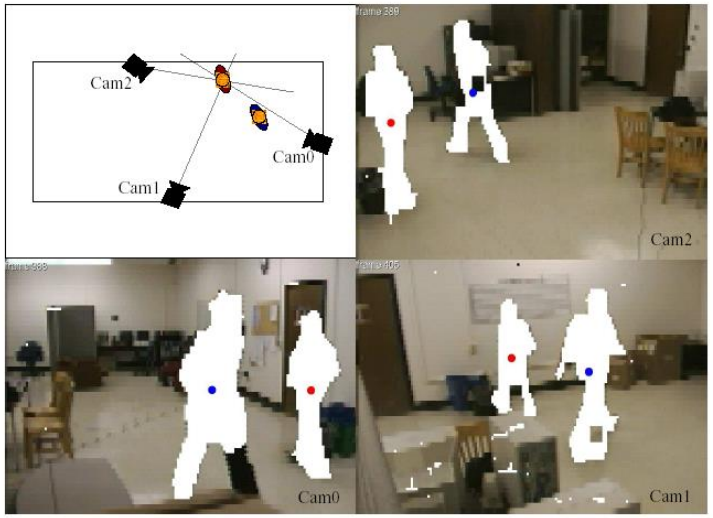
- Each person is modelled by a dedicated Kalman-Filter Tracker using a position-velocity motion model
- The observations are matched to the trackers such that the overall distance between prediction and measurement is minimal
- For each un-matched measurement that lasts longer than a given threshold, a new tracker is generated
- If a tracker is not supported by measurements for longer than a given threshold, the tracker is deleted



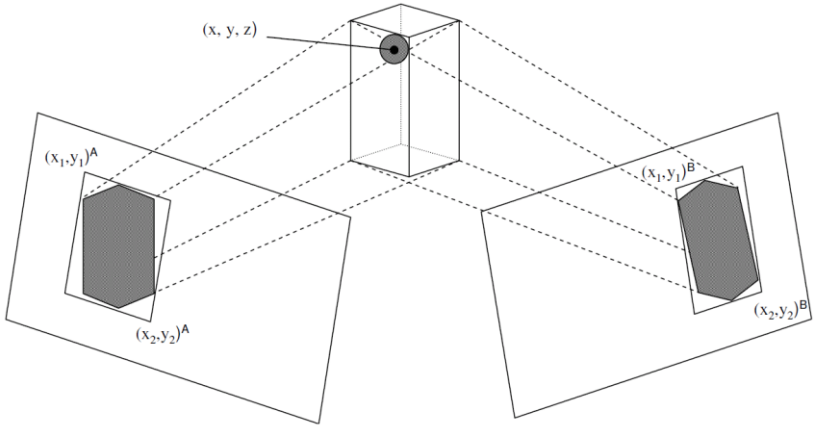
Video

Multi-camera tracking without explicit triangulation

- Problems:
  - Correspondence problem in triangulation: there are many possible matches
  - Finding the correct match is time consuming
  - Can lead to errors



- Solution: Avoid explicit triangulation!
  - For each 3D hypothesis, check for supporting observations in all camera views
  - Can be nicely put into a particle filter framework



Audio-Visual Tracking of a Speaker

- **Target**: the speaker in a lecture
- **Sensors**: multiple fixed cameras and microphones
- **Features**: background subtraction, face and upper body detection, GCC of the audio signal
- **Tracking Scheme**: Particle Filter

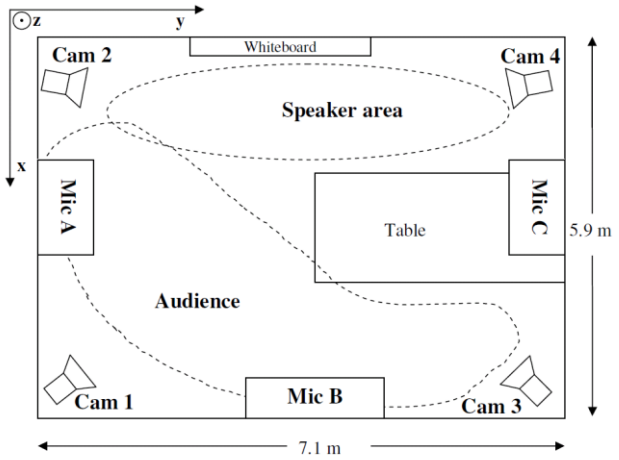


Figure 4: The lecture room is equipped with four fixed cameras and three 4-channel microphone arrays. Lecturer and audience typically reside in the depicted areas, but are not limited to these areas.



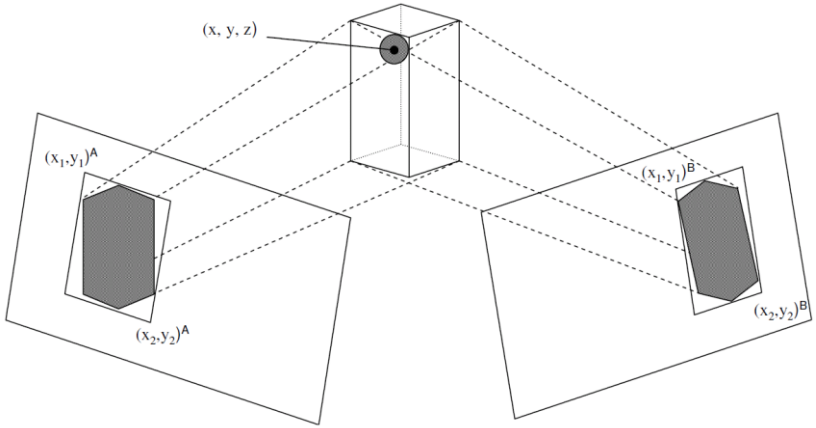

Tracking Scheme

Condensation Algorithm with …

- **State space**: the speaker's head centroid $s_i = (x, y, z)$
- **Motion model**: particles are propagated using Gaussian diffusion ("0-th order" motion model)
- **Observation model**: a particles's weight π_i is determined by local visual and acoustical information gathered from all cameras and from all microphone pairs

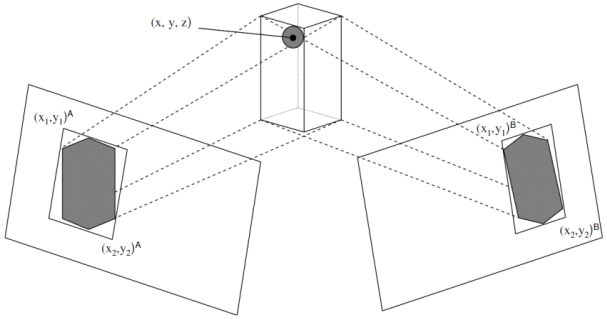
Visual Features - Foreground

- A person-sized cuboid around s_i is projected to all views.
- The percentage of foreground pixels inside the bounding box determines the score π_i
- When the projected boxed are forced to be orthogonal to the image coordinate system, the sum of foreground pixels can be calculated using the *integral image*



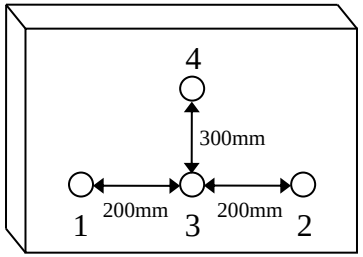
Visual Features - Detectors

- A head-sized cuboid around s_i is projected to all views.
- A face detector is run *once* on the projected box
- If a face is found, π_i increases
- The same is done using an upper-body detector

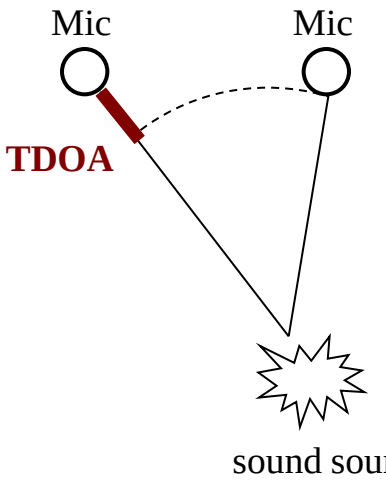


Acoustic Features

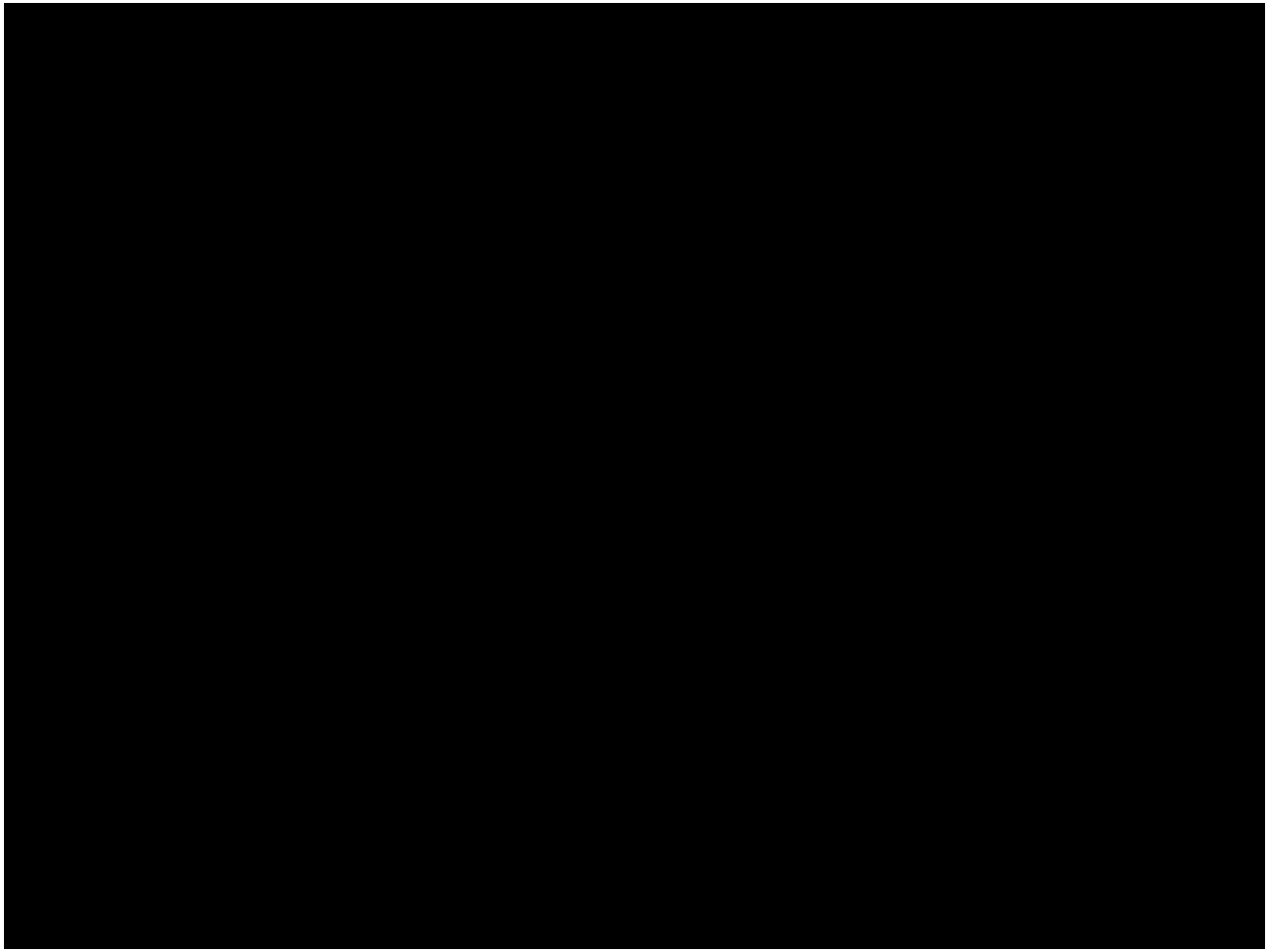
- Given a pair of microphones at know locations
- Each position s_i represents a hypothetical *time-delay-of-arrival* (TDOA)
- Using the *generalized cross correlation* function (GCC), it can be checked, whether there is high or low signal correlation given the TDOA
  → π_i



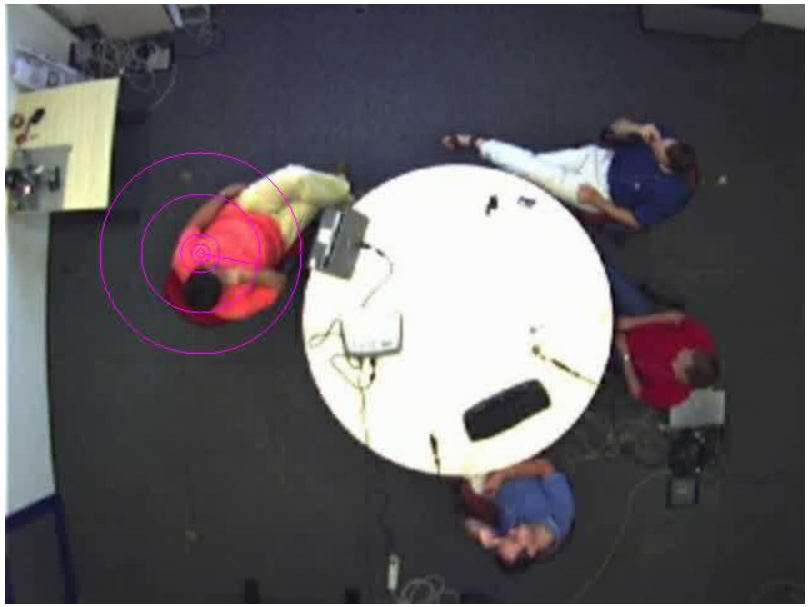
4ch Microphone array



Video



Multi-camera, multi-person tracking with particle filters



- One particle filter to explore the room
- One particle filter per person
- Used Observations:
  - Head and upper body detectors, foreground, color
  - Audio: Time delay of arrival (TDOA) / Gen. Cross-Corr.
- Runs in real-time

Personen-Tracking für einen Roboter

Herausforderungen:

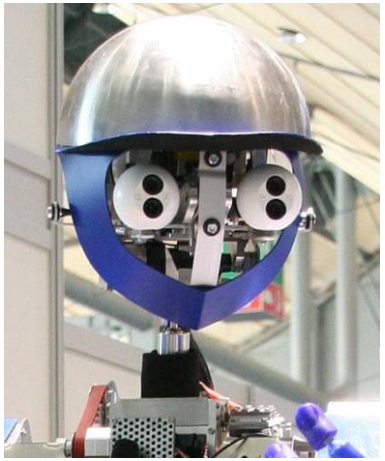
- Dramatische Veränderung des Ausssehens, je nach Distanz
- Veränderliche Beleuchtung
- Begrenztes Sichtfeld
- Mobile Plattform
- Echtzeitanforderung

Ansatz:

- Tracking als Zustandsschätzung
  - → $p(s_t \mid z_{0:t}) \approx p(z_t \mid s_t)\, p(s_t \mid s_{t-1})$
  - → gelöst mit Partikelfilter
- Dynamische Gewichtung der Merkmale
  - Kopf, Oberkörper, Beine
  - Farbe, Bewegung, Disparität, Detektoren

0.6m 4.0m

Experimental Head
- 2 Cameras
- 8 Microphones
- Pan/Tilt Unit



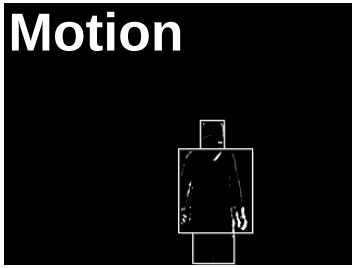
ARMAR III
- 4 Cameras
- 6 Microphones
- Anthropomorphic neck

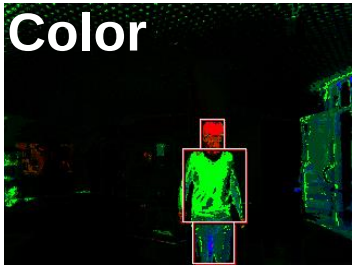
Dynamische Merkmalsgewichtung

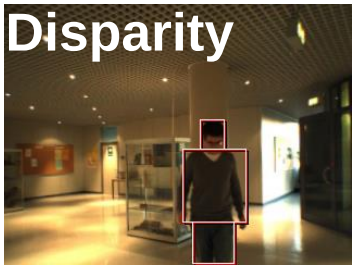
- Beobachtungsmodell besteht aus 13 gewichteten, konkurrierenden Merkmalen
  - → $p(z_t \mid s_t) = \sum_{c=1..C} r_c \, p_c(z_t \mid s_t)$
- Gewichtung durch „Democratic Integration“[Triesch & v.d.Malsburg 2001]
  - Merkmale, die mit dem Gesamtergebnis “übereinstimmen”, bekommen mehr Gewicht
  - Bewertet automatisch aktuell zuverlässige Merkmale / Regionen

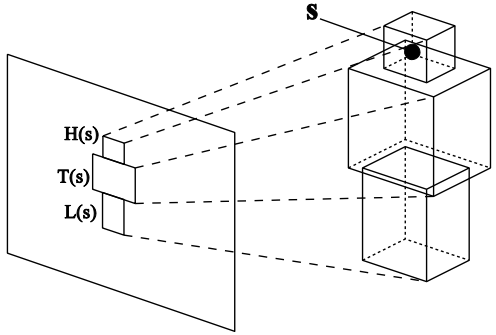
  → robuste Merkmale

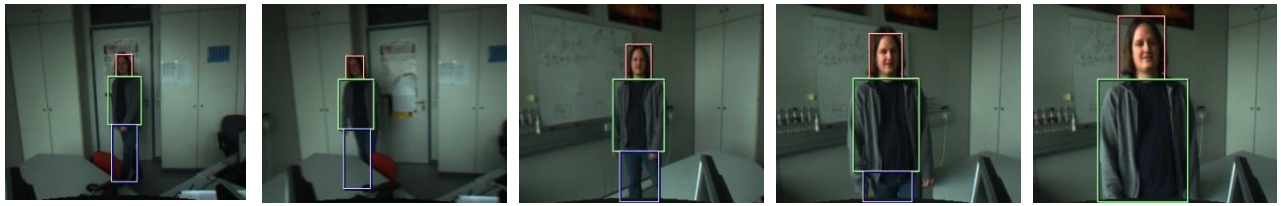
  → Behandlung von Verdeckungen











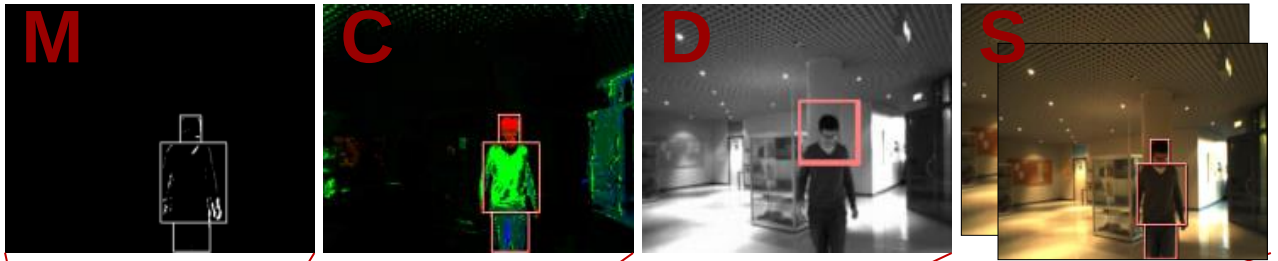
Merkmale

Bewegung (Motion)

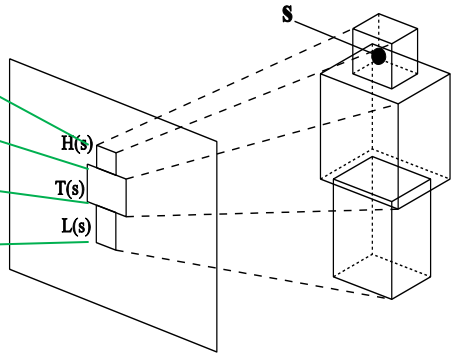
Farbmodelle (Color)

Kopf- und Oberkörper-detektoren

Stereo Korrelation

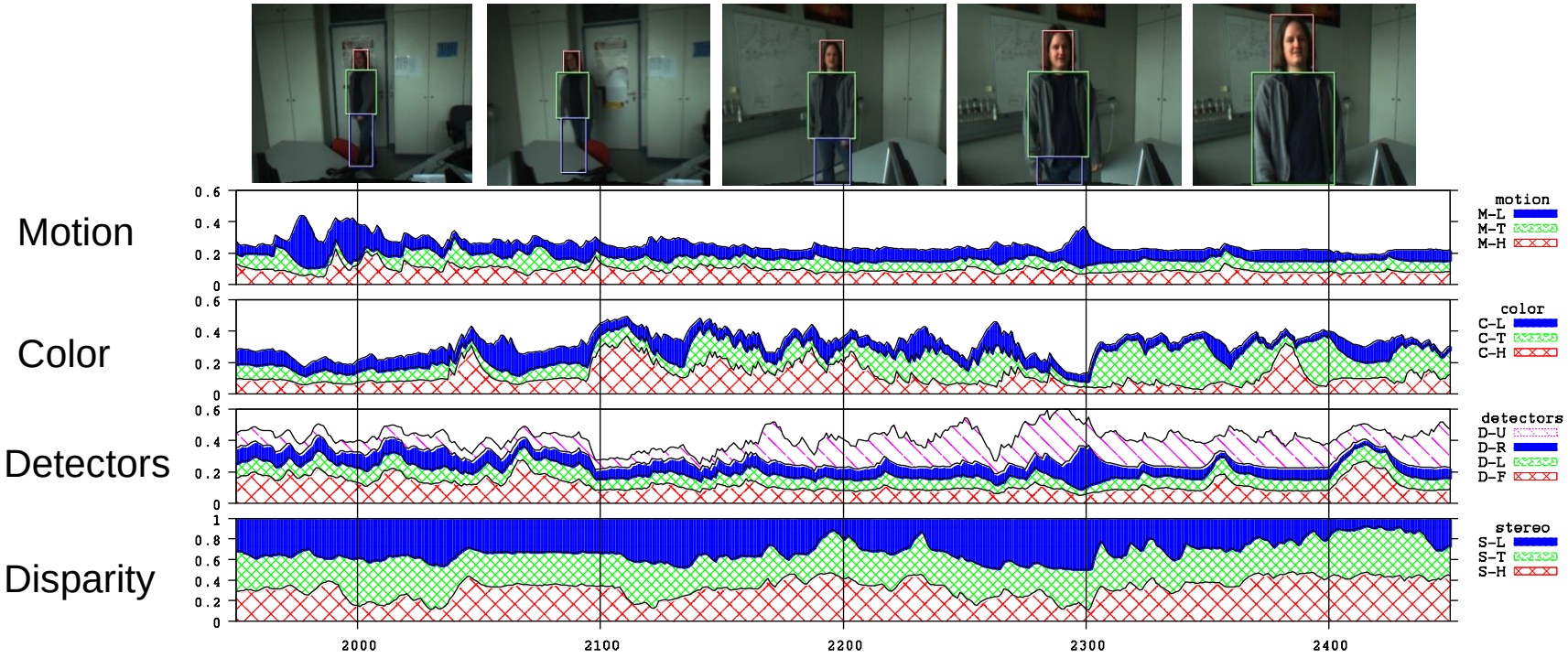


M-H	C-H	D-H_{1-3}	S-H
M-T	C-T	D-T	S-T
M-T	C-L		S-L



Video: Personentracking Roboter

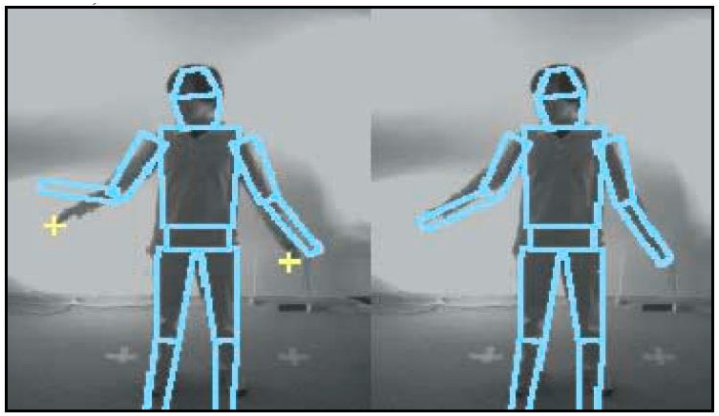
Adaptive Merksmalsgewichtung

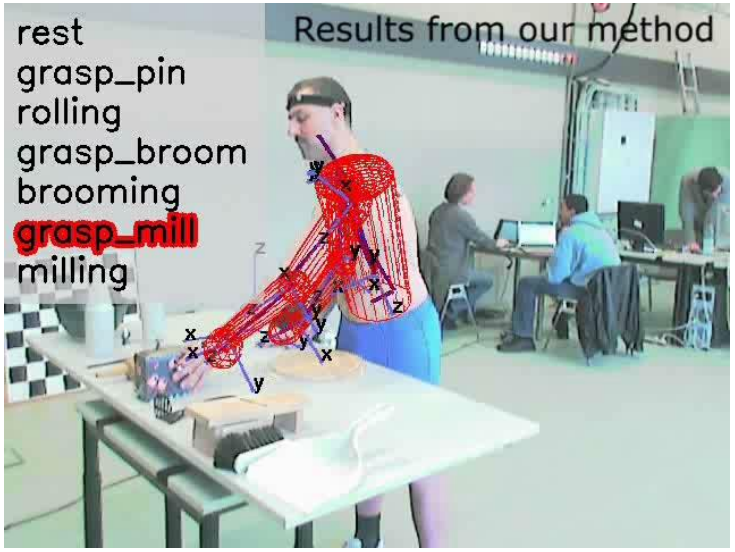


~ Frame 2000: Person beginnt zu laufen → motion cues +
~ Frame 2100-2150: Kamerabewegung / motion blur → Detektoren –
~ Frame 2200: Beine nicht sichtbar wegen Verdeckung
~ Frame 2400: Beine nicht sichtbar wegen Nähe zur Kamera

Articulated Body Tracking

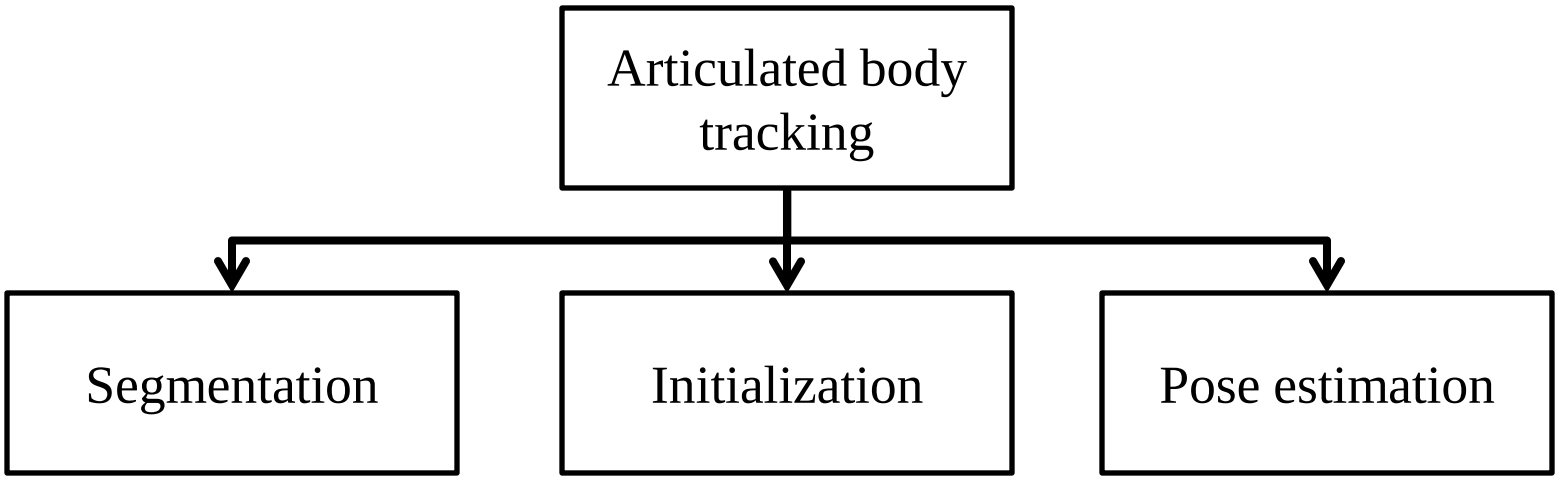
- Automatic capture and analysis of large scale body movements over time
  - including movements of legs, arm, head, torso, …
  - In contrast to tracking of *rigid* bodies



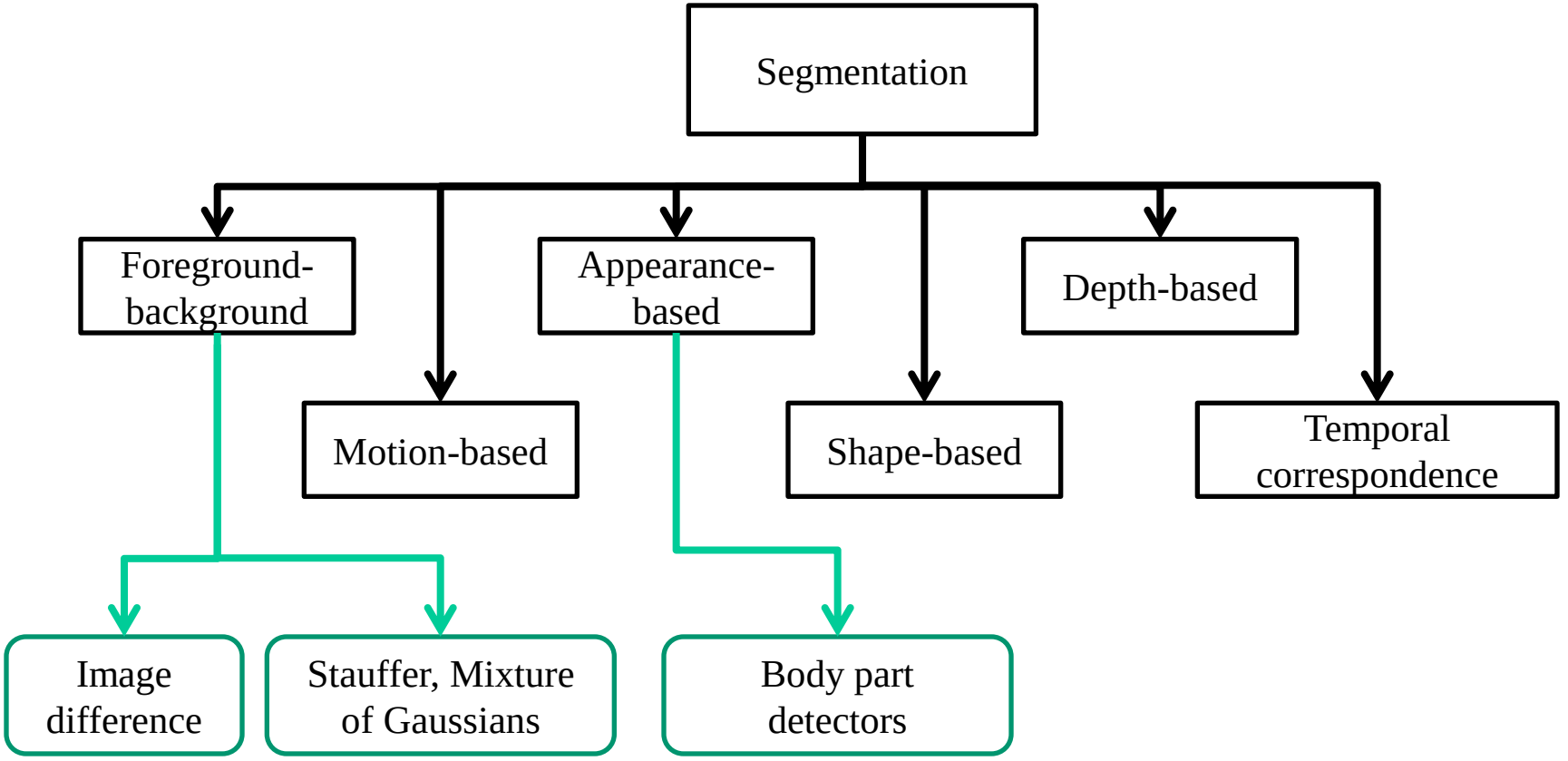


Taxonomy

Based on survey of Moeslund et al. (2006)

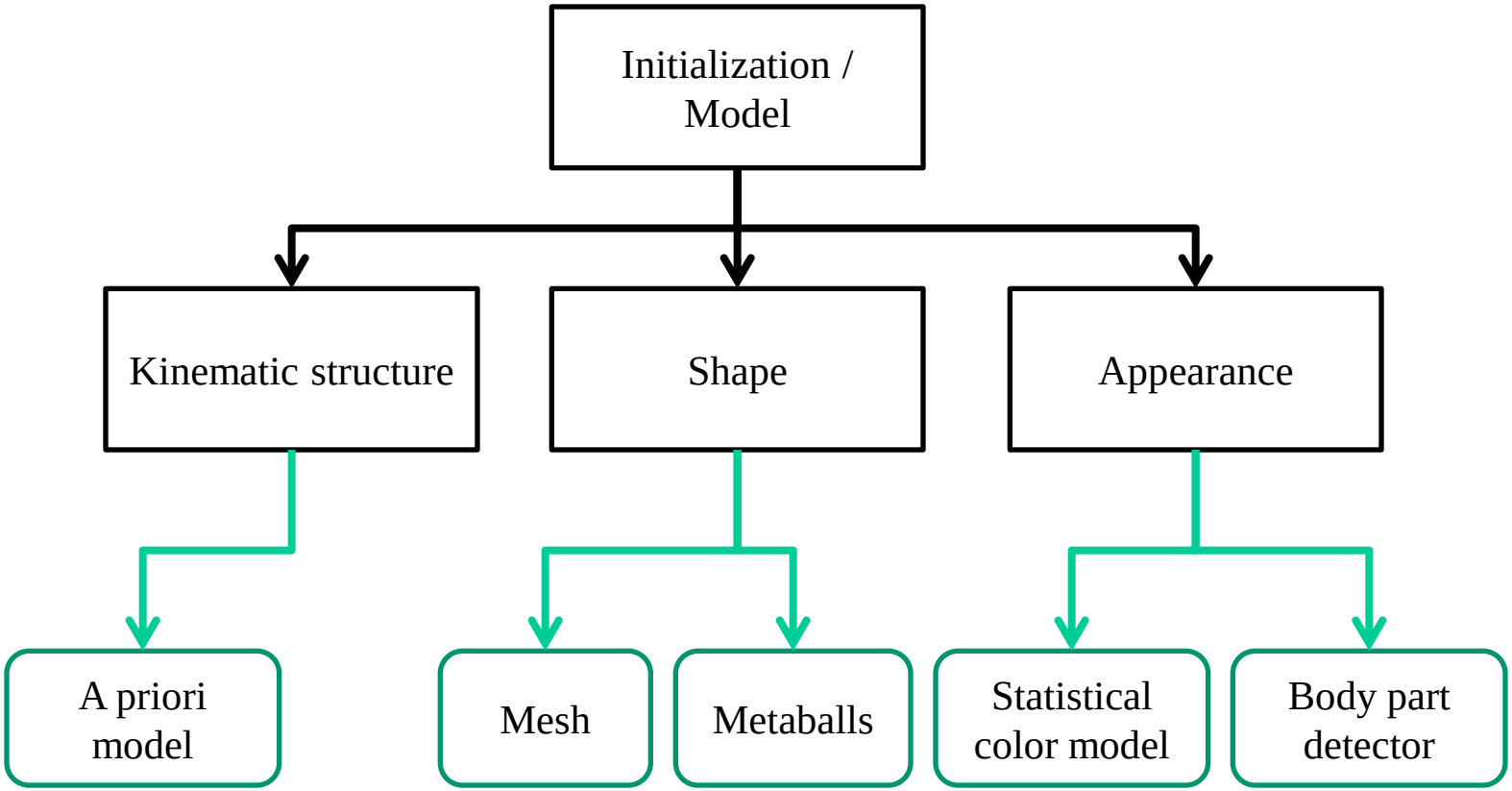


Taxonomy – Segmentation



(e.g. Viola & Jones, poselets, …)

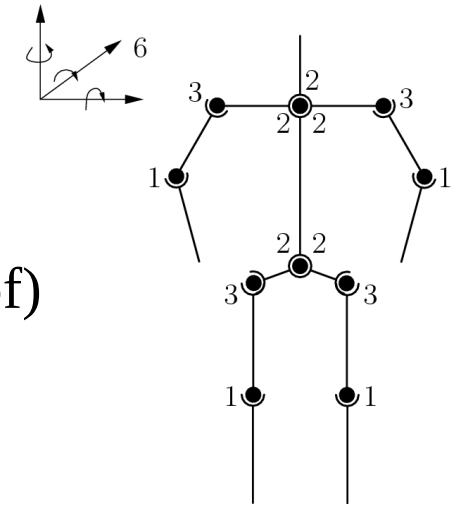
Initialization / Modellierung



Example: A priori model

- Model: rigid limbs connection with joints

- For example:
  - 13 limbs
  - 10 joints
    (varying dof, total 32 dof)



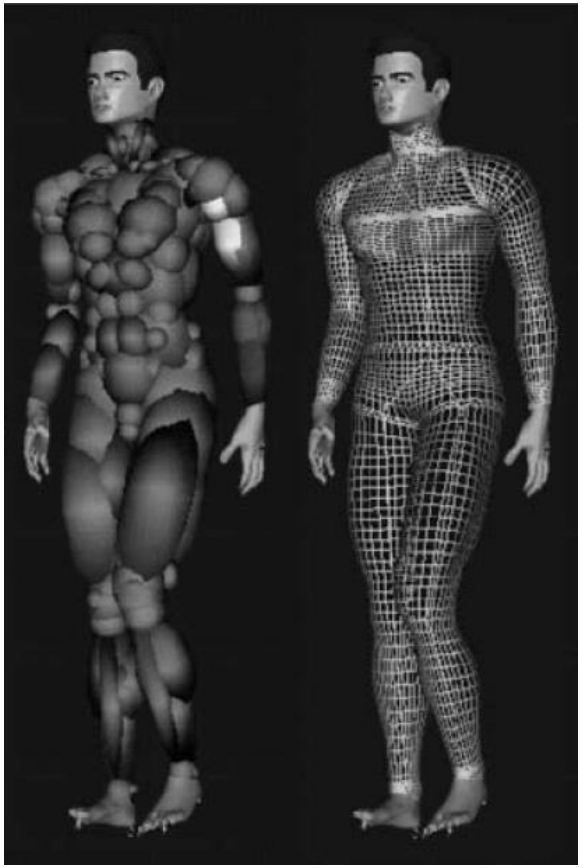
Σ 32 dof

$\vec{x} = (x, y, z, \alpha, \beta, \gamma, ...)$

- Challenge: huge search-space,
  around 180^32 (assuming 180 distinct positions for every joint)

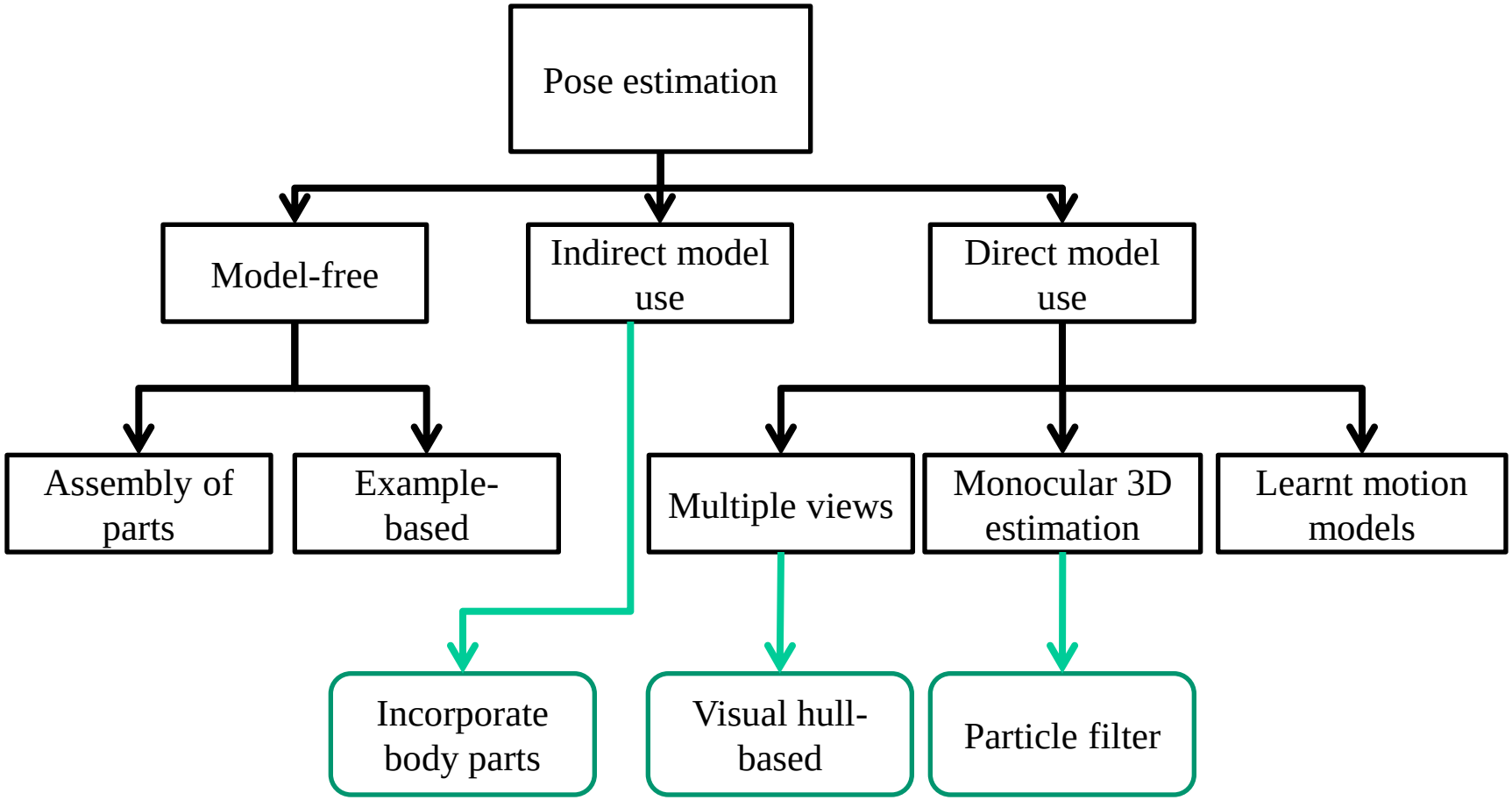
Example: Metaballs

- A metaball is described by a 3D Gaussian density function
- The metaballs are attached to the skeleton
- The surface is given by fitting the parameters of the Gaussians to the observation, e.g. silhouettes



- Plänkers, R., Fua, P.: "Articulated soft objects for multiview shape and motion capture", IEEE Transactions on Pattern Analysis and Machine Intelligence (2003)

Taxonomy – Pose estimation



2D image space

- Extract silhouettes of the people
- Try to match projection of 3D body model to 2D silhouette

- Advantages:
  - Can be applied to most camera setups and video data

- But:
  - Sensitive to occlusions

A Particle Filter for Articulated Body Tracking [Lee02]

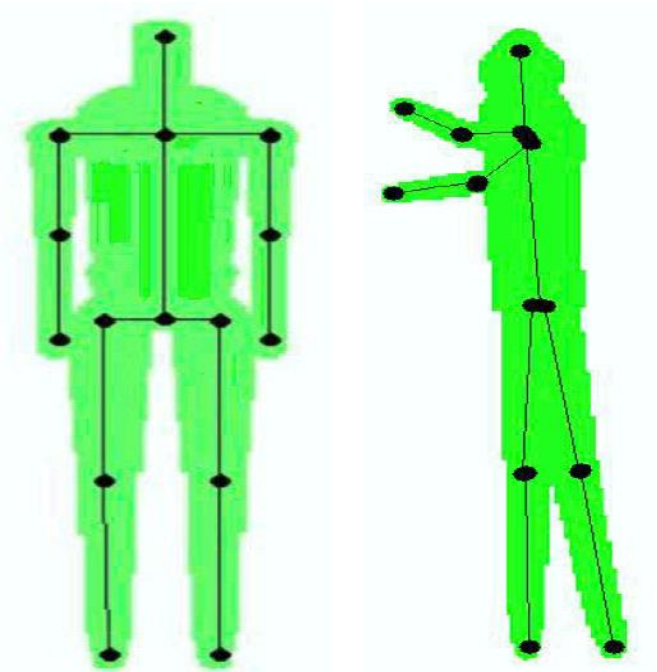
Target: Articulated body

- 14 segments (cones)
- 10 joints
- 32 dof

Feature: background subtraction

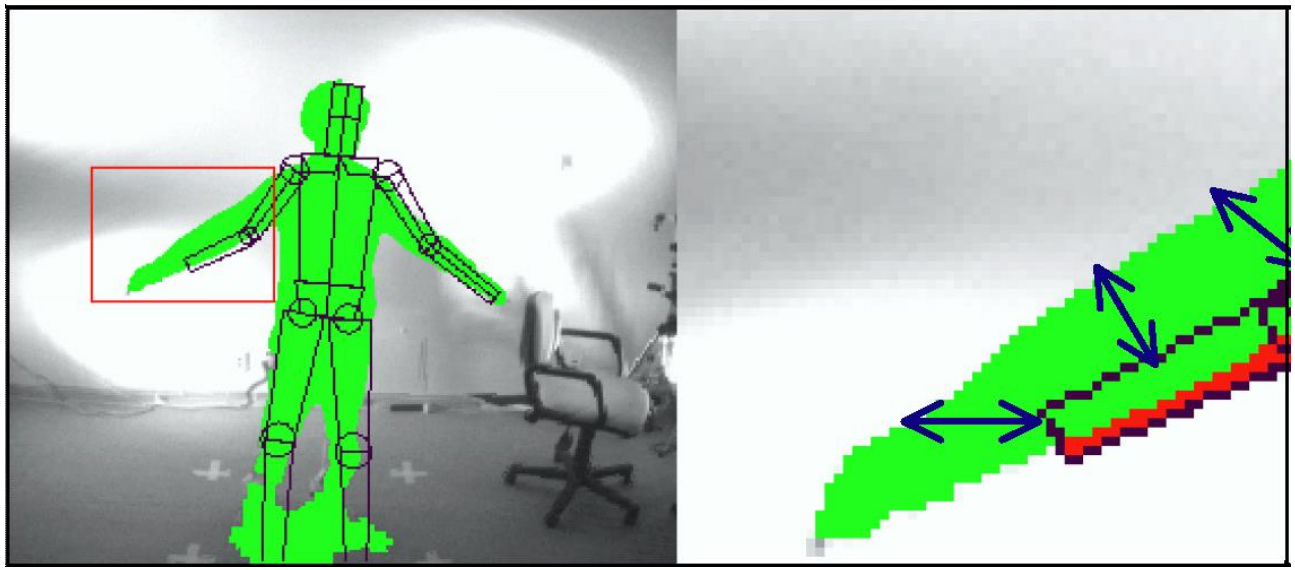
Sensors: 3 calibrated cameras

Tracking Scheme: Particle Filter



The Observation Model

Match each particle's state x_t with the extracted foreground silhouette y_t → $p(y_t|x_t)$

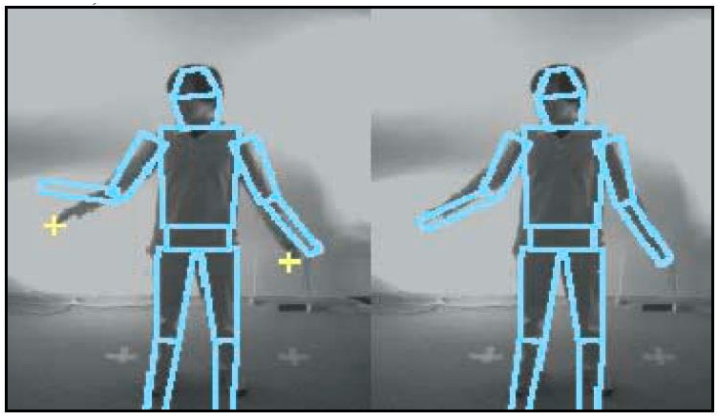


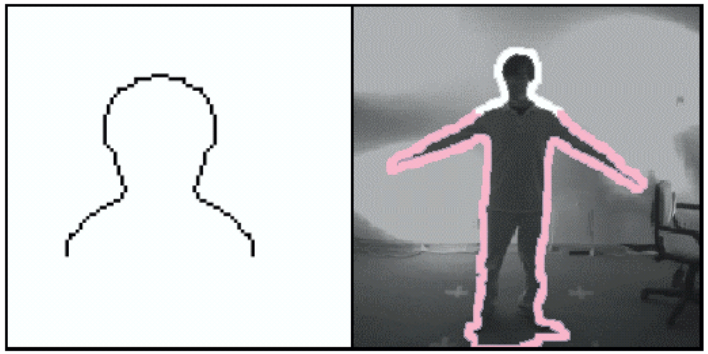
Detection of Body Parts

- Problem: the 32-dimensional state space is very high for a Particle Filter
- Solution: detect head, hands und torso using additional cues - and limit the PF state space using the detected positions
- Side-effect: automatic initialization

Hand/Head Detection

- **Hands**: Search peaks of curvature along the silhouette boundary
- **Head**: Search for a omega-shaped contour near the predicted head position
- Match the detections in multiple views → 3D positions
- Compare them with prior estimates of hand/head position





Torso Detection

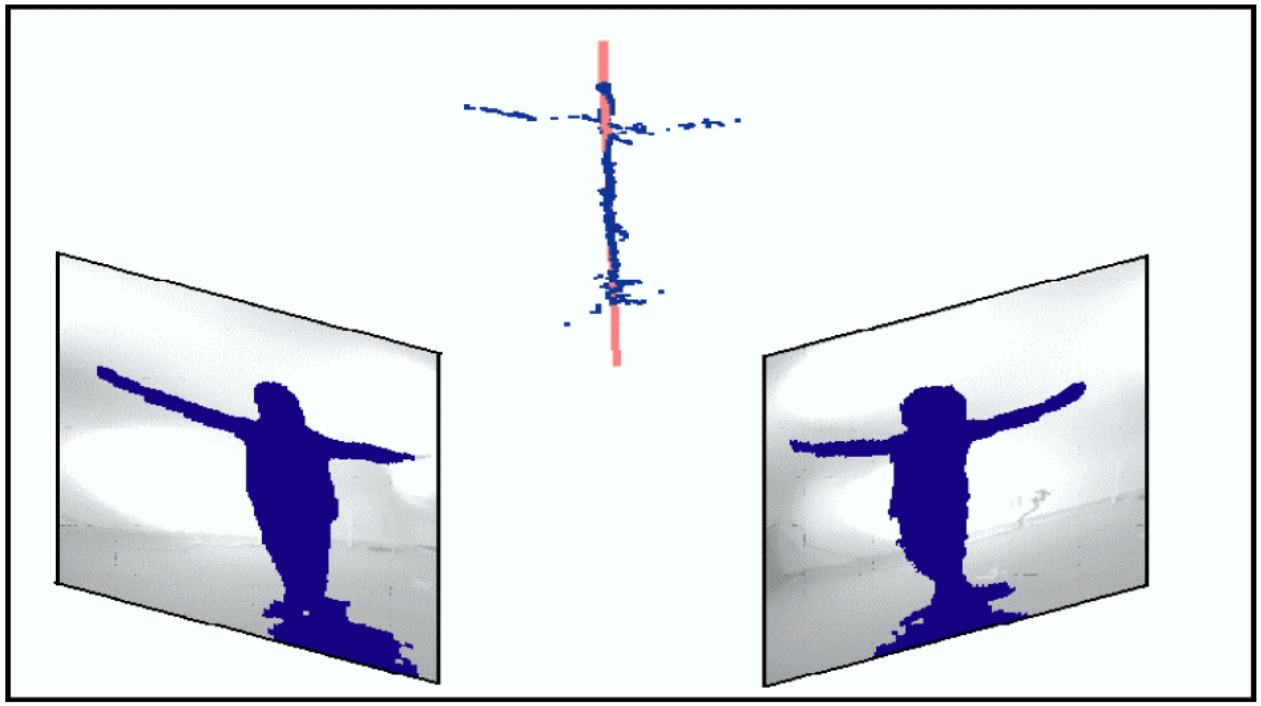
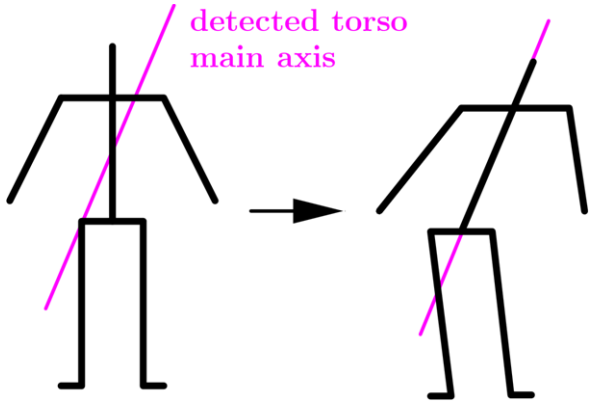


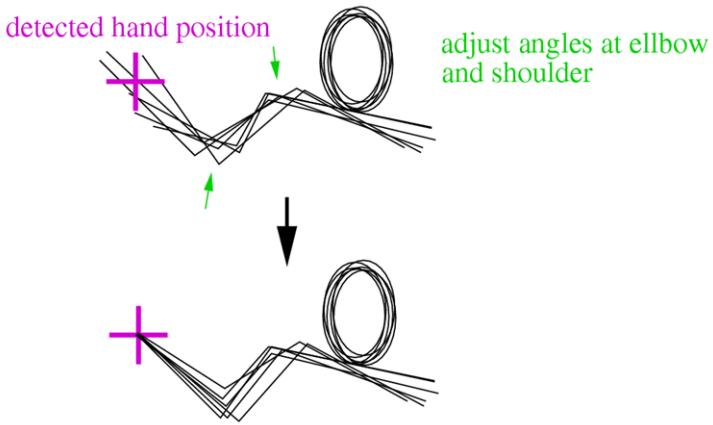
Figure 6: Torso detection. The medial axis of the silhouette in two views is extracted, matched and reconstructed in 3D. The torso axis is found by fitting a line to the medial axis points in 3D.

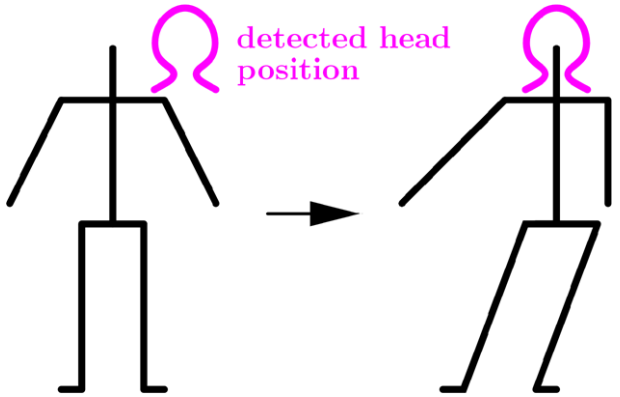
Incorporate Detections into PF

Force the particles to adapt to the detected body parts

→ reduces the complexity of the problem







3D space

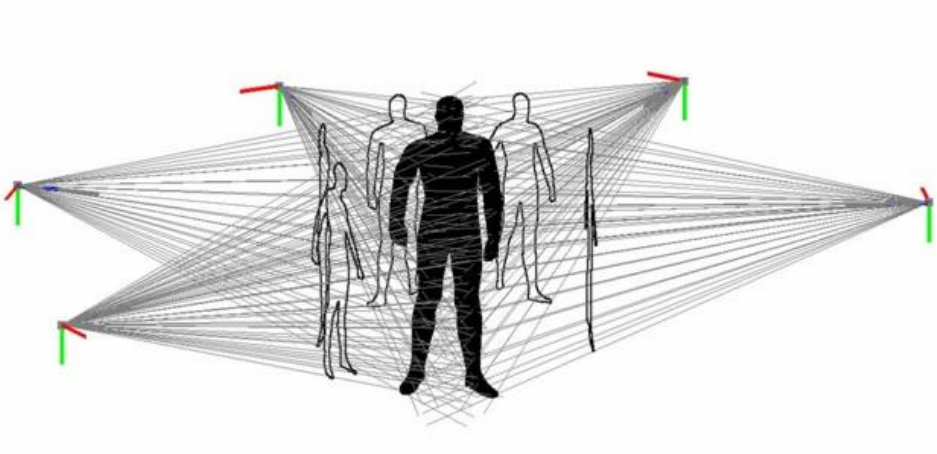
- Multiple calibrated cameras, usually 4 or more
- Transformation from multiple 2D images to one 3D representation

- Advantages:
  - Fusion of multiple camera images
  - Better occlusion reasoning due to more information

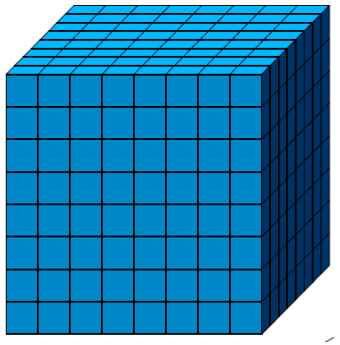
- But:
  - Computationally expensive preprocessing step
  - More hardware needed
  - Calibration of cameras necessary

Visual hull

- Definition: Maximal approximation of 3D shape S that can be substituted for S without affecting any silhouette [Laurentini94]
- Test: Project visual hull back into the image and compare to the silhouette
- Volume carving, constructive solid geometry, stereo vision, …



Volume carving I



- Approximate the visual hull using voxels
- Voxel: volumetric pixel (e.g. 3D cube)

- Simple indoor example:
  - Assume fixed voxel size, e.g. side length = 5cm
  - Define grid based on voxel size
  - Check for every voxel and for every camera view if the projection of the voxel into the image lies inside the silhouette
  - If this is the case, then the voxel belongs to the visual hull

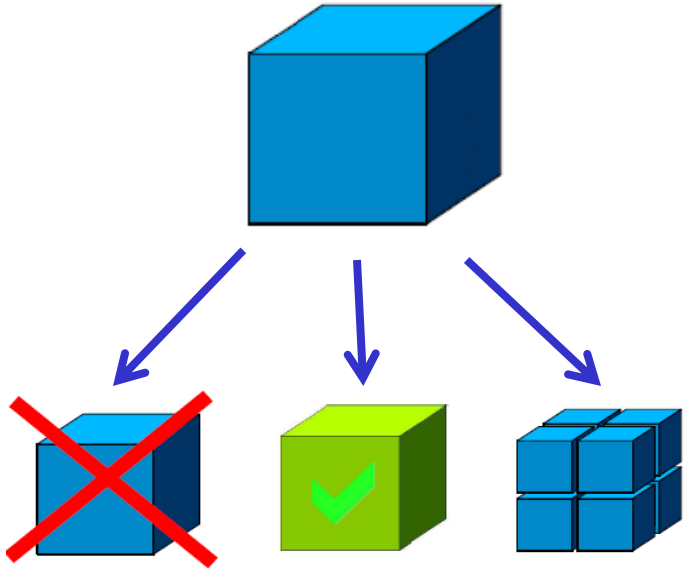
Volume carving II

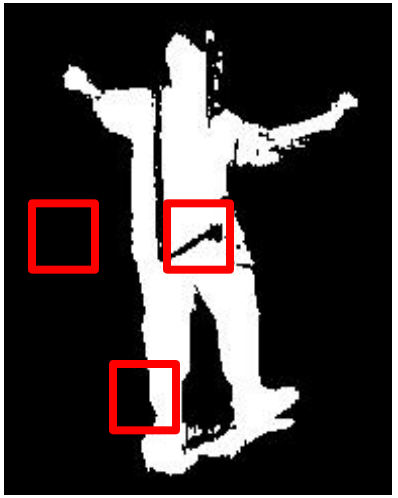
- Problem: Computationally expensive
  - Voxel side length: 5cm,
  - Room size: 4m x 4m x 3m
  - Camera views: 4
  - → Projections: 1.536.000 each frame
- Solutions:
  - Better hardware (GPU programming)
  - Lookup-tables
  - Octree-based voxels

Volume carving III

Octree-based voxel:

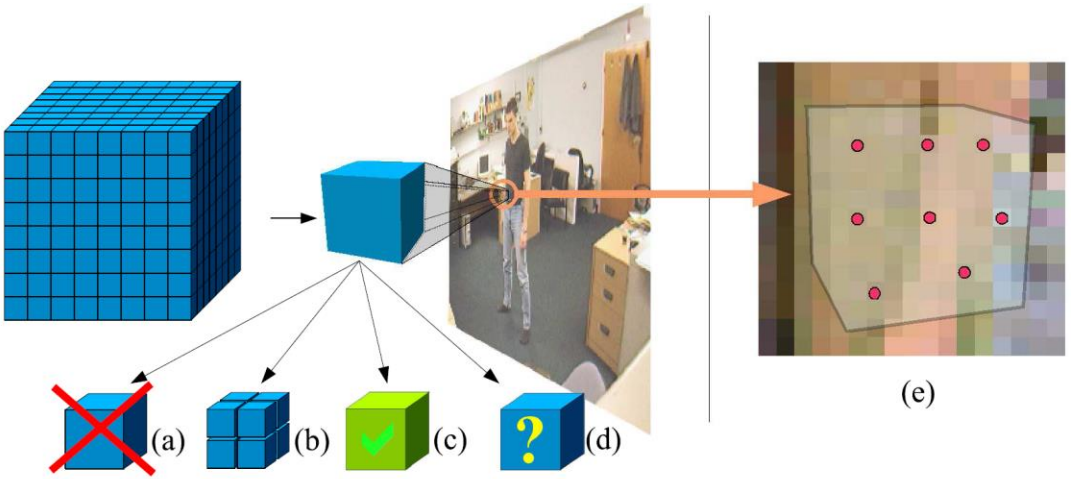
- Start with one big voxel
- Next step depends on projection:
  rate projection, e.g. fraction of silhouette pixels
  - Actions: delete, accept, split

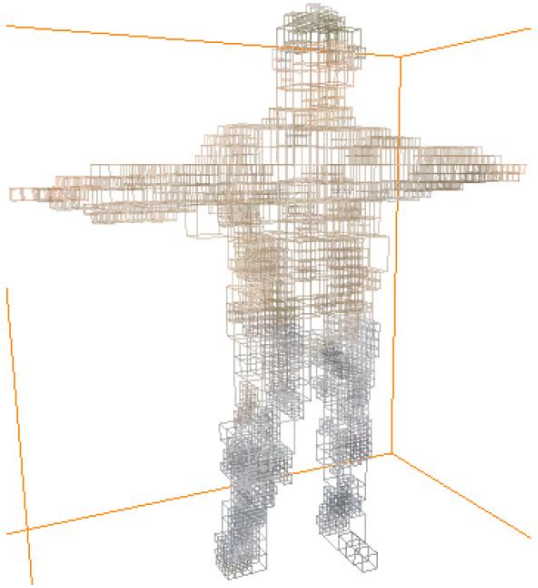




Real-Time Markerless Human Body Tracking with Multi-View 3-D Voxel Reconstruction

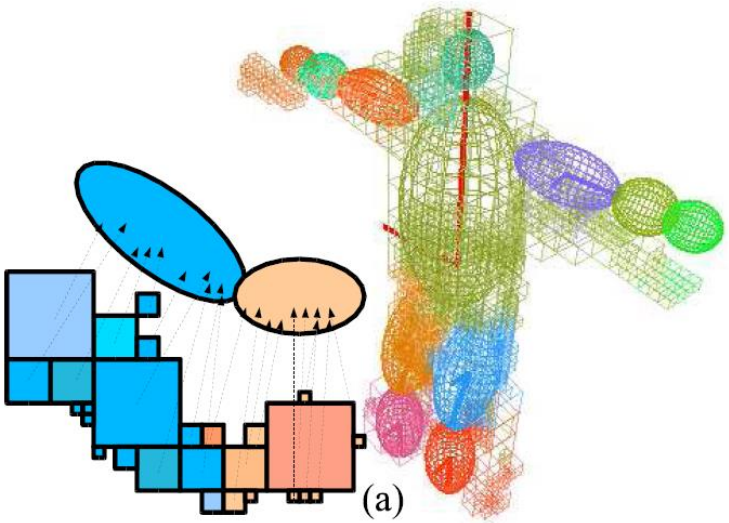
- Caillette and Howard (2004)
- Implicit color-based foreground segmentation
- Positive side effect: color information for voxels

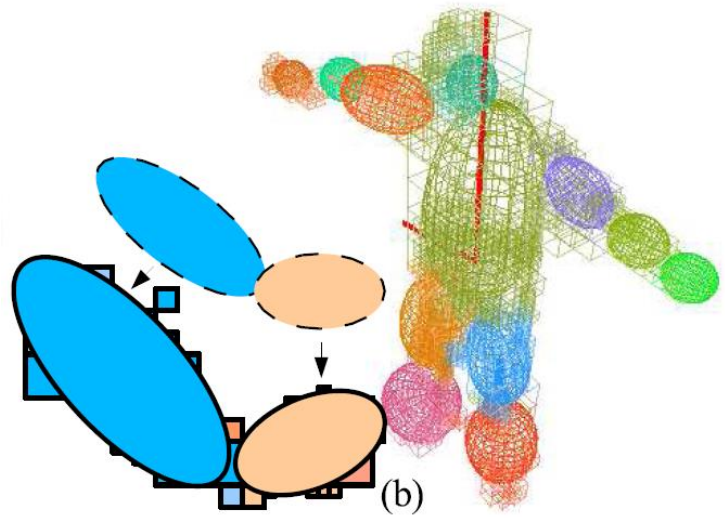




Articulated body tracking with voxels I

- Initialize colors of the body model (starfish position)
- Construct visual hull using colored voxels
- In every timestep:
  move limbs to voxels using Expectation Maximization

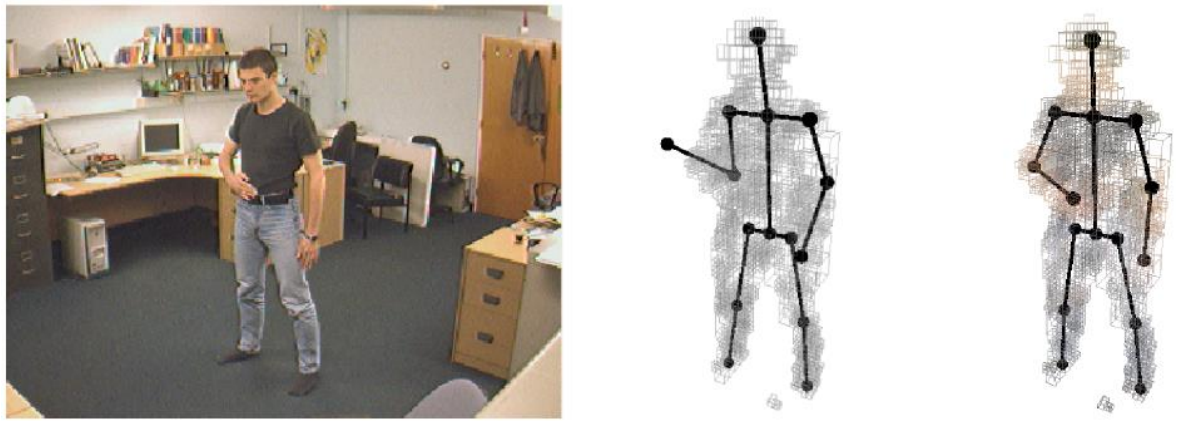




Articulated body tracking with voxels II

- Results:

- Colors help to reason about limbs close to the body



Summary

- Introduction:
  - Multi-camera topologies, 3D to 2D projection, Intrinsics vs. Extrinsics, Calibration
- Multi-object tracking:
  - Using multiple KF-based trackers and triangulation
  - Tracking without triangulation, using particle-filters
    - Smart Room – multiple cameras / microphones
    - Robot – stereo camera
- Articulated body tracking:
  - Taxonomy from Moeslund et al.
  - 2 Examples: Silhouette matching (2D), Voxel-based (3D)

References

[Lee 2002]
Mun Wai Lee, Isaac Cohen and Soon Ki Jung. *Particle Filter with Analytical Inference for Human Body Tracking.* Institute for Robotics and Intelligent Systems, Integrated Media Systems Center, University of South California, 2002.

[Focken 2002]
D. Focken, R. Stiefelhagen. *Towards Vision-based 3-D People Tracking in a Smart Room. IEEE* International Conference on Multimodal Interfaces, Pittsburgh, PA, USA, October 14-16, 2002, pp. 400-405.

[Caillette 2004]
Fabrice Caillette and Toby Howard. *Real-Time Markerless Human Body Tracking with Multi-View 3-D Voxel Reconstruction*. In Proc. ISMAR, 2004, pp. 597—606.

[Laurentini 1994]
A. Laurentini. *The visual hull concept for silhouette-based image understanding*. IEEE Transactions on Pattern Analysis and Machine Intelligence, 1994 (16), pp. 150-162.

[Moeslund et al. 2006]
Thomas B. Moeslund, Adrian Hilton, Volker Krüger. *A survey of advances in vision-based human motion capture and analysis,* Computer Vision and Image Understanding, 2006 (104), pp. 90-126.