Dual Stochastic and Silhouette-Based 2D-3D Motion Capture for Real-Time Applications

Pedro Correa Hernández

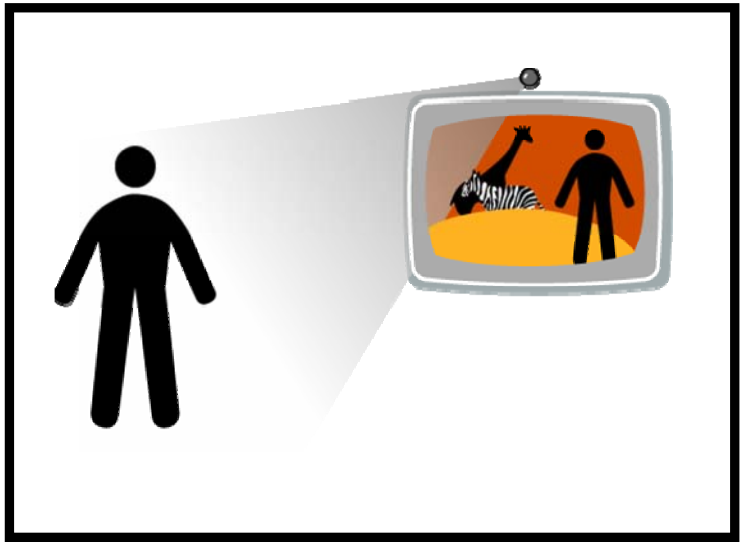
Benoit Macq (UCL), Xavier Marichal (Alterface), Ferran Marqués (UPC)

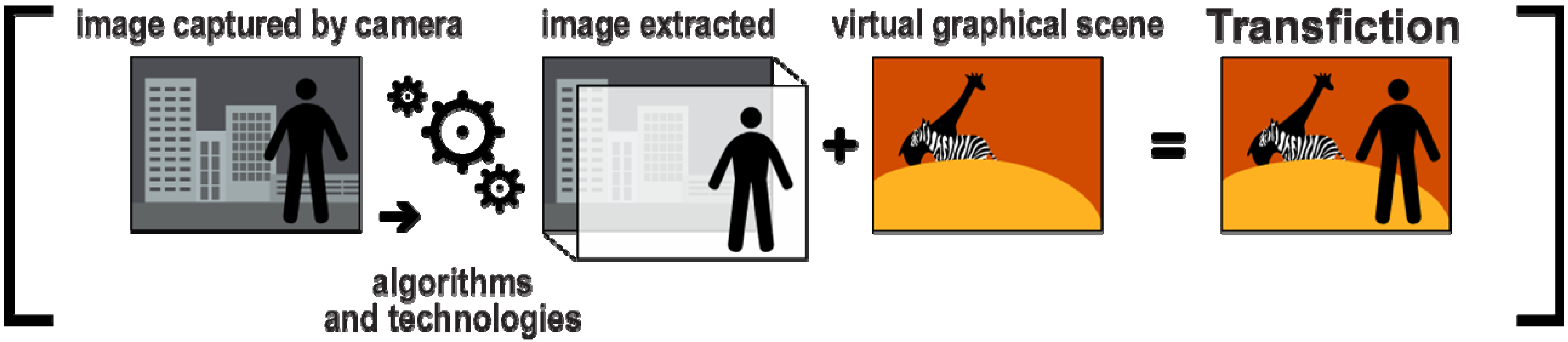
Presentation Overview

- The Augmented Reality Concept
- Our goal
- The Intra-Image Phase
  - Results
- The Inter-Image Phase
  - Results
- Conclusions and Future Work

The Augmented Reality concept

- Augmenting the real world scene but still maintaining a sense of presence of the user in that world.

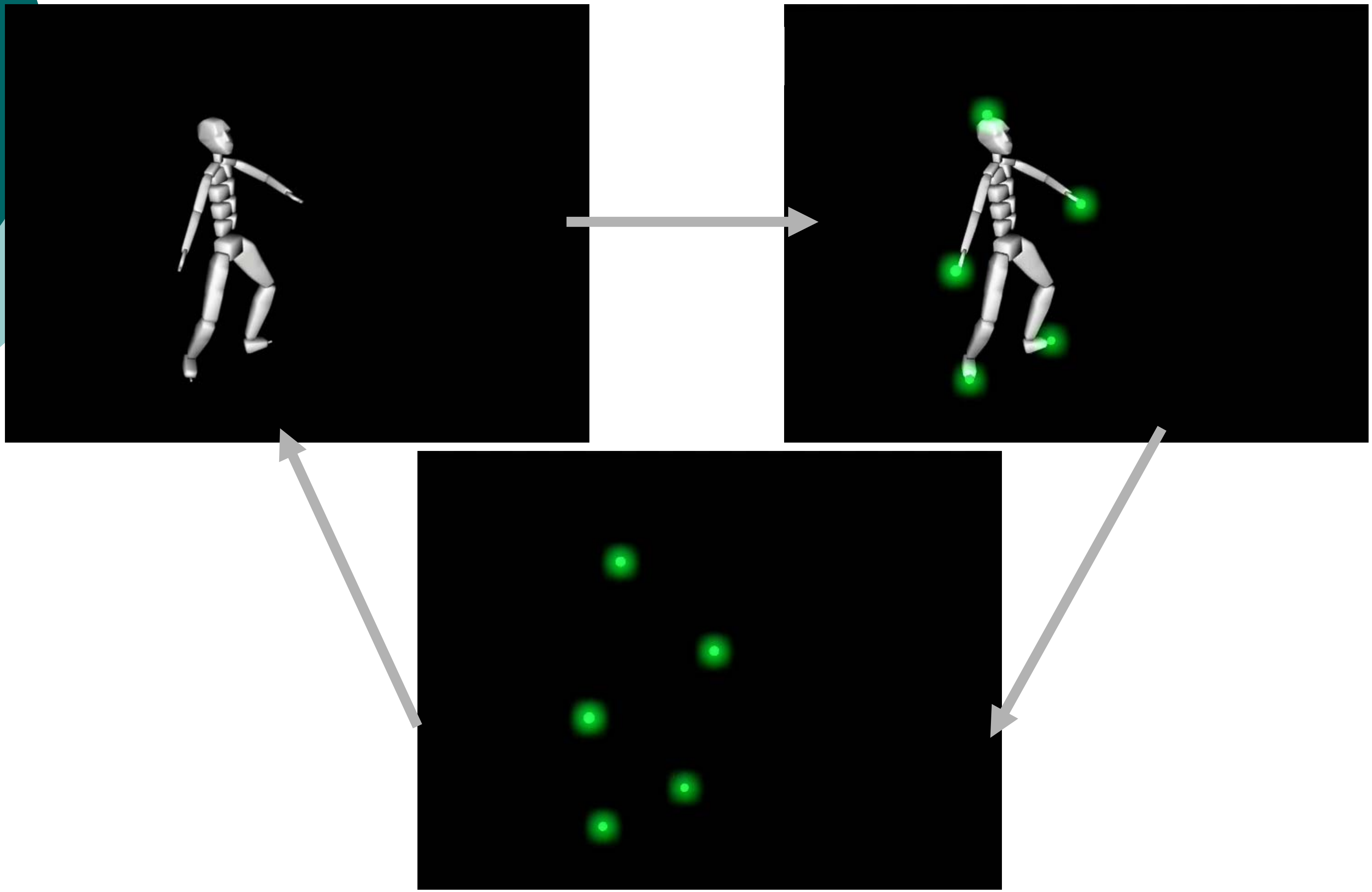




Presentation Overview

- The augmented reality concept
- Our goal
- The Intra-Image Phase
  - Results
- The Inter-Image Phase
  - Results
- Conclusions and Future Work

Our goal

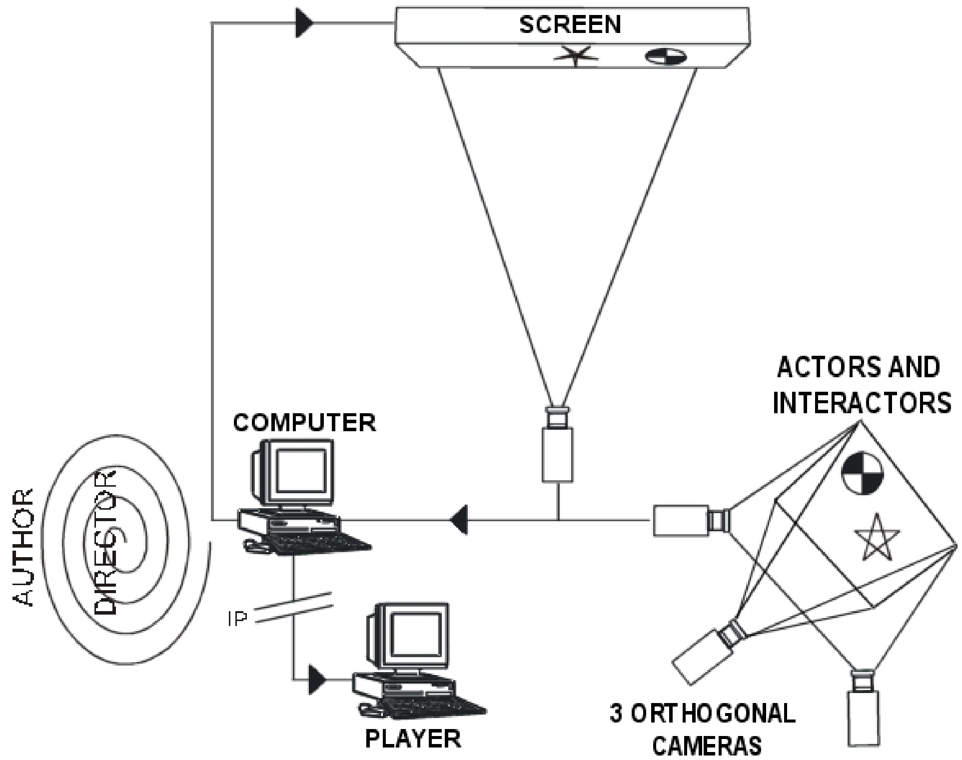


Presentation Overview

- The augmented reality concept
- Our goal
- The Intra-Image Phase
  - Results
- The Inter-Image Phase
  - Results
- Conclusions and Future Work

Infrastructure

- 2, non calibrated, relatively orthogonal cameras
- A controlled scenario



Overview of the algorithm

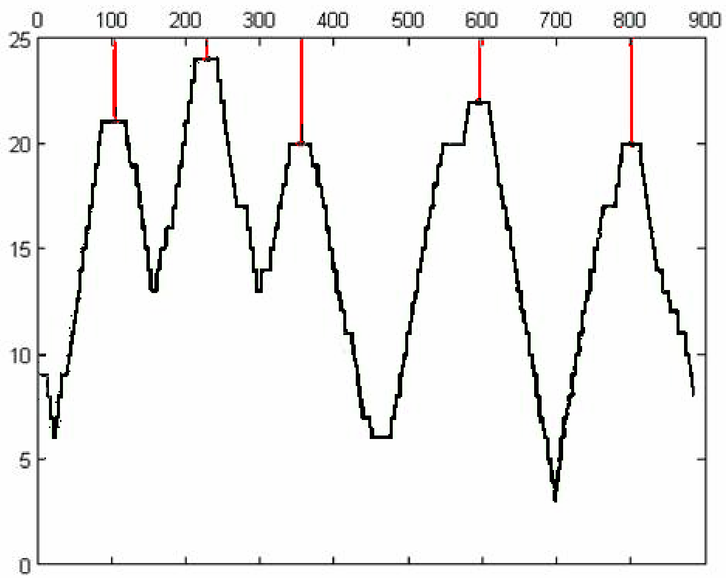
- Based on silhouette analysis
- No a priori average human limb lengths knowledge

- Main steps
  - Extraction of the crucial points
  - Labeling (Crucial point A=Head)
  - 3D Fusion

Crucial Points Extraction

- Crucial Points : human features that overall define a specific posture
- These are (in our application): the head, hands and feet
- They are the farthest points of the silhouette with respect to a certain point: the Center of Gravity (*COG*)
- Morphological information to extract them:
  - They are located on the silhouette's border
  - They represent 5 local geodesic distance maxima with respect to the **C**enter **o**f **G**ravity

Crucial Points Extraction (Frontal View)

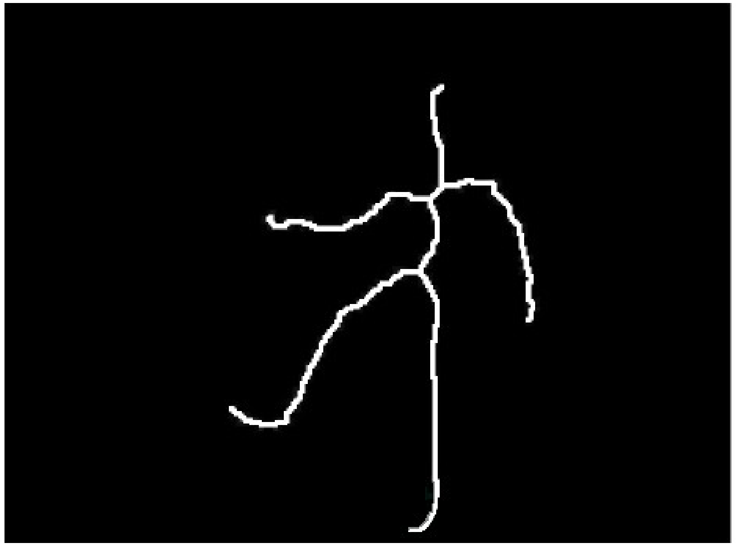


- Scene **capture** (three orthogonal views)
- Actor **segmentation**
- **CoG** computation
- Creation of the **geodesic distance map**
- **Contour tracking**
- Creation of the distance/silhouette border position **function**
- One-dimensional dilation of the function
- **Local maxima** extraction

Crucial Point extraction, Real-Time

Creation of the skeletons

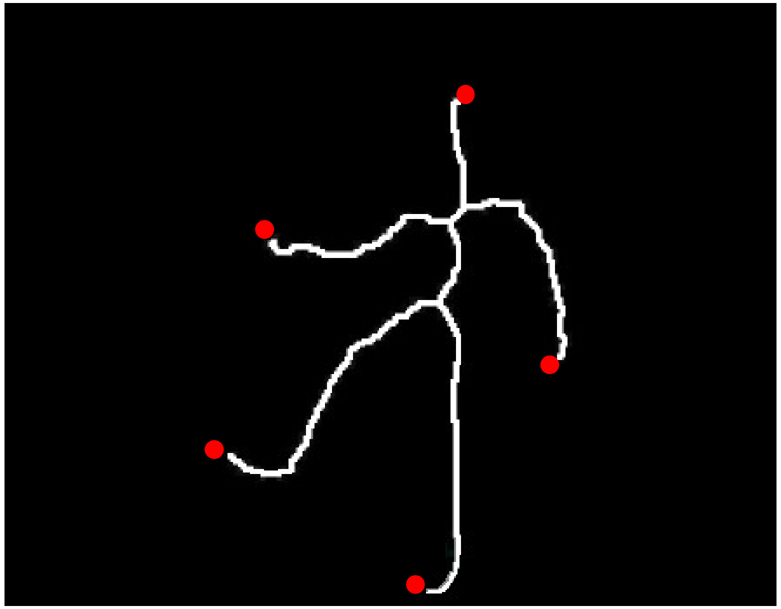
Morphological skeleton

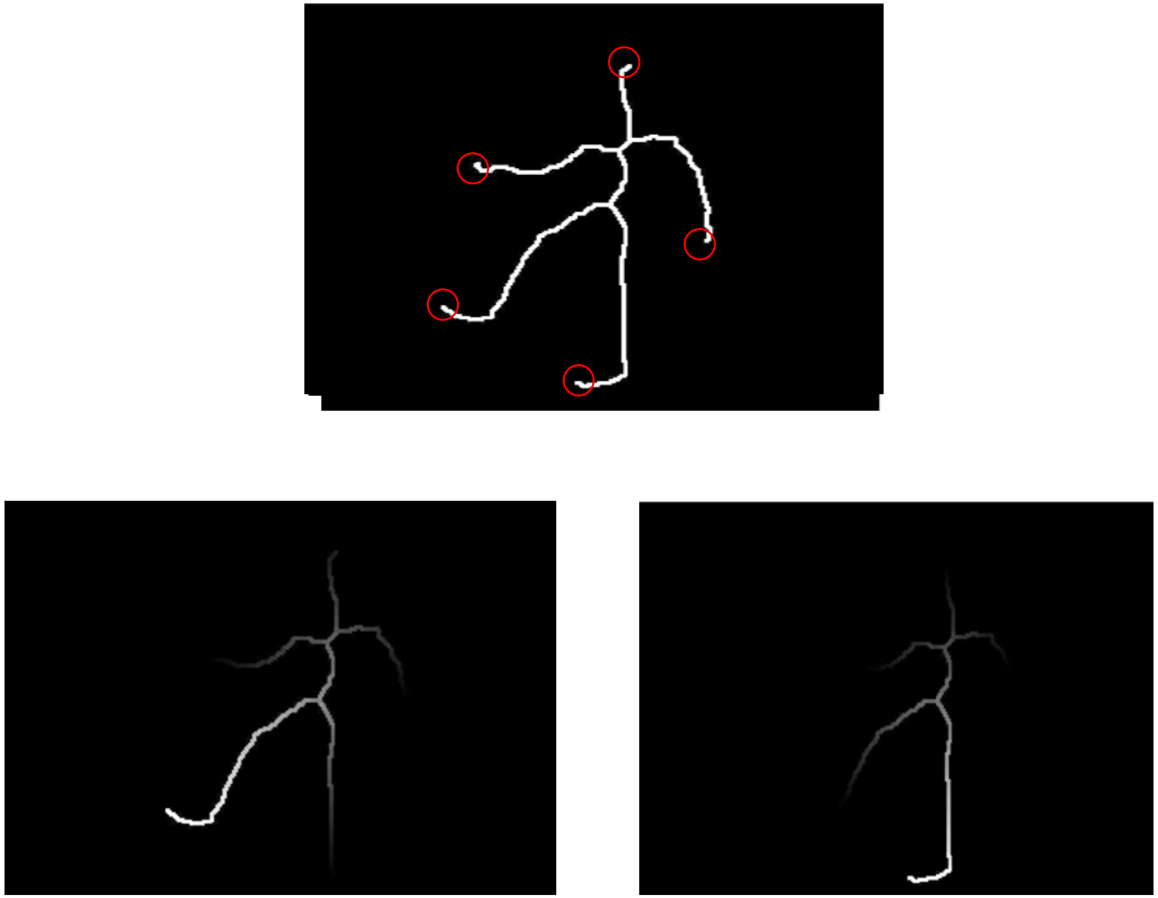


Noise-free skeleton

Labeling of Crucial Points

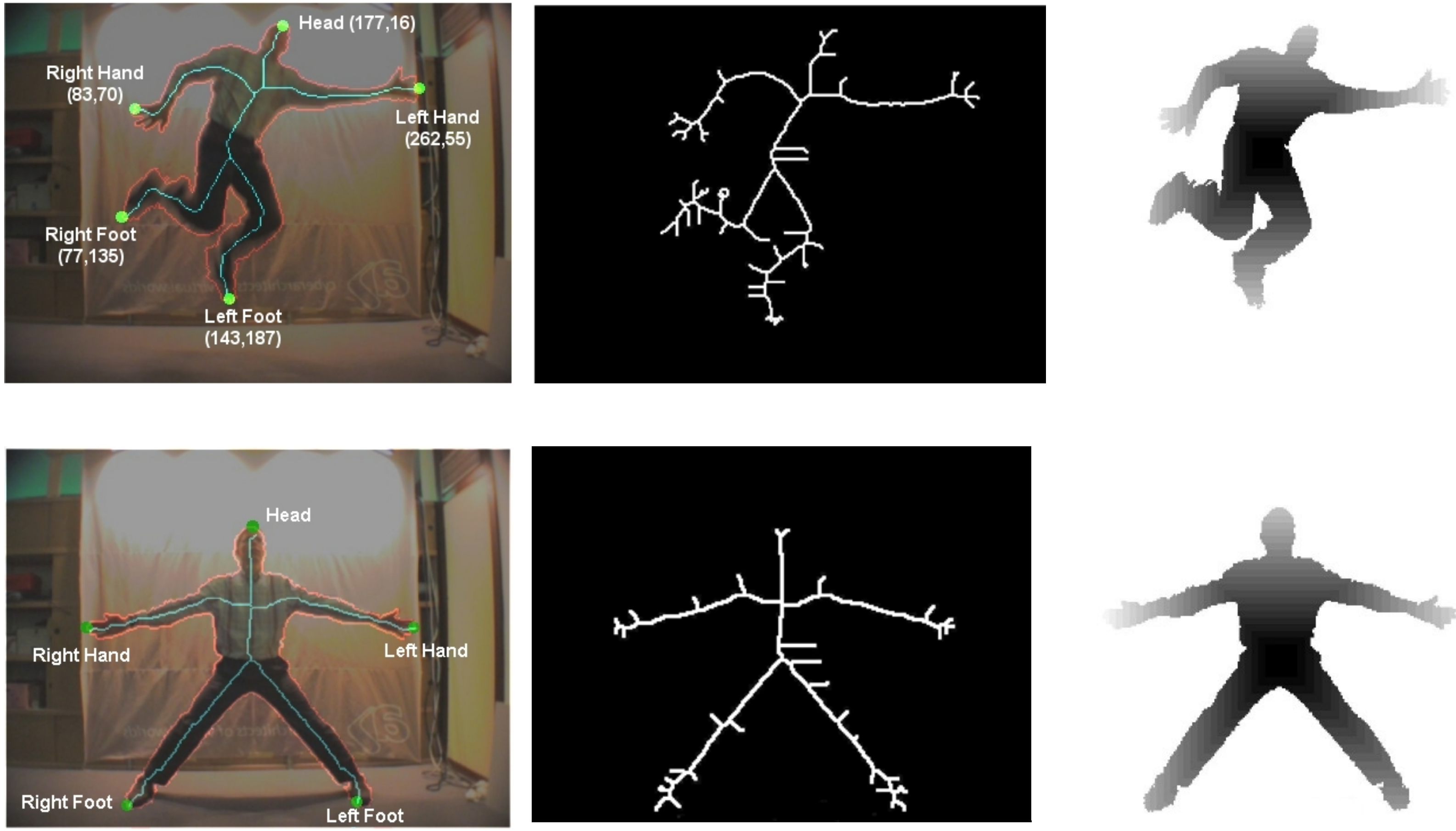
- Goal: match each crucial point with the human feature they correspond
- How: Using noise-free morphological skeletons



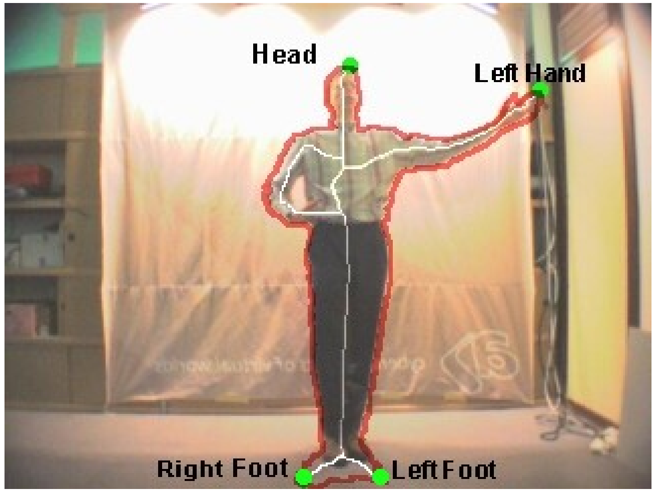


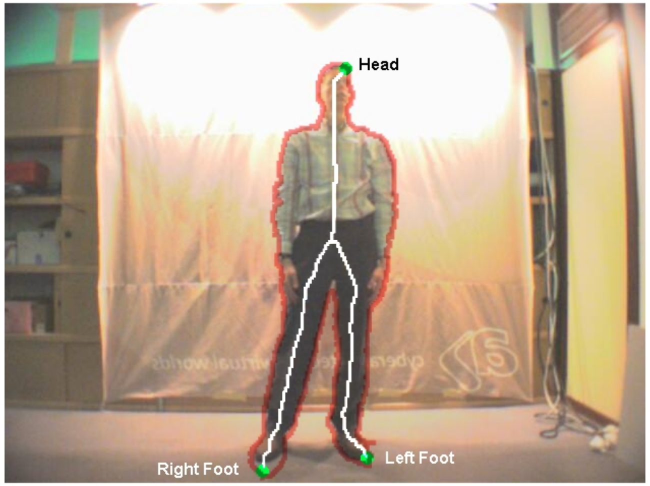
Final result

More results:



Results dealing with self-occlusions

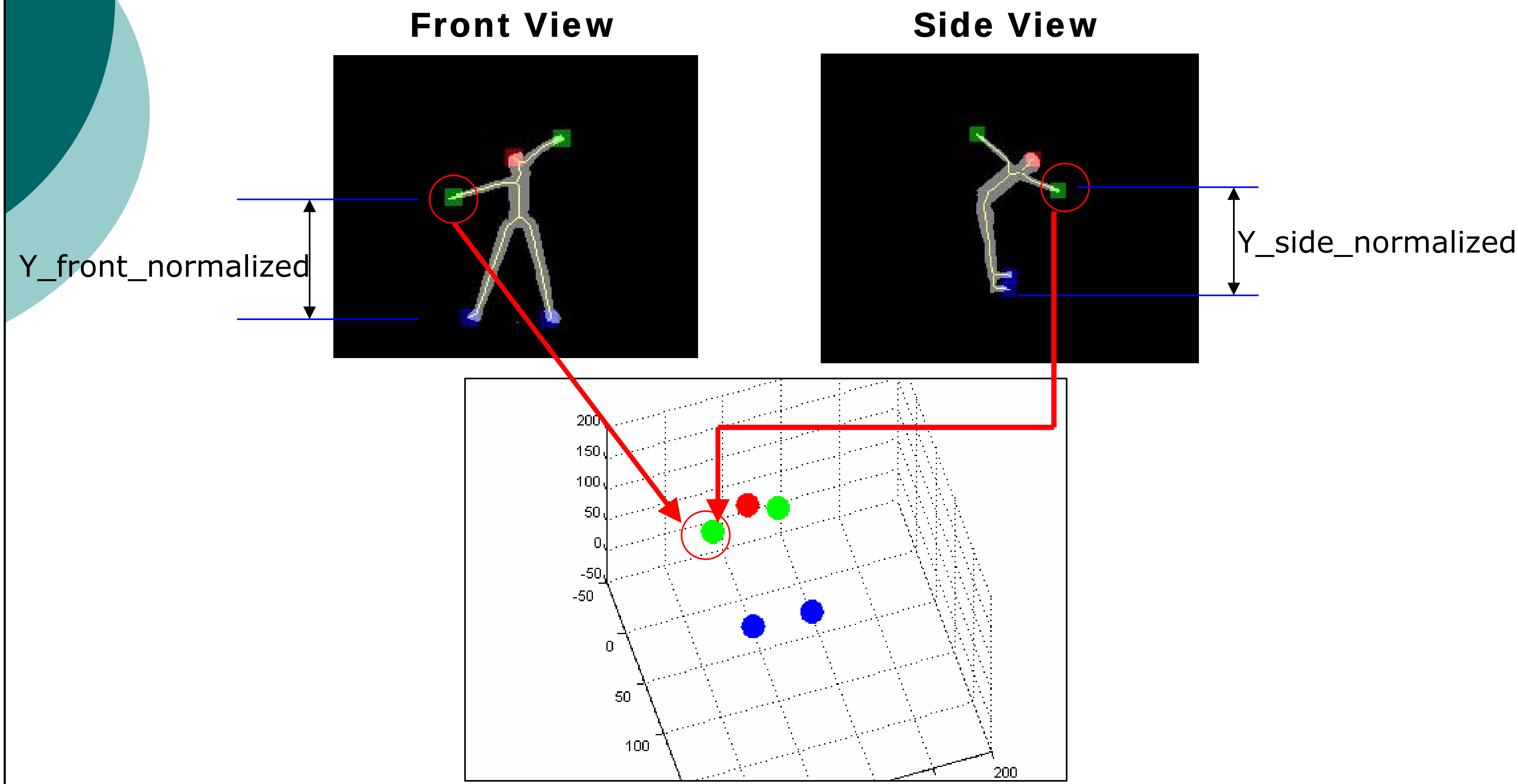




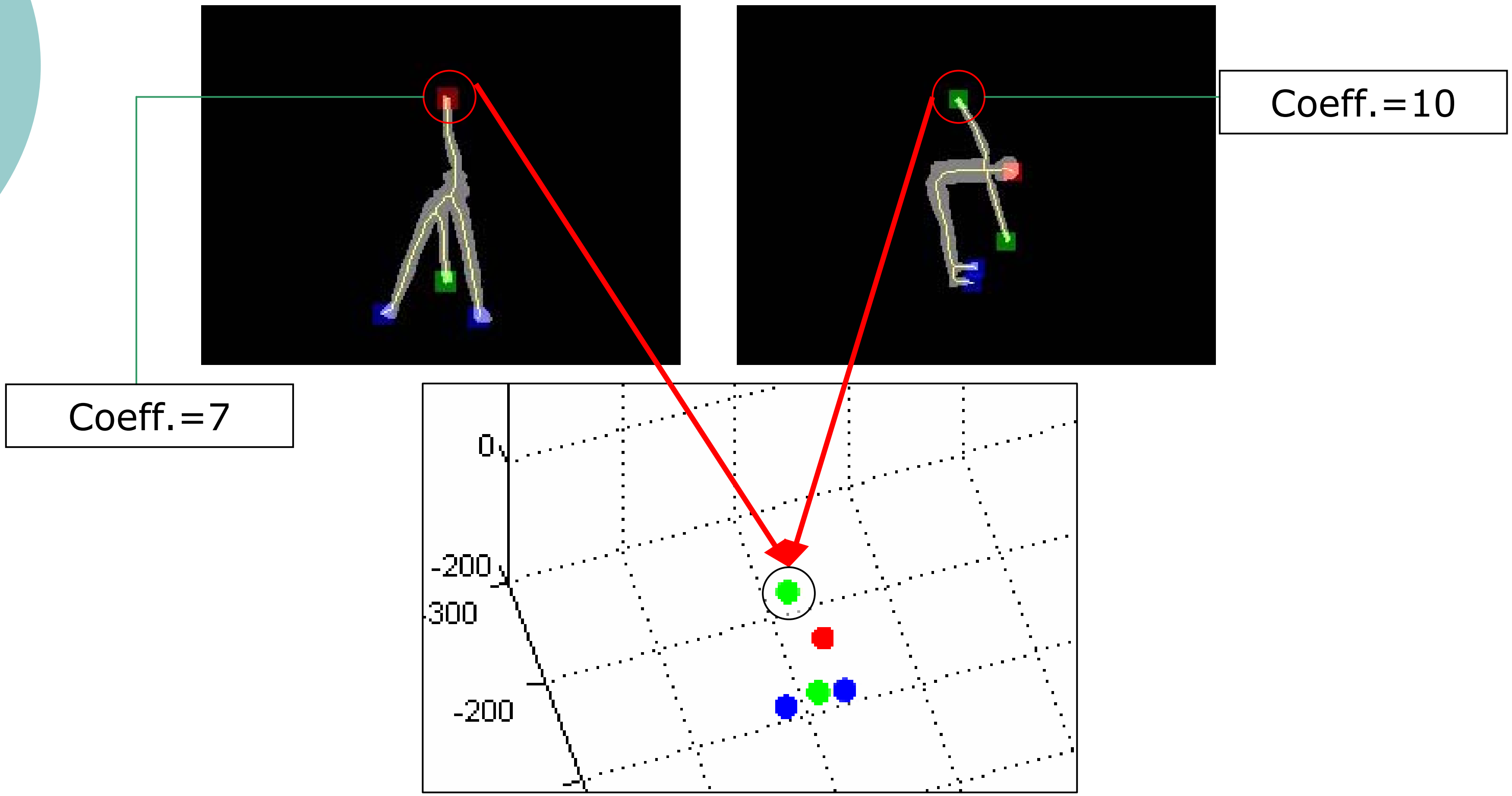
3D reconstruction

- Goal: Match previously labeled points of the two orthogonal views
- Benefits:
  - Verification of labeling
  - Use of non-occluded points of each view
  - Retrieval of 3D information

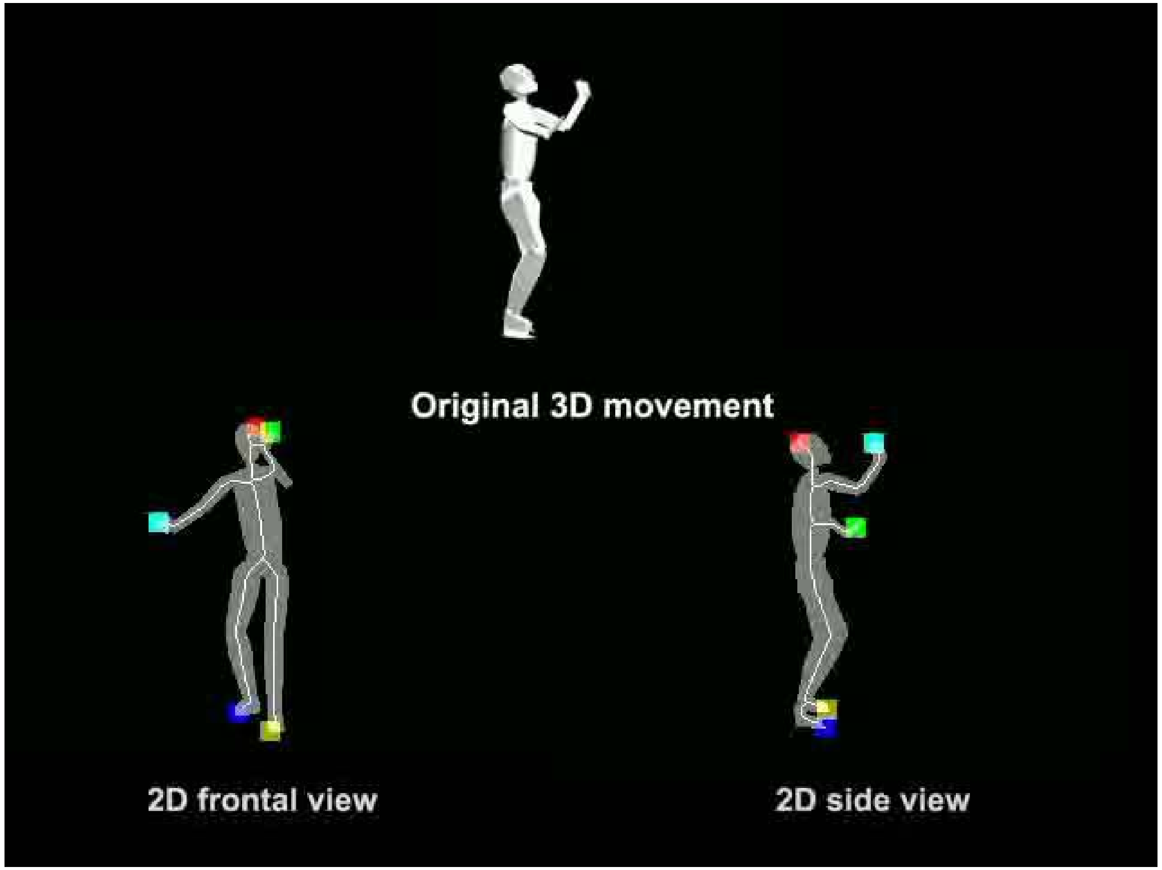
3D reconstruction



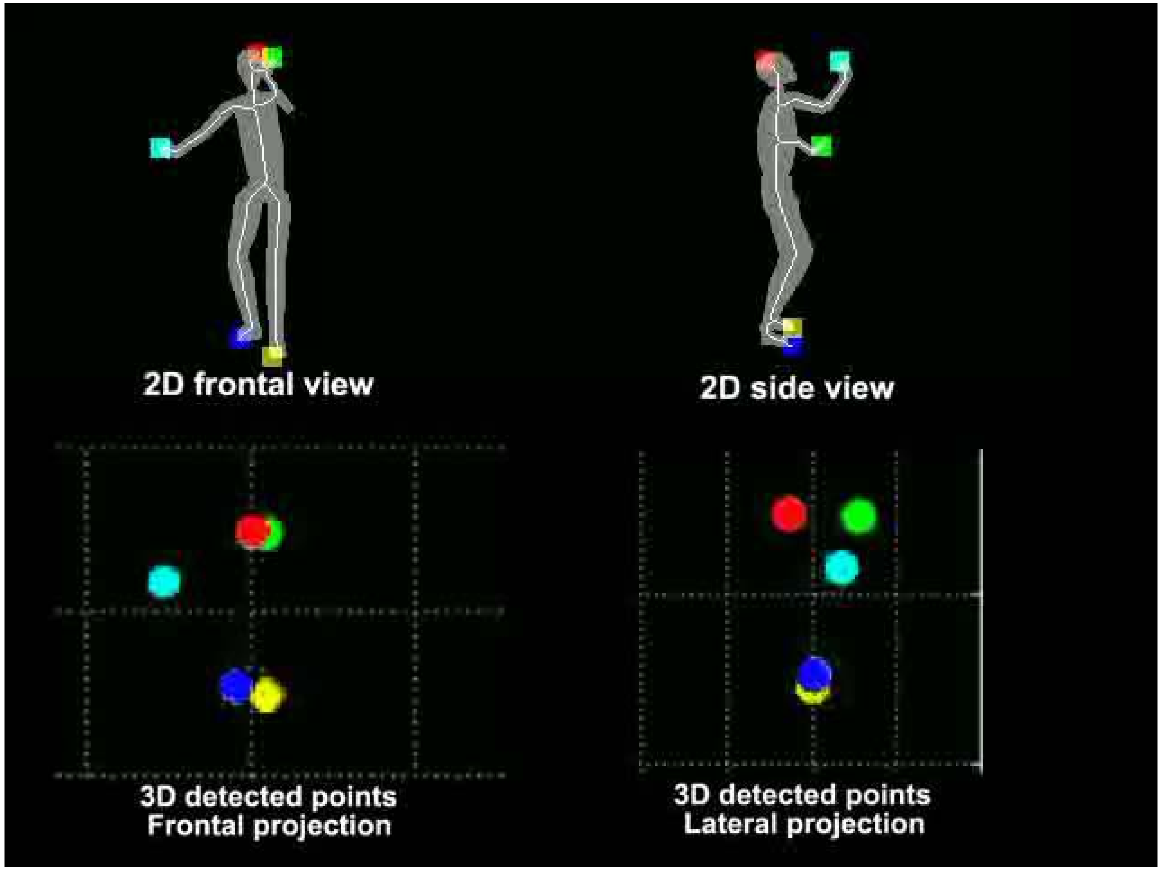
3D reconstruction: reliability coef.



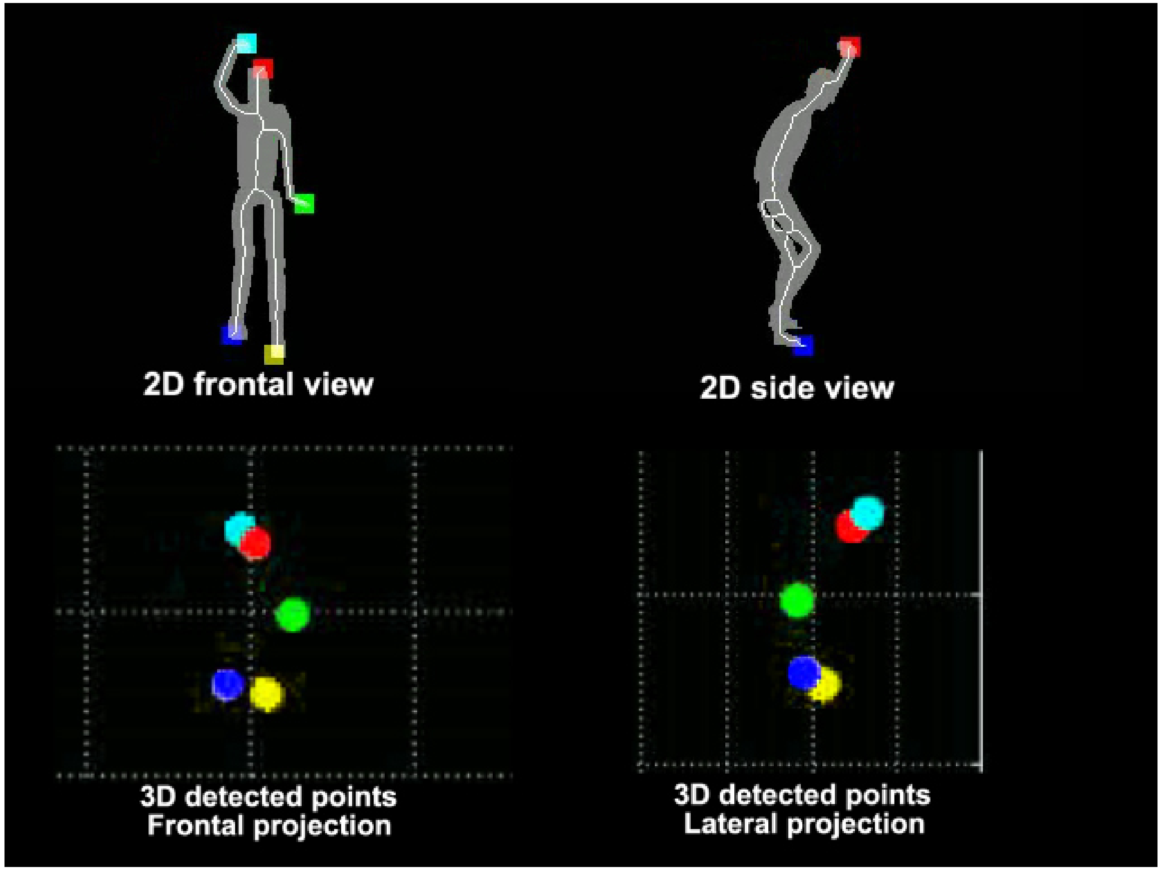
Intra frame detection: 2D Views



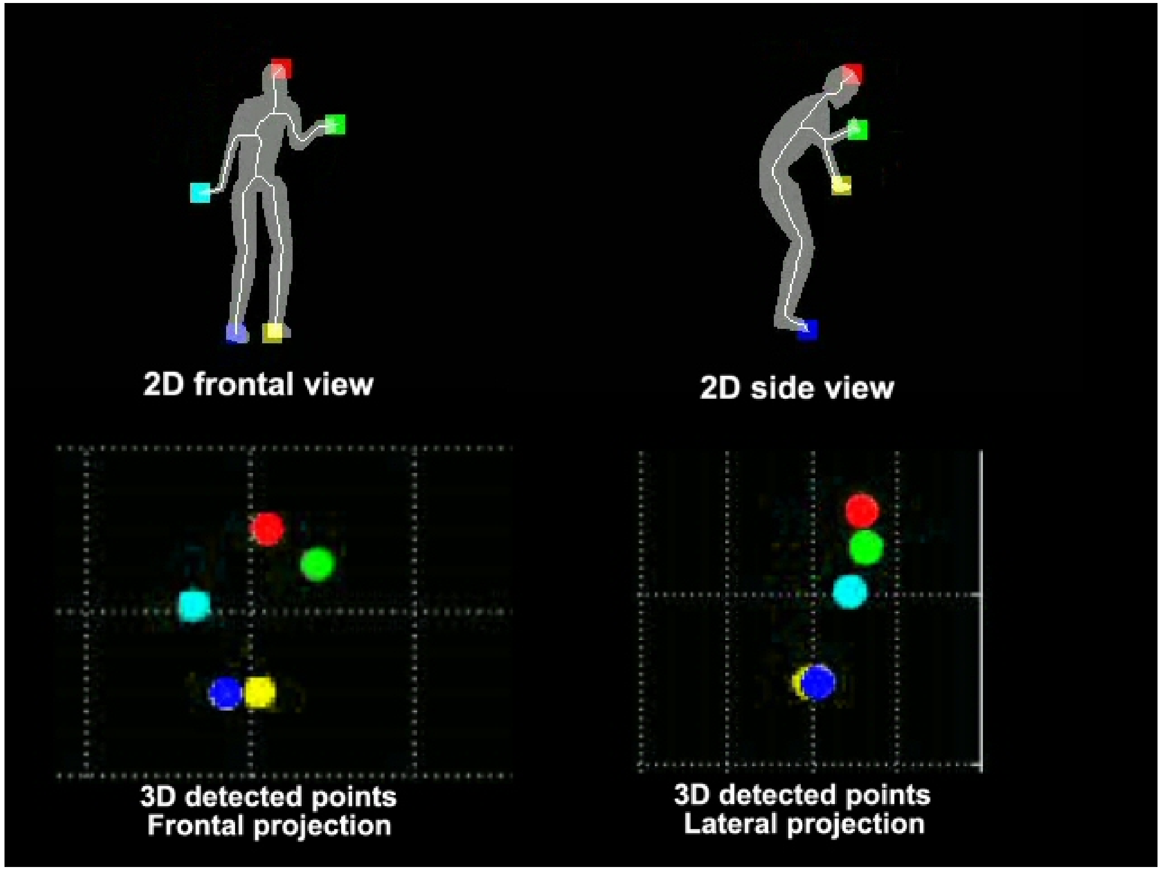
Intra frame detection : 2D → 3D



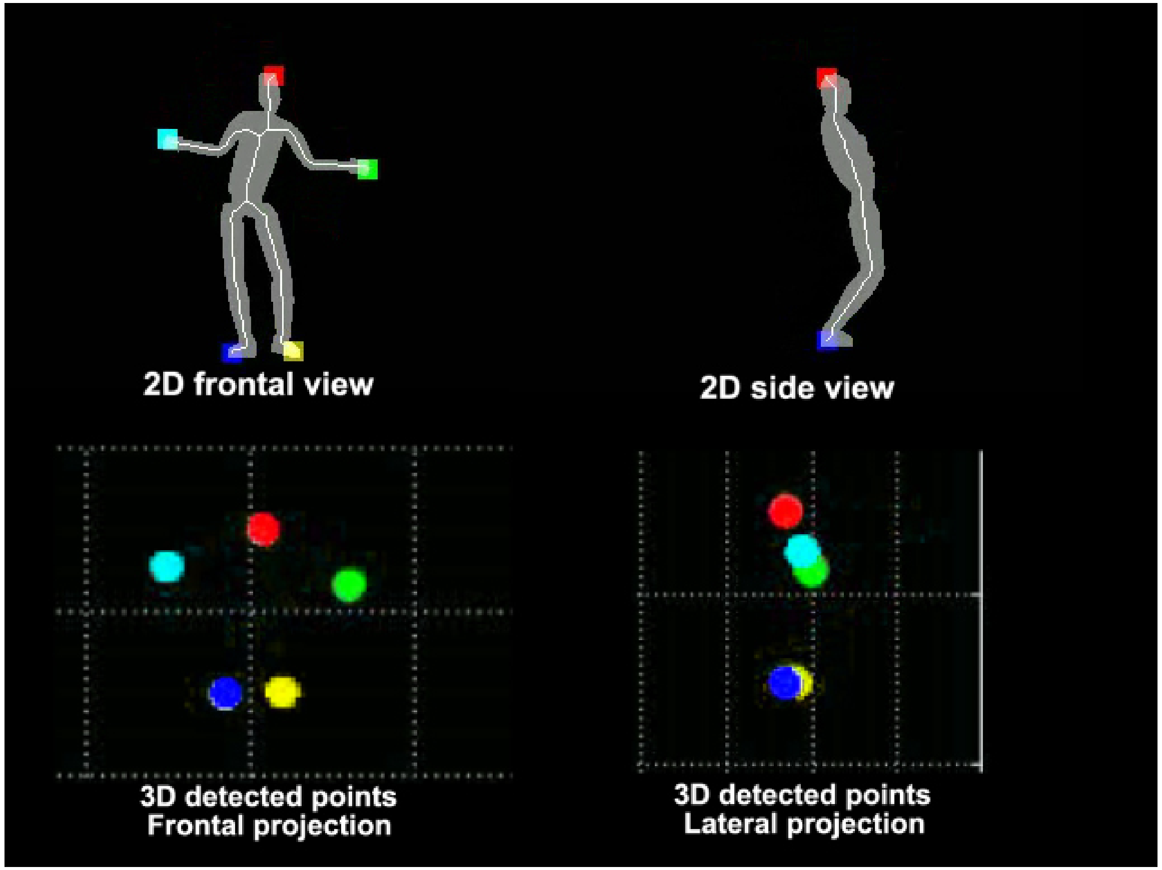
Snapshot: Reliability Coefficient. Example 1



Snapshot: Reliability Coefficient. Example 2



Snapshot: Cases of occlusion. Example



Presentation Overview

- The Augmented Reality Concept
- Our goal
- The Intra-Image Phase
  - Results
- The Inter-Image Phase
  - Results
- Conclusions and Future Work

Stochastic Analysis

- Once the crucial points are labelled we need to track them in order to
  - Prevent point flickering (self occlusions)
  - Avoid label inversions
  - Correct labelling errors
- Major problem: Standard Kalman is not apropriate in this context:
  - Points have very irregular trajectories
  - They are (obviously) dependent
    - Self occlusions
    - Fusions

Stochastic Analysis

- Labeling and tracking become achieved in a single merged module.
- Points are labeled and tracked using a MAP weighted by an adaptative a priori probabilistic human model. Two steps:
  - In the first step (tracking): crucial points already labeled in the previous frame are matched with candidate's crucial points.
  - In the second step (detection), we assign to crucial point candidates labels that were not assigned during the first step.

Crucial Point Labeling and Tracking: First Step

- The crucial point selection step produces $z^{(i)}_t = (x, y)$ and associated *intensities* $I^{(i)}$.

- Classification of $(z^{(i)}_t , I^{(i)})$ into one of the six classes: $\Omega = \{h, lf, rf, lh, rh, n\}$.

Crucial Point Labeling and Tracking: First Step

- Candidate $z^{(i)}$ is labeled using a MAP rule. We compute

$$P(\omega_\alpha \mid z_t, z_{t-1})$$

for each $\omega_\alpha \in \Omega_T \bigcup \{n\}$ (Ω being a subset of tracked points)

- The point is assigned to the class that has maximum probability.

$$\omega^* = \arg\max P(\omega_\alpha \mid z_t, z_{t-1})$$

Crucial Point Labeling and Tracking: First Step

- Using Bayes law, the a posteriori probability can be written as a product of three factors, i.e.

$$P(\omega_\alpha \mid z_t, z_{t-1}) = \frac{p(z_t, z_{t-1} \mid \omega_\alpha) P(\omega_\alpha)}{p(z_t, z_{t-1})}$$

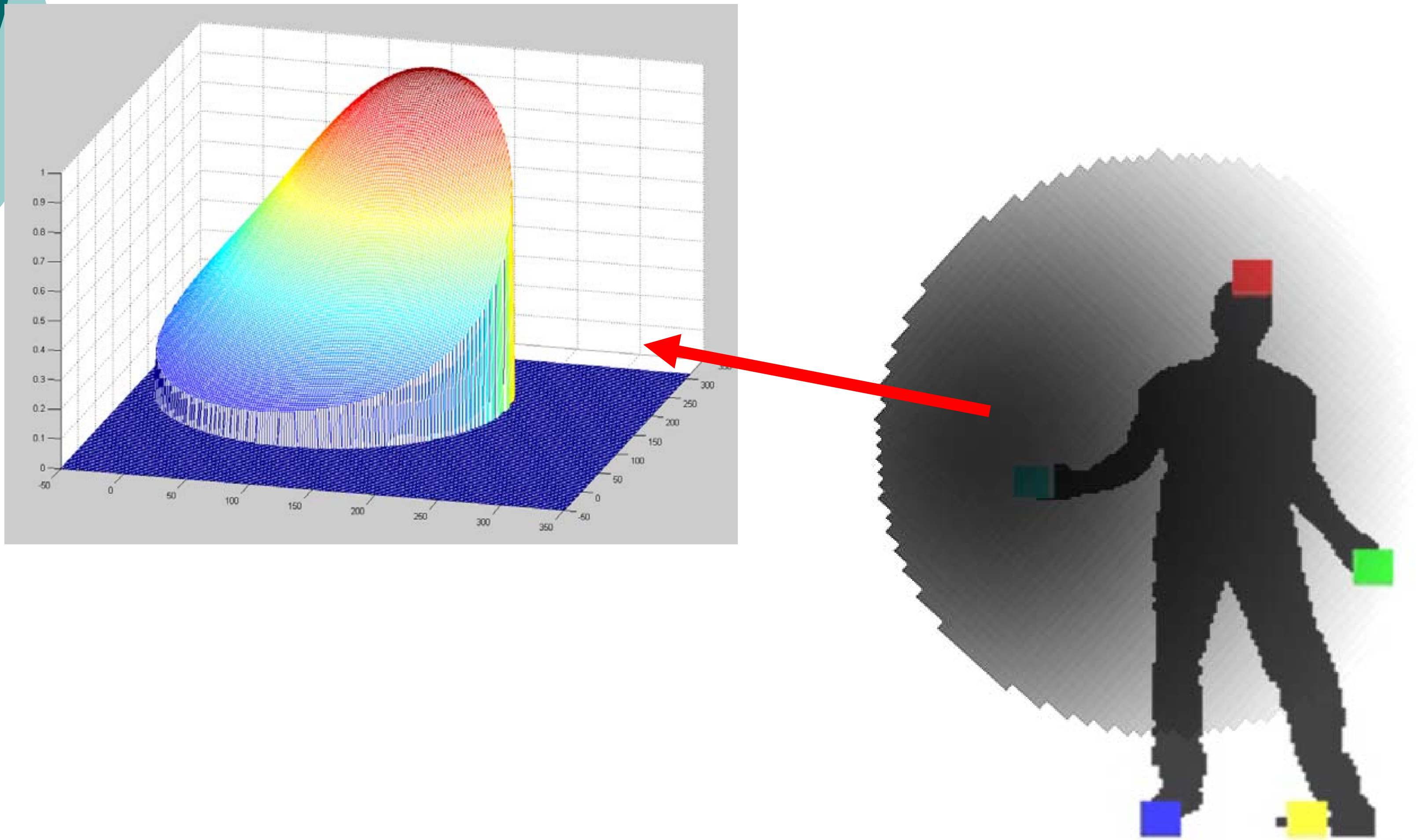
$$\propto p(z_t \mid \omega_\alpha)\, p(z_t \mid z_{t-1}, \omega_\alpha)\, P(\omega_\alpha)$$

A priori knowledge available on that position

$$= N(z_t; z_{t-1}, S_\alpha)$$

A priori knowledge on class ω_α

Prior Probability maps



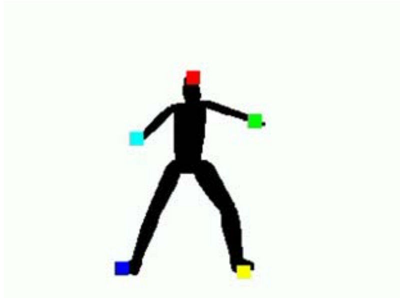
Crucial Point Labeling and Tracking: Second Step

- Detection step: we try to find new crucial points, if any, that were occluded or not detected before.
- We classify the remaining candidate points in the remaining classes applying the same technique but using the a priori probability map and the intensity of the candidate crucial points:

$$P(\omega_\alpha \mid I_t, z_t) \propto p(z_t \mid \omega_\alpha) P(\omega_\alpha) p(I_t \mid \omega_\alpha)$$

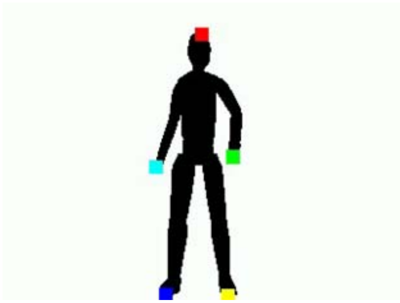
- Hence, the system does not need any kind of forced initialization => for the first frames of a sequence system works in pure detection mode until reliable crucial points are found.

Results. Perfect Segmentation.
Average Error Rate: 3%



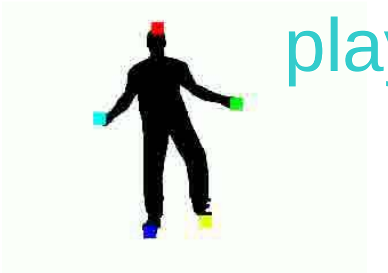
play

	Left Hand	Right Hand	Head	Left Foot	Right Foot
Label Error	5	5	9	0	0
Missed Detect.	3	3	1	0	0
Error Rate (%)	4.4	4.4	5.5	0	0



play

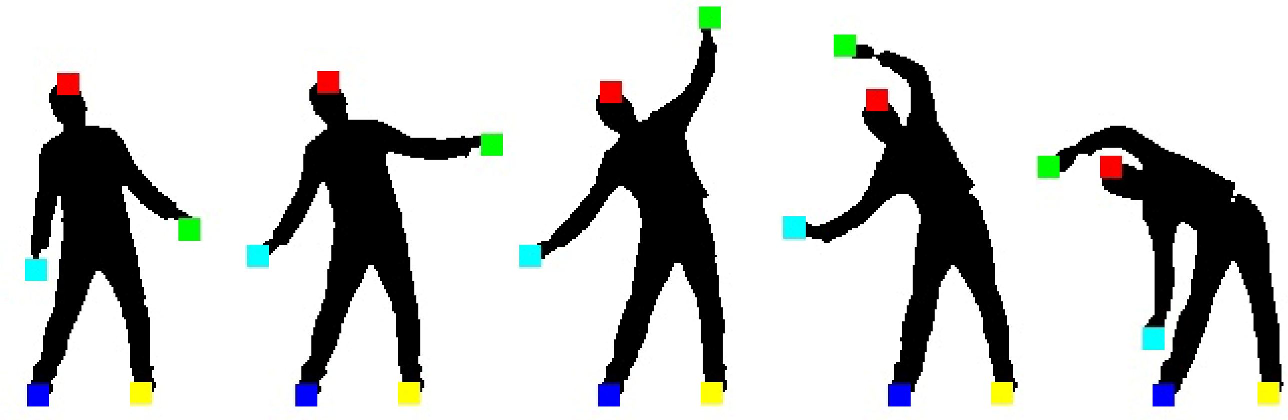
	Left Hand	Right Hand	Head	Left Foot	Right Foot
Label Error	44	16	1	0	0
Missed Detect.	2	6	1	0	0
Error Rate (%)	15	7.1	0.6	0	0



play

	Left Hand	Right Hand	Head	Left Foot	Right Foot
Automatic Segmentation					
Label Error	4	6	1	47	22
Missed Detect.	8	4	3	2	4
Error Rate (%)	3.3	2.7	1.0	13.3	7.0
Manually Corrected Segmentation					
Label Error	4	6	1	1	2
Missed Detect.	8	4	3	2	4
Error Rate (%)	3.3	2.7	1.0	0.8	1.6

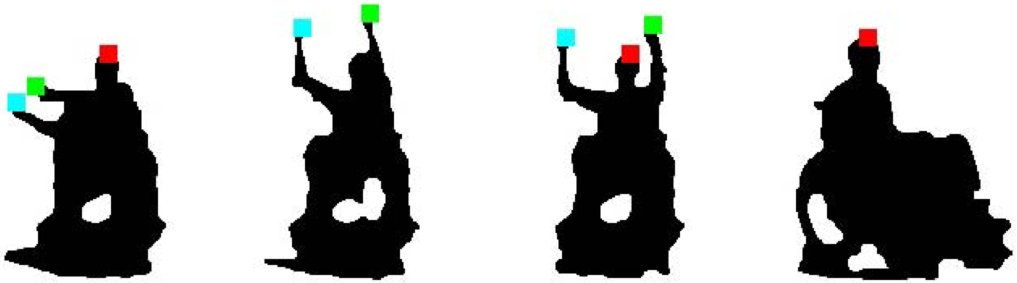
Results. Application 1: Virtual aerobic tranning. 706 frames long. Average Error Rate: 5.86%



play

	Left Hand	Right Hand	Head	Left Foot	Right Foot
Label Error	47	25	2	37	32
Missed Detect.	13	10	22	9	12
Error Rate (%)	8.4	4.9	3.3	6.5	6.2

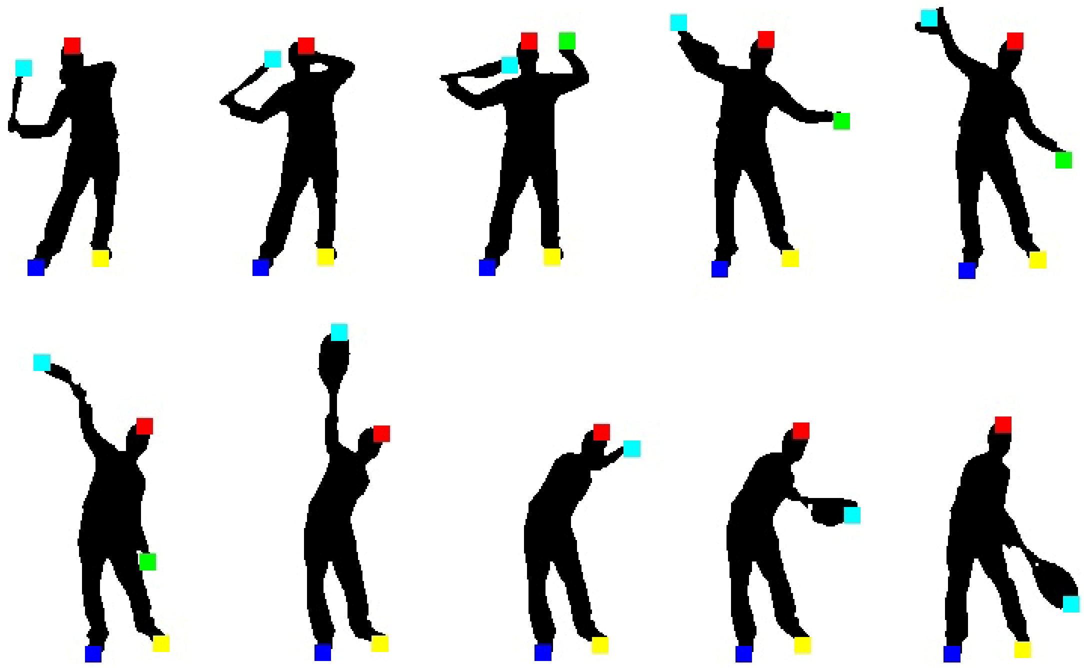
Results. Testing the algorithm flexibility 1: Wheelchair user. 180 frames long. Average Error Rate: 2%



play

	Left Hand	Right Hand	Head	Left Foot	Right Foot
Automatic Segmentation					
Label Error	0	2	1	-	-
Missed Detect.	10	4	1	-	-
Error Rate (%)	5.5	3.3	1.1	-	-

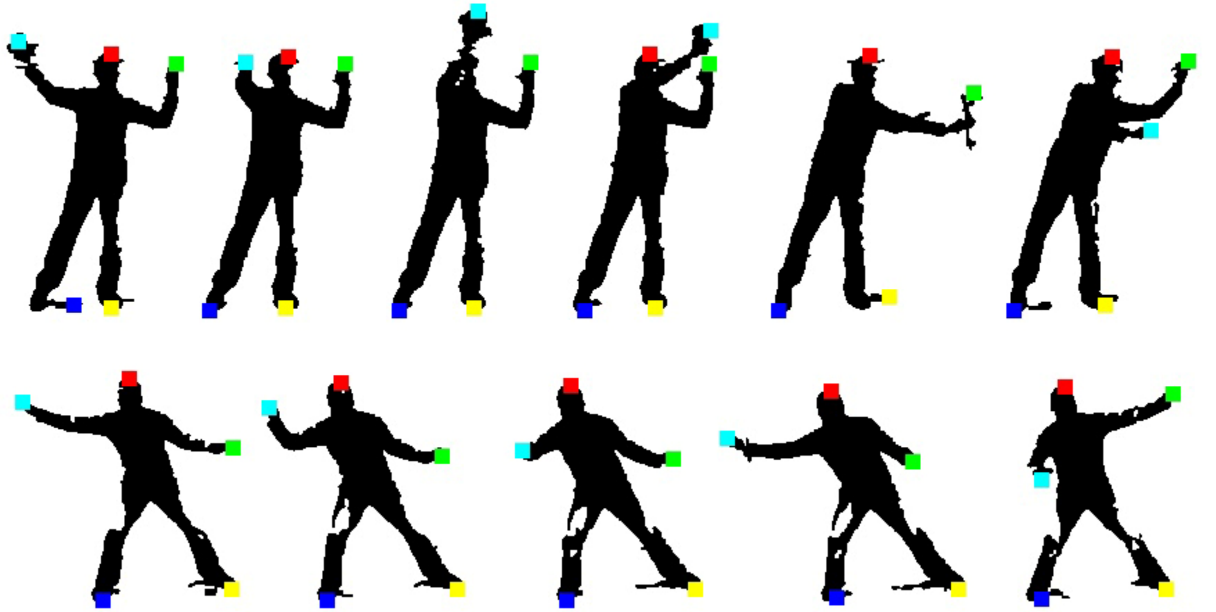
Results. Testing the algorithm flexibility 2. Application 2: Virtual tennis game. 758 frames. Average Error Rate: 2.7%



play

	Left Hand	Racket	Head	Left Foot	Right Foot
Label Error	4	20	2	4	6
Missed Detect.	39	3	11	8	5
Error Rate (%)	5.6	1.7	3.3	1.5	1.4

Testing the robustness regarding segmentation. Application 3: Gestural Navigation.726 frames.AER: 6.76%



play

	Left Hand	Right Hand	Head	Left Foot	Right Foot
Label Error	45	38	10	27	49
Missed Detect.	11	11	16	19	21
Error Rate (%)	7.7	6.7	3.5	6.3	9.6

Conclusions and future work

- Intra-Image Phase
  - Produced the core of the algorithm: crucial point detection using geodesic distance maps
  - Average error rate (2D) of 8,5%
- Inter-Image Phase
  - Robust labeling and tracking
  - Average error rate (2D) of 5,5%
- Future work
  - Bring the whole chain a step further into 3D
    - 2 orthogonal cameras
    - Stereovision
  - Use skin detection as a backup technique