Enabling Technologies for Sports (5XSF0), Module 07

Motion analysis & Sports training case study

Peter H.N. de With

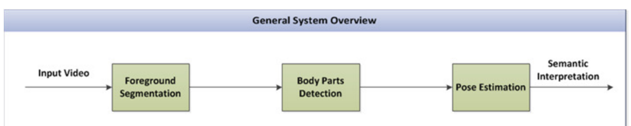
(p.h.n.de.with@tue.nl)

slides version 1.0

Example: Home-based Cycle training

* **Sports training of people at home**
  - Monitoring of people on a regular basis
  - People are stimulated to perform their exercises
  - Improve endurance and condition
* **Vision-based cycling analysis at home**
  - Low requirements: home-trainer/bicycle and video camera are often present and affordable
  - Information about pose, movements and efforts of cyclist
  - Try to estimate health condition improvement

Cycle Training System Overview



* **Input Video**
  - Static Background
  - Basic video camera/webcam
* **Interpretation**
  - Total body configuration (location, orientation)
  - Movement parameters (Pose variation, RPM)

Example Appl. / Tennis Semantic analysis

- **Application: semantic event detection of tennis play**
  - Analyze single match and output important events, like “service” and “net-approach”
  - Analyze double match and output basic tactics like “both back”
  - 85%~90% accuracy

Module 07 Motion Analysis / Overview

- **1. Applications and understanding**
  - Example applications
  - What is “motion”?
- **2. Block-based motion estimation**
  - as used in MPEG-1/2 and H.264 video coding
  - Various algorithm examples
  - Parametric motion: affine motion
- **3. Case study on Home cycle training**
  - Architecture and motion analysis role

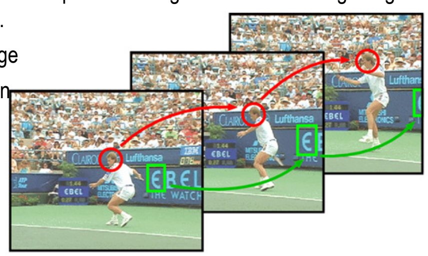
Module 07 – Part 1

Example Applications and Understanding of Motion

Coding, registration, etc. applications, and the description of motion models

Motion Applications / Video Coding – (1)

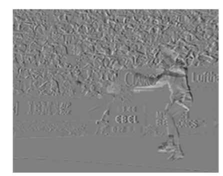
- **High correlation between temporally successive frames.**
- **Use motion-compensation to align same image content.**
  - Copy image content from previous images into current image to get a prediction image.
  - Code residual image (difference between prediction and current image).



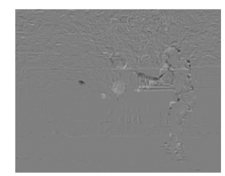
Motion Applications / Video Coding – (2)

- **Residual images without and with motion compensation**
- **Refers to hybrid coding architecture (in coding 5DD40)**

input frame



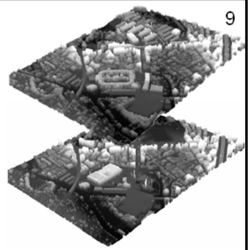
difference to last frame



motion-compensated difference to last frame

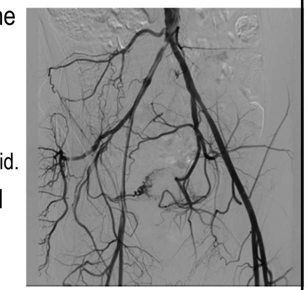
Motion Applicat's / Image registration – (1)

- Two images are captured from the same object at different times (e.g., satellite crossing the same area).
- Changes between images should be identified (changes are visible in residual image).
- Images must be geometrically aligned.



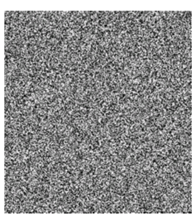
Motion Applicat's / Image registration – (2)

- Medical application, e.g., display the position of blood vessels.
- Two X-ray images are taken:
  - 1 "background" image,
  - 1 image after injection of contrast fluid.
- Difference image shows only blood vessels.
- Key: the two images have to be aligned since the patient will move.

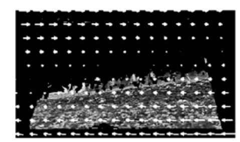


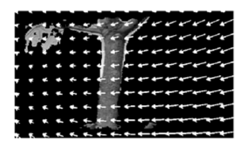
Motion Applic's / Motion segmentation – (1)

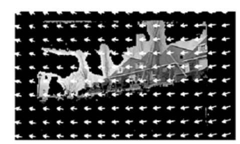
- Main problem of segmentation problem is that computer has no semantic understanding.
- Which areas belong together?



Motion Applic's / Motion segmentation – (2)



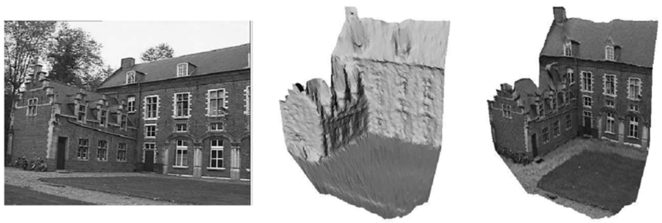




J. Wang and E. Adelson, *Representing Moving Images with Layers*, IEEE Trans. Image Proc., Vol 3, No. 5, pp. 625-638, Sep. 1994.

Motion Appl.’s / Structure from Motion – (1)

- Estimate the 3-D structure of objects from their motion



Marc Pollefeys

Mod 07 Motion Analysis

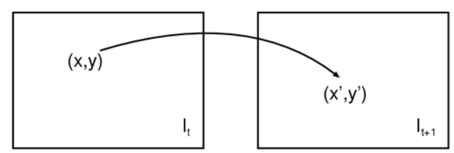
Motion Understanding and Description

Understanding motion / Motion types

- **Object motion vs. Apparent motion**
  - **Object motion**: real motion, not always visible in the image (rotating sphere).
  - **Apparent motion**: observed motion, not always corresponding to real object motion (specular light reflections)
- **Whenever we say “motion”, we mean “apparent motion”.**
  - Key: we only have the image/video available, not the truth!

Understanding / Motion field definition – (1)

- **Motion field describes movement of**
  - a position (x,y) in a first image to
  - a position $(x',y')=m(x,y)$ in a second image

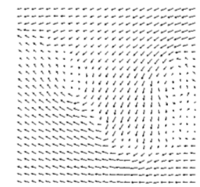


- **How should the function $m(x,y)$ be defined?**

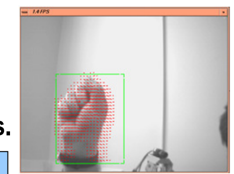
Understanding / Motion field definition – (2)

- **Each parameterization for $m(x,y)$ can represent a sub-class of motions out of all possible motion-fields**
- **Properties of motion-models:**
  - invertible
    $$\exists m^{-1}(x,y) : (x,y) = m^{-1}(m(x,y))$$
  - composable
    $$\exists m_3(x,y) : m_3(x,y) = m_2(m_1(x,y))$$

Motion Description – (1) / Dense motion (pixel-based)



- **Define $m(x,y)$ as a look-up table. Motion of every pixel recorded independently.**
  - good for analysis non-rigid motion (e.g., fluids)
  - not good for video coding since coding of the motion-field is expensive
- **Composable (ignoring discretization issues), but generally not invertible.**
- **Estimation with optical flow algorithms.**



B. Horn and B. Schunck: Determining Optical Flow. Artificial Intelligence, Vol. 17, pages 185-203, 1981.

Motion Description –(2) / Parametric motion

- **Describe motion field with a small number of parameters.**
  - motion induced by camera movement
  - rigid object motion
- **Affine:** $\begin{pmatrix} x' \\ y' \end{pmatrix} = \begin{pmatrix} a_{00} & a_{01} \\ a_{10} & a_{11} \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} t_x \\ t_y \end{pmatrix}$
- **Quadratic:** $x' = ax^2 + bxy + cy^2 + dx + ey + f$
- **Projective:** $x' = \dfrac{a_{00}x + a_{01}y + a_{02}}{a_{20}x + a_{21}y + a_{22}}$
- **Invertibility and composability depend on model.**
- **Projective motion will be discussed elsewhere.**

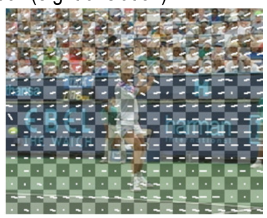
Motion Description –(3) / Parametric motion

- ***Translatorial* motion:** $(x', y') = (x, y) + (t_x, t_y)$
- ***Scaling* only:** $(x', y') = (s \cdot x, s \cdot y)$
- ***Rotation* only:** $\begin{pmatrix} x' \\ y' \end{pmatrix} = \begin{bmatrix} \cos\alpha & \sin\alpha \\ -\sin\alpha & \cos\alpha \end{bmatrix} \begin{pmatrix} x \\ y \end{pmatrix}$
- **All together: *Affine*:** $\begin{pmatrix} x' \\ y' \end{pmatrix} = \begin{pmatrix} a_{00} & a_{01} \\ a_{10} & a_{11} \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} t_x \\ t_y \end{pmatrix}$
- **Invertible (if $\{a_{ij}\}$ nonsingular) and composable.**

Mot. Description –(4) / Block-based motion

21

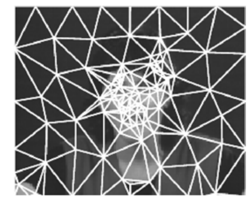
- Pixel-based flow fields are complex
  - Parametric motion can only describe a limited set of motions.
- Subdivide image into blocks and describe motion of each block independently.
  - Use simple motion-model for each block (e.g. translation).
- Compromise between
  - ability to describe general motion-fields,
  - efficiency to code motion
- Easy motion estimation.
- Used in MPEG video coding



Motion Description –(5) / Mesh-based motion

22

- Subdivide image with mesh (e.g., triangular faces).
  - define motion of control points
  - motion within face is interpolated from motion of control points
- Supports concept of general motion-fields
- Efficient to code motion information
- Invertible, (but not easily composable)



Problem of Motion Estimation

23

- Basic assumption: brightness constancy constraint
  - brightness of corresponding pixels in images I_t and I_{t+1} are equal.

$$I_{t+1}\left(m(x,y)\right) = I_t(x,y)$$

- Find motion parameters θ that minimize energy in motion-compensated residual image:

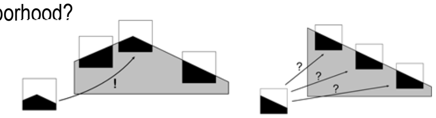
$$\min_{\theta} \sum_{(x,y)} \left(I_{t+1}\left(m_\theta(x,y)\right) - I_t(x,y)\right)^2$$

- Number of parameters depends on motion model
  - Can be **simple**: translation of subimage with 2 parameters only
  - Can be **complex**: pixel-based motion with 2 parameters per pixel!

Motion Estimation / Measurement Problems

24

- Aperture problem
  - How far in neighborhood?



- Repetitive patterns
  - To which location?
- Occlusions
  - Motion of covered/uncovered areas ?

Module 07 – Part 2

Block-based Motion

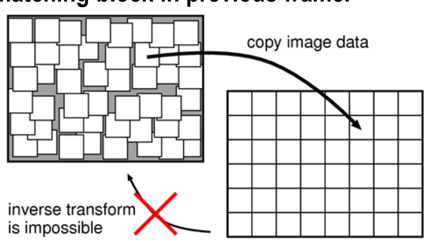
Motion description, estimation principles, example algorithms

Block-Based Motion / Principle

- **Partition image into small blocks B (e.g., 16x16)**
- **Each block is moved with a simple translation.**
- **Translation of a block is specified with motion vector** (dx, dy).
- **Common misunderstanding:**
  - Motion-vector defines where a block in the predicted frame comes from, not where it will move! (Important since it is not invertible.)

Block-based motion / Description vector – 1

- **Subdivide image into regular blocks.**
- **Search matching block in previous frame.**



- **Block-based motion is not invertible and not composable.**

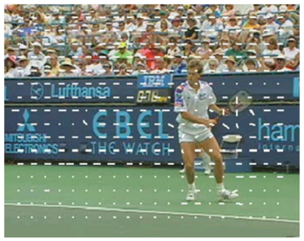
Block-based motion / Description vector – 2

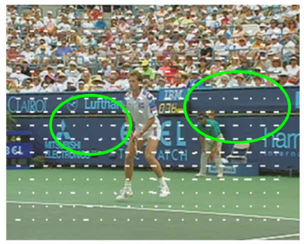
- **Complex motion is approximated**

reference frame   prediction frame

Block-Based Motion / Examples – (1)

* Panning motion

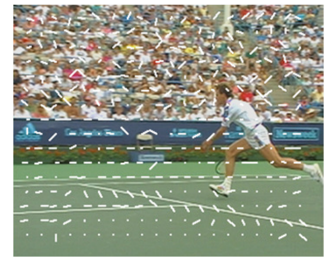




* Notice zero-vectors in homogeneous areas…

Block-Based Motion / Examples – (2)

* Fast motion, displacement larger than search-range.



* Notice vectors along line…

Block-Based Motion / Metrics Estimation

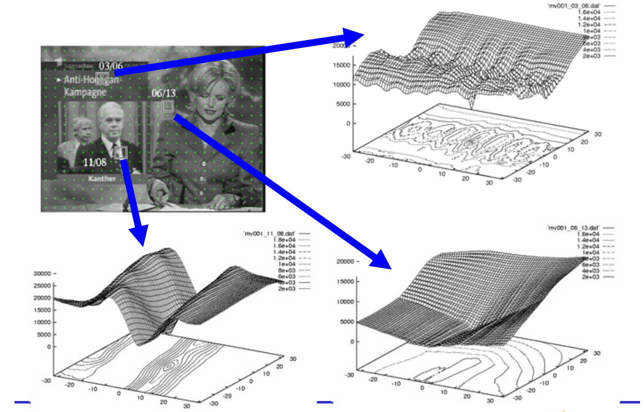
* Since blocks get independent parameters, they can also be estimated independently.
* Find (dx,dy) by minimizing the residual image energy (Mean Square Error):

$$E_{MSE} = \frac{1}{|\mathcal{B}|} \sum_{(x,y)\in\mathcal{B}} (I_{t+1}(x+dx, y+dy) - I_t(x,y))^2$$

* or the Sum of Absolute Differences:

$$E_{SAD} = \sum_{(x,y)\in\mathcal{B}} |I_{t+1}(x+dx, y+dy) - I_t(x,y)|$$

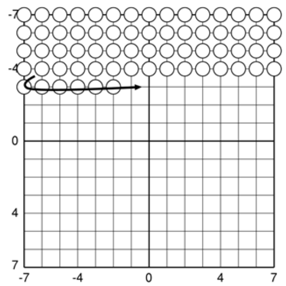
Metrics /Estimation Error forms Surface…



Block-based motion
Algorithm 1: Full Search (FS)

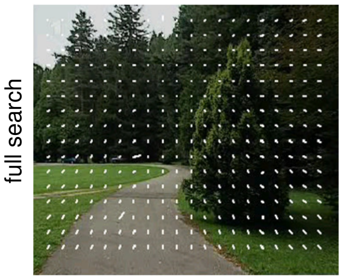
* Define a search window $-L \le dx, dy \le L$
* Compute E(dx,dy) for every possible dx,dy.
* Select dx,dy that gives minimum $E(dx,dy)$.

* Finds optimal solution (not necessarily true motion).
* Computation intensive !

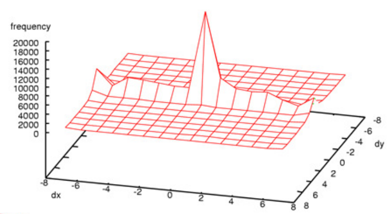


Block-Based Motion /
Comparison of FS Results

full search

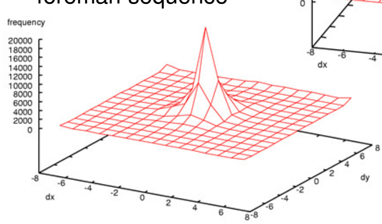


Block-Based Motion /
Distribution of Motion-Vectors



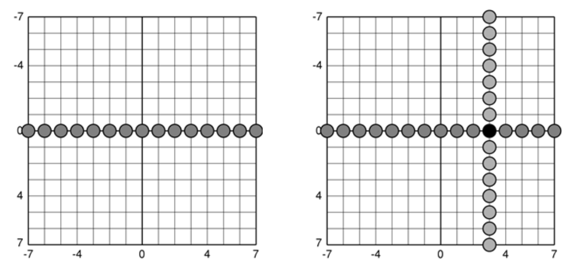
stefan sequence

foreman sequence



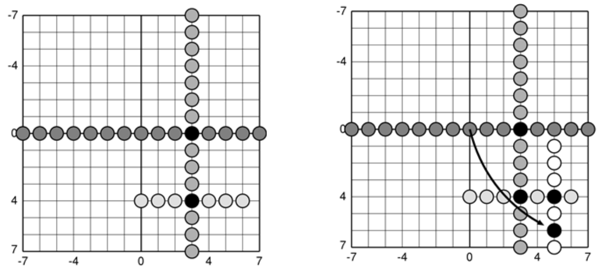
Block-based motion Algorithm 2 – (1)
One-Dimensional Full Search (1DFS)

* Idea: most motion is pure horizontal or vertical.



* Step 1: search pure horizontal motion.
* Step 2: search pure vertical motion (from best hor. motion).

Block-based motion Algorithm 2 – (2)
One-Dimensional Full Search (1DFS)



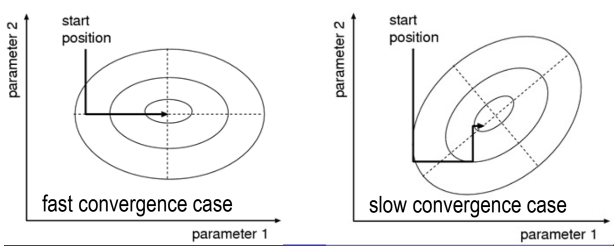
* Step 3 & 4: repeat first two steps with half search range from new position.

Block-based motion Algorithm 2 – (3)
One-Dimensional Full Search (1DFS)

* Algorithm can also be understood as iterative optimization over an alternating subset of the parameters.
  – Since searching for x,y together is too complex, hence:
  – keep y fixed, search x, then
  – keep x fixed, search y, ...

Block-based motion Algorithm 2 – (4)
One-Dimensional Full Search (1DFS)

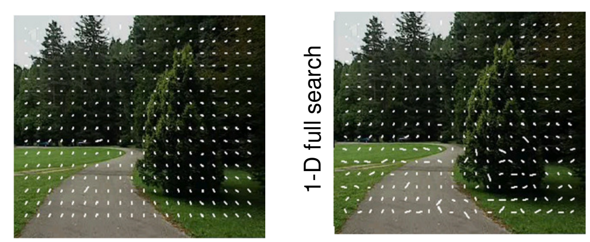
* Works best if error E is decomposable as $E(dx,dy)=f(dx)*g(dy)$ (f and g unknown of course)
* Error isolines:



Block-Based Motion /
Comparison of FS and 1DFS Results

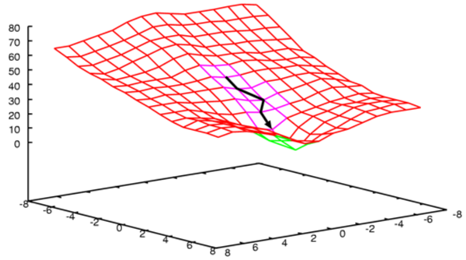
full search

1-D full search



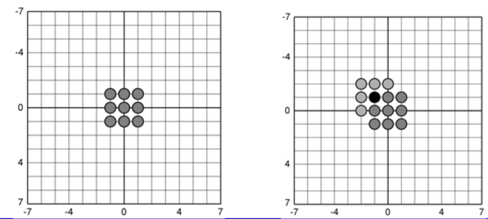
Block-based motion Algorithm 3 – (1)
Gradient Descent (GD)

- Idea: error function is smooth. Explore neighborhood around current position. Proceed into direction of decreasing error.

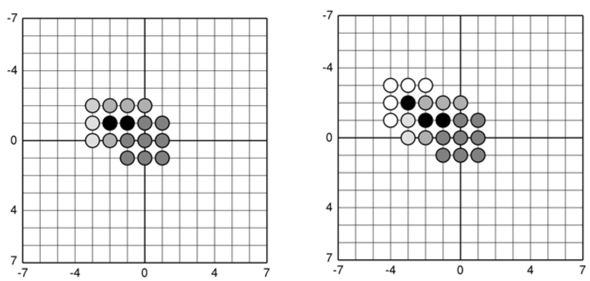


Block-based motion Algorithm 3 – (2)
Gradient descent (GD)

- **Start at (0,0), compute E in 3x3 neighborhood.**
- **If minimum E is in the center,**
  - a local minimum has been found => end the search,
  - else: take new minimum as new search center and iterate.

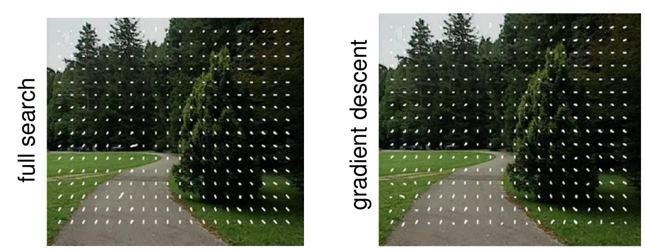


Block-based motion Algorithm 3 – (3)
Gradient descent (GD)



- **Fast for small motions**
- **Gets trapped in local minima (especially in textured regions)**

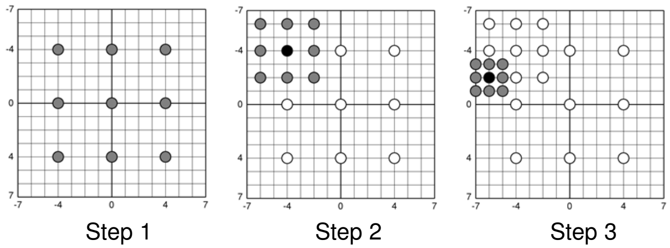
Block-Based Motion / Comparison of FS and GD Results



45

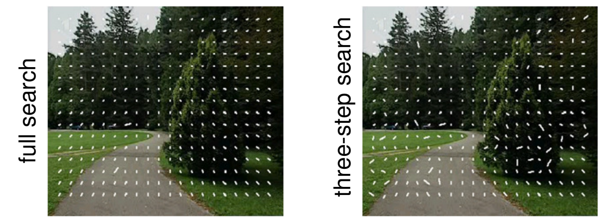
Block-based motion Algorithm 4: Three Step Search (TSS)

- **Idea: error function is smooth. Search on a coarse scale first and then refine to higher accuracy.**



46

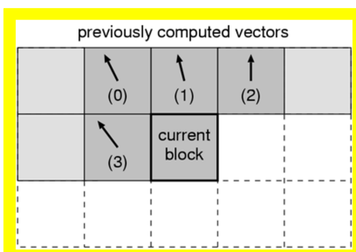
Block-Based Motion Comparison of FS and TSS Results



47

Block-based motion Algorithm 5: Predictive Search – (1)

- Idea: motion of neighboring blocks is similar.
- Start search not at (0,0), but at a vector position computed from the prediction vectors.
- Often, a median vector is computed:

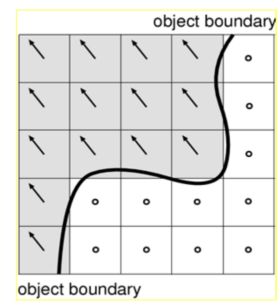


$$(dx_m, dy_m) = (median(dx_1, dx_2, dx_3),\ median(dy_1, dy_2, dy_3))$$

48

Block-based motion Algorithm 5: Predictive Search – (2)

- **Predictor lock problem:**
  - Along motion discontinuities, the prediction is wrong (e.g., between foreground/background regions).
- **Additionally to prediction vector, also test zero-vector.**

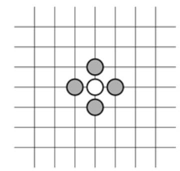


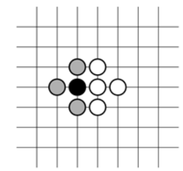
Block-based motion
Algorithm 6 (practical): MVfast - (1)

* Idea: combination of several techniques.
* Classify the motion-type depending on previously computed vectors into three classes (*low, medium, large*).
* Compute maximum magnitude M of the prediction vectors. Classify according to
  - $M < L_1$ => low
  - $L_1 \leq M < L_2$ => medium
  - $L_2 \leq M$ => large
* and use a different estimation algorithm for each class.

Block-based motion
Algorithm 6 (practical): MVfast - (2)

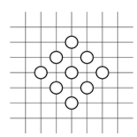
* For the area at: Low motion speed
* start search at (0,0) and carry out gradient-descent seach with small diamond pattern

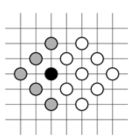

Search pattern

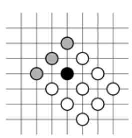

GD-step to the left

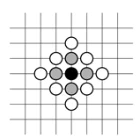
Block-based motion
Algorithm 6 (practical): MVfast - (3)

* Medium motion speed
* Start GD search at (0,0), but use large diamond pattern.



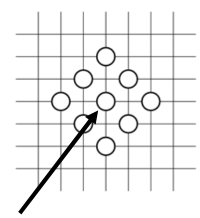

best match in corner
=> proceed


best match at edge
=> proceed


best match in center
=> refine and stop

Block-based motion
Algorithm 6 (practical): MVfast - (4)

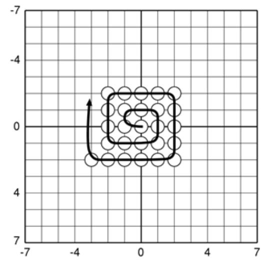
* Large motion speed
* Also use large diamond pattern.
* Start GD search at median prediction vector position.



53

Block-based motion Algorithms: Initial preferred search method

- Preferred search order.
  - Start with vector that probably gives good SAD bound.



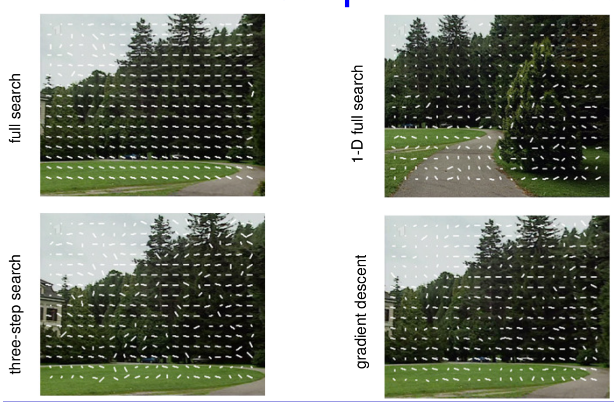
54

Summary of block-based ME algorithms

- **Full-Search:** enumeration of all parameter values.
- **1D-FS:** alternate the (x, y) parameters to search.
- **Gradient-Descent:** assumes smooth error function without local minima.
- **Three-step-search:** search from coarse to fine sampling.
- **Predictive:** use context information to start with a better approximation.
- **MVFast:** classify the expected problem and use an adapted search algorithm for each class.

55

Block-Based Motion: Comparison of Results



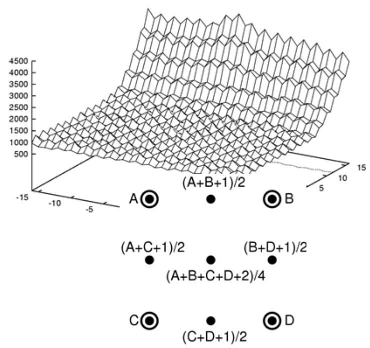
56

Block-based motion Extensions: sub-pel accuracy – (1)

- Object motion is not always a multiple of integer shifts.
- More accurate motion-estimation requires **sub-pixel resolution.**
- MPEG-1/2: **half-pel**
  - Pixel value at half-pel positions obtained with bi-linear interpolation.
- MPEG-4 / H.264: **quarter pel**
  - Half-pel positions using a 8-tap interpolation filter.
  - Quarter-pel position using bi-linear interpolation from half-pel positions.

Block-based motion
Extensions: sub-pel accuracy – (2)

- Error at sub-pel positions is generally lower, as the image is smoothed and noise is filtered
- Hence, first search on full-pel resolution and then
- add a half-pel refinement step.



Block-b. motion / Concluding Remarks

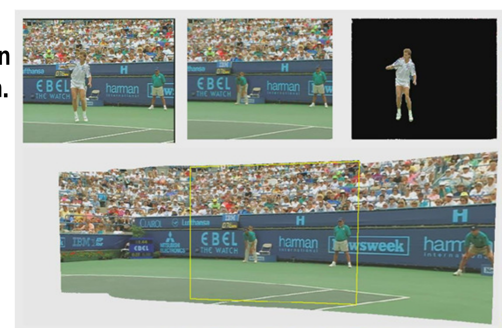
- Block-motion is a simple approach, but still the most frequently-used technique for video-coding (even for wavelet coders).
  - can approximate any motion
  - easy estimation
- Block motion can also be used as first step to estimate parametric motion from these vectors.
- The presented estimation principles can be reused for many other tasks than motion estimation.
- Parametric Motion Estim. approaches real motion better.

Mod 07 Motion Analysis

Parametric Motion: Affine Motion (revisited)

Parametric Motion / Motivation

- MPEG-4 camera motion compensation.
- Camera motion model required to build background sprite.



[insert MPEG-4 stefan sequence 2]

Parametric motion / System aspects

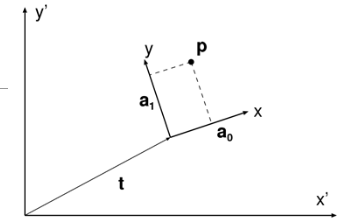
- Motion in the image is usually composed of
  - camera motion,
  - rigid object motion, and
  - non-rigid object motion.
- The camera and rigid objects cannot move arbitrarily.
- Their motion can be described with some parameters derived from the geometrical setup.
- We will describe the observed image motion in terms of the 3-D object motion (we do not know what happened..).

Affine 2D Coordinate Transform – (1)

- Express coordinate frame (x,y) in coordinate frame (x',y').
- In motion estimation:
  - (x,y) could be the coordinate in the last frame, and
  - (x',y') the corresponding coordinate in the current frame.

$$\mathbf{p} = (x \;\; y)^\top$$
$$\mathbf{p}' = (x' \;\; y')^\top$$

$$\begin{aligned}\mathbf{p}' &= \mathbf{t} + x \cdot \mathbf{a}_0 + y \cdot \mathbf{a}_1 \\ &= (\mathbf{a}_0 \;\; \mathbf{a}_1)\,\mathbf{p} + \mathbf{t} \\ &= \mathbf{A}\mathbf{p} + \mathbf{t}\end{aligned}$$



Affine 2D Coordinate Transform – (5)

- Rotation matrix for rotation by an angle α

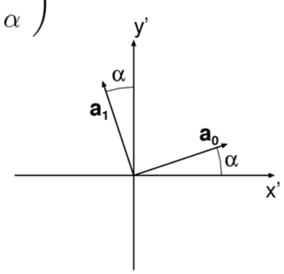
$$\mathbf{a}_0 = \begin{pmatrix}\cos\alpha \\ \sin\alpha\end{pmatrix} \quad \mathbf{a}_1 = \begin{pmatrix}-\sin\alpha \\ \cos\alpha\end{pmatrix}$$

$$\mathbf{A} = \begin{pmatrix}\cos\alpha & -\sin\alpha \\ \sin\alpha & \cos\alpha\end{pmatrix}$$

- rotation by $-\alpha$

$$\mathbf{A}^{-1} = \begin{pmatrix}\cos\alpha & \sin\alpha \\ -\sin\alpha & \cos\alpha\end{pmatrix}$$

$$\mathbf{A}^{-1} = \mathbf{A}^\top$$



Affine 2D Coordinate Transform – (9)

- Here the trick: augment each point-coordinate with a dummy '1':

$$\mathbf{p} = \begin{pmatrix}x \\ y \\ 1\end{pmatrix}$$

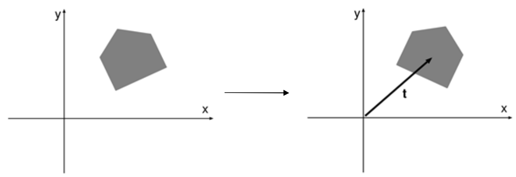
- Now, write affine transform as simple matrix multiplication:

$$\begin{pmatrix}x' \\ y' \\ 1\end{pmatrix} = \begin{bmatrix}a_{00} & a_{01} & t_x \\ a_{10} & a_{11} & t_y \\ 0 & 0 & 1\end{bmatrix}\begin{pmatrix}x \\ y \\ 1\end{pmatrix}$$

$$\mathbf{p}' = \mathbf{A}\mathbf{p} + \mathbf{t}$$

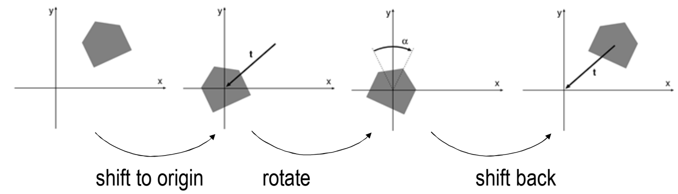
Affine 2D Coordinate Transform
Example: Transform Sequence – (1)

* What is the transformation between these two images?



Affine 2D Coordinate Transform
Example: Transform Sequence – (2)

* Break the transformation into sequence of elementary steps.



$$\mathbf{A_1} = \begin{pmatrix} 1 & 0 & t_x \\ 0 & 1 & t_y \\ 0 & 0 & 1 \end{pmatrix} \quad \mathbf{A_2} = \begin{pmatrix} \cos\alpha & -\sin\alpha & 0 \\ \sin\alpha & \cos\alpha & 0 \\ 0 & 0 & 1 \end{pmatrix} \quad \mathbf{A_3} = \begin{pmatrix} 1 & 0 & -t_x \\ 0 & 1 & -t_y \\ 0 & 0 & 1 \end{pmatrix}$$

Affine 2D Coordinate Transform
Example: Transform Sequence – (3)

* Successive steps:

$$\mathbf{A_1} = \begin{pmatrix} 1 & 0 & t_x \\ 0 & 1 & t_y \\ 0 & 0 & 1 \end{pmatrix} \quad \mathbf{A_2} = \begin{pmatrix} \cos\alpha & -\sin\alpha & 0 \\ \sin\alpha & \cos\alpha & 0 \\ 0 & 0 & 1 \end{pmatrix} \quad \mathbf{A_3} = \begin{pmatrix} 1 & 0 & -t_x \\ 0 & 1 & -t_y \\ 0 & 0 & 1 \end{pmatrix}$$

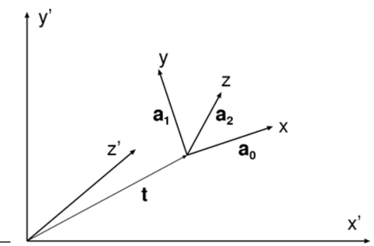
* Combined transform:

$$\mathbf{p}' = \mathbf{A_3 A_2 A_1 p}$$

* Multiplied together: $\begin{pmatrix} t_x \cos\alpha & -t_y \sin\alpha & -t_x \\ t_x \sin\alpha & t_y \cos\alpha & -t_y \\ 0 & 0 & 1 \end{pmatrix}$

Affine 3D Coordinate Transform – (1)

* Straight-forward extension to affine 3-D motion



$$\mathbf{p} = (x \;\; y \;\; z)^\top$$
$$\mathbf{p}' = (x' \;\; y' \;\; z')^\top$$

$$\begin{aligned} \mathbf{p}' &= \mathbf{t} + x \cdot \mathbf{a_0} + y \cdot \mathbf{a_1} + z \cdot \mathbf{a_2} \\ &= (\mathbf{a_0} \;\; \mathbf{a_1} \;\; \mathbf{a_2})\, \mathbf{p} + \mathbf{t} \\ &= \mathbf{Ap} + \mathbf{t} \end{aligned}$$

Affine transform / Rotation in 3D Space – (1)

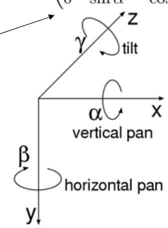
- **Elementary rotation matrices:**

around z-axis, around y-axis, around x-axis

$$\mathbf{R_z} = \begin{pmatrix} \cos\gamma & -\sin\gamma & 0 \\ \sin\gamma & \cos\gamma & 0 \\ 0 & 0 & 1 \end{pmatrix} \quad \mathbf{R_y} = \begin{pmatrix} \cos\beta & 0 & \sin\beta \\ 0 & 1 & 0 \\ -\sin\beta & 0 & \cos\beta \end{pmatrix} \quad \mathbf{R_x} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos\alpha & -\sin\alpha \\ 0 & \sin\alpha & \cos\alpha \end{pmatrix}$$

- **Note: points on rotation axis do not move.**
- **The inverse is the transpose:**

$$\mathbf{R}^{-1} = \mathbf{R}^{\top}$$



References (Block-Matching)

- J. Mitchell, W. Pennebaker. MPEG Video Compession Standard, Chapter 13: Motion Estimation. Kluwer.
- TSS: T. Koga, K. Iinuma, A. Hirano, Y. Iijima, and T. Ishiguro, Motion-compensated interframe coding for video conferencing. Proc. NTC 81, pp. C9.6.1-9.6.5, Dec. 1981.
- TSS: R. Li, B. Zeng, M. Liou. A New Three-Step Search Algorithm for Block Motion Estimation. IEEE Trans. Circuits Systems Video Technology, Vol. 4, No. 4, Aug. 1994.
- GS: L. Liu, E. Feig: A Block-Based Gradient Descent Search Algorithm for Block Motion Estimation in Video Coding, IEEE Trans. Circ. Sys. Video Techn., Vol. 6, No. 4, 1996.
- 1DFS: M. Chen, L. Chen, T. Chiueh: One-Dimensional Full Search Motion Estimation Algorithm For Video Coding, IEEE Trans. CSVT, Vol. 4, No. 5, Oct. 1994.
- SEA: M. Brünig, B. Menser: Fast Full Search Block Matching using Subblocks and Successive Approximation of the Error Measure. SPIE Image and Video Comm. & Proc., Vol. 3974, pp. 235-244, 2000.

Module 07 – Part 3

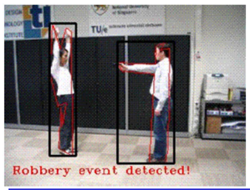
Home cycle training & motion analysis

Application issues, principles for measurement and results

Example: Home-based Cycle training

- **Sports training of people at home**
  - Monitoring of people on a regular basis
  - People are stimulated to perform their exercises
  - Improve endurance and condition
- **Vision-based cycling analysis at home**
  - Low requirements: home-trainer/bicycle and video camera are often present and affordable
  - Information about pose, movements and efforts of cyclist
  - Try to estimate health condition improvement

Introduction / Pose Estimation



- Behavior analysis applications
- Surveillance applications
- Control applications

Introduction / Heart Rate Monitoring

- Electrocardiography (ECG)
- Photoplethysmography (PPG)
- Phonocardiography (PCG)

Problem Description: Pose and Heart

Pose Estimation

- **Project Goal:**
  Automatically identify and estimate the cyclists' pose and movements based on the video images of a single camera.
- **Requirements:**
  - Single camera
  - Detect body parts, both side and frontal camera viewpoint
  - Extract pose and movement parameters
  - Marker-less
  - Invariant to human appearance

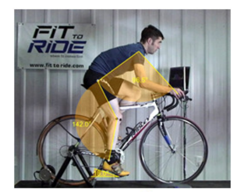
Heart Rate Detection

Project Goal:
Automatic heart rate detection based on the video images of a single camera.

Requirements:
- **Single camera**
- **Unobtrusive**
- **Robust against subject movements**

Related Work / PE for sports analysis

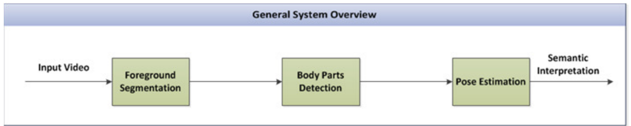
- Open source
- Marker-less
- Manual or semi-automated tracking of limbs



- Commercial
- Marker-based

System Overview (1)

77



- **Input Video**
  - Static Background
  - Basic video camera/webcam
- **Interpretation**
  - Total body configuration (location, orientation)
  - Movement parameters (Pose variation, RPM)

System Overview (2)

78

- **Foreground Segmentation**
  - Background image

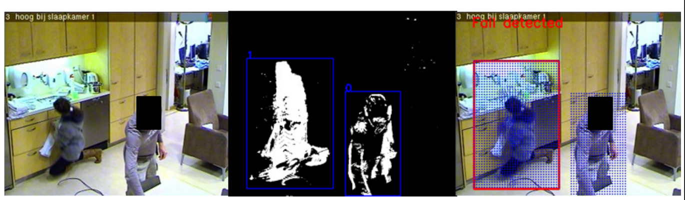
  - Subtract

  - Morphological filtering

Experimental systems

79

- **Assess motion only in Region Of Interest (ROI)**
  - Gaussian filter blurring/noise reduction
  - ROI detection based on motion history image
  - Dense motion vectors calculation in ROI
  - Classification of each ROI into fall/no-fall classes

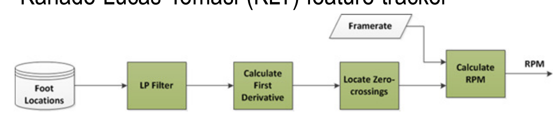


Input Video Frame | Motion History Image | Output Video Frame

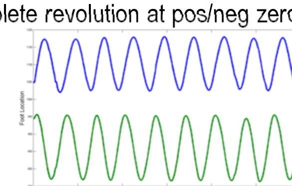
System Overview (4)

80

- **Foot Motion Estimation**
  - Kanade-Lucas-Tomasi (KLT) feature tracker



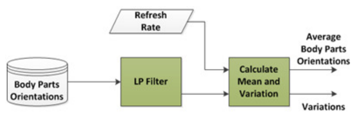
  - One complete revolution at pos/neg zero-crossings



System Overview (5)

81

* Arm and Leg Estimation
  – Block matching (binary mask) on binary foreground mask

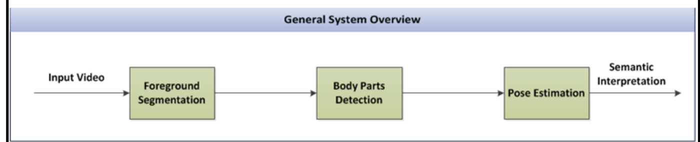


Frame: 174
Torso angle: -8 +/- 1 degree
Upperarm angle: 44 +/- 1 degree
Lowerarm angle: 63 +/- 1 degree
Upperleg angle: 69 +/- 24 degree
Lowerleg angle: -28 +/- 20 degree
RPM: 53

Pose estimation / Methodology - (1)

82

System Overview



- Cyclist:
  - Front , Side, or
  - Entire human body visible
- Static background
- Camera, video/webcam, 25 fps

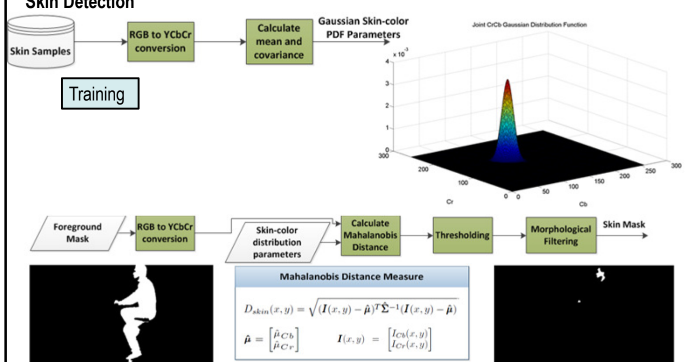
- Total body configuration:
  - Locations (all limbs)
  - Orientations
- Movement parameters:
  - Pose variation
  - RPM

TU/e PdW-SZ / 2015 Fac. EE SPS-VCA Enabling Technologies for Sports / 5XSF0 / Module 07 Motion Estimation VCA

Pose estimation / Methodology – (2)

83

Skin Detection



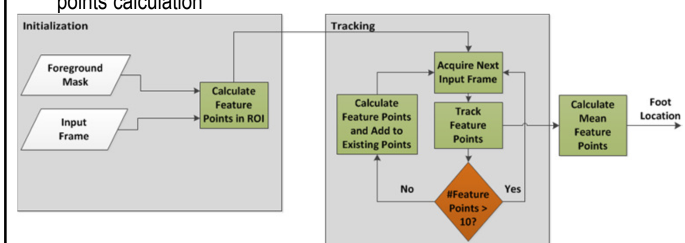
TU/e PdW-SZ / 2015 Fac. EE SPS-VCA Enabling Technologies for Sports / 5XSF0 / Module 07 Motion Estimation VCA

Pose estimation / Methodology – (3)

84

Foot Tracking (robust due to periodic overlap)

- Kanade-Lucas-Tomasi (KLT) feature tracker
- Minimum Eigenvalue algorithm for feature points calculation

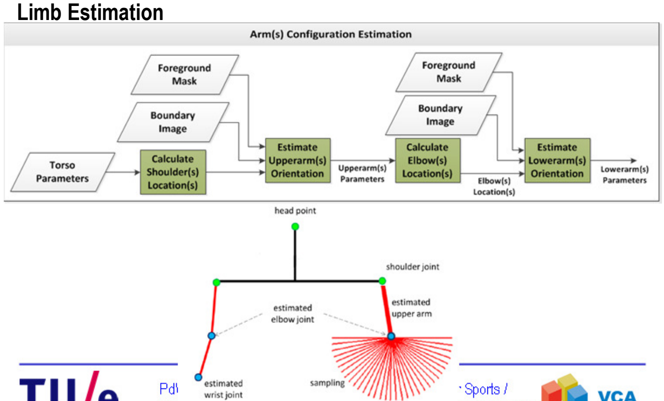


TU/e PdW-SZ / 2015 Fac. EE SPS-VCA Enabling Technologies for Sports / 5XSF0 / Module 07 Motion Estimation VCA

Pose estimation / Methodology – (4)

85

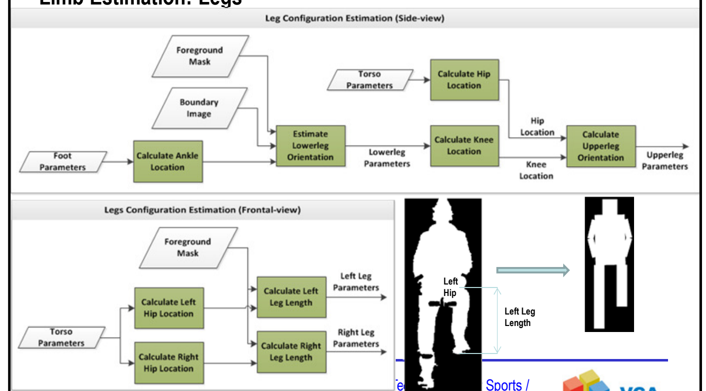
Limb Estimation



Pose estimation / Methodology – (5)

86

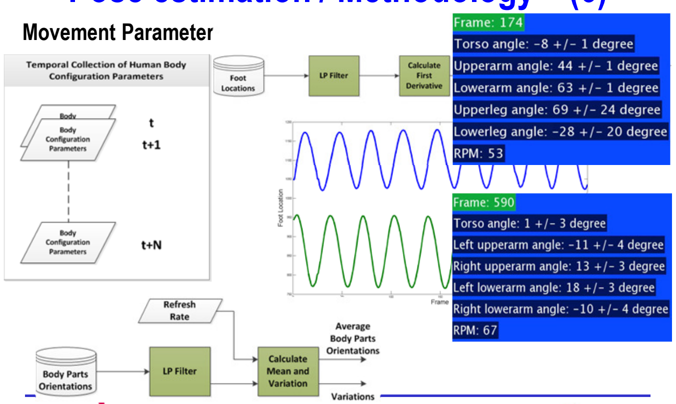
Limb Estimation: Legs



Pose estimation / Methodology – (6)

87

Movement Parameter

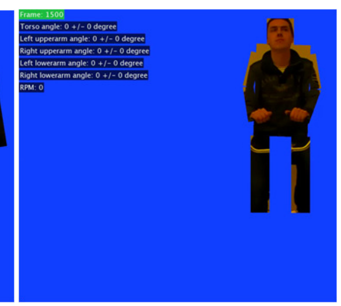


Pose estimation / Visual Results

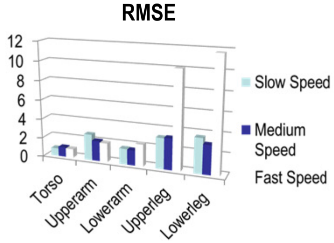
88

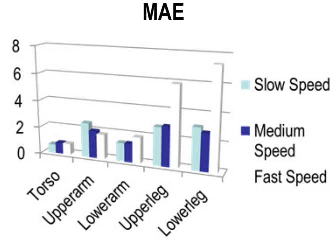
Side View

Frontal View



Pose estim. / Results Side View





- Mean absolute error within [0.6-7.4] degrees
- No significant difference in arms and torso orientation errors for different cycling speeds
- Leg orientation error increases with cycling speed due to inaccuracies in foot tracking

Demonstration