Pose Estimation: Alternatives

Computer Vision for Computer Graphics Seminar, SS 2013
Max-Planck-Institut für Informatik

Zornitsa Kostadinova

Department of Computer Science, University of Saarland

Jul 02, 2013
Contents

Introduction

1. Sum of Gaussians
   Kinematic Skeleton
   Body Approximation
   Objective Function
   Initialization and Tracking
   SoG Similarity
   Conclusions

2. Body Part Recognition
   Data
   Body Part Labeling
   Image Features
   Decision Forests
   Proposed Joint Positions

Summary
Introduction

1. Fast Articulated Motion Tracking using a Sums of Gaussians Body Model
   - Carsten Stoll, Nils Hasler, Juergen Gall, Hans-Peter Seidel, Christian Theobalt
   - ICCV 2011
   - MPII

2. Real-Time Human Pose Recognition in Parts from a Single Depth Image
   - Jamie Shotton, Andrew Fitzgibbon, Mat Cook, Toby Sharp, Mark Finocchio, Richard Moore, Alex Kipman, and Andrew Blake
   - CVPR 2011
   - MRC, Xbox Incubation
Pose Estimation

Introduction

1. Sum of Gaussians
   - Kinematic Skeleton
   - Body Approximation
   - Objective Function
   - Initialization and Tracking
   - SoG Similarity

Conclusions

2. Body Part Recognition
   - Data
   - Body Part Labeling
   - Image Features
   - Decision Forests
   - Proposed Joint Positions

Summary
Alternative to? Related Work

Previously on CVCG Seminar

Pose estimation: Foundations

Tracking People with Twists and Exponential Maps [Bregler 1998]

- track 3D pose of a rigid object under scaled orthographic projection
- product of exponential maps and twist motions
Previously on CVCG Seminar

Pose estimation: Foundations

Optimization and Filtering for Human Motion Capture
[Gall 2010]

Multi-layer framework:

1. global stochastic optimization
2. filtering
3. local optimization
Previously on CVCG Seminar

Shape capture: Performance capture

Two ways towards markerless MoCap:

Acurate, robust, unsupervised methods, relying on Laplacian deformations to obtain vertices locations / geometry.
Previously on CVCG Seminar

Shape capture: Facial performance capture  
Darya Dedik

Anchor frames [Beeler 2011]  
Laplacian regularization [Valgaerts 2012]

Lighting and illumination, temporal correspondence, multiple cameras, high-quality, detailed face geometry.
Fast Articulated Motion Tracking using a Sums of Gaussians Body Model

In a nutshell:

- multi-view video – multiple cameras (more than 6)
- novel human model, optimization problem with nice analytical properties
- no background subtraction
- efficient, 5-15 fps, real-time

[ www mpi-inf mpg de/~stoll/ ]
Method Overview

**Input:** videos from $n_{cam}$ synchronized, calibrated, static cameras

**Output:** pose parameters allowing to reconstruct a kinematic skeleton
Method Overview

1. actor-specific **3D body model** (Gaussians)
   - default human model
   - several multi-view images
     - poses selected to articulate wide range of skeletal joints
     - manually segmented
     - allow accurate estimate of bone lengths

2. convert input images to **2D SoG**, use **quad-tree**

3. estimate the skeletal pose of the actor by maximizing the similarity between 1. and 2.
The Human Body Model

Kinematic skeleton

[ www.mpi-inf.mpg.de/~edeaguia/mocapskeleton.html ]

Body approximation (sum of spatial 3D Gaussians)
The Human Body Model - Kinematic Skeleton

58 joints described by 61 parameters Λ:

- 58 rotational
- 3 translational
- each joint has a limit range of motion ∈ [l_l, l_h]

Θ - n_{DoF} pose parameters comprise a separate complexity hierarchy

\[ Λ = \begin{pmatrix} Θ \\ \Lambda \end{pmatrix} \]

\[ M \]

influence weights

→ smooth bending allows to reproduce natural deformations of the spine and clavicles*
Mathematical Model of Spatial Similarity

Introduction

1. Sum of Gaussians
   - Kinematic Skeleton
   - Body Approximation
   - Objective Function
   - Initialization and Tracking
   - SoG Similarity
   - Conclusions

2. Body Part Recognition
   - Data
   - Body Part Labeling
   - Image Features
   - Decision Forests
   - Proposed Joint Positions

Summary
Model of Spatial Similarity

\[ B(x) = \exp \left( -\frac{||x - \mu||^2}{2\sigma^2} \right) \]

- \( \mu \in \mathbb{R}^d \) - mean, \( \sigma^2 \) - variance

One equation, two use cases:

- \( d = 2 \) image domain \( \Omega \in \mathbb{R}^2 \)
- \( d = 3 \) 3D human body model

[ www.leftovercurrency.com ]
Model of Spatial Similarity

\[ B(x) = \exp \left( -\frac{||x - \mu||^2}{2\sigma^2} \right) \]

\( \mu \in \mathbb{R}^d \) - mean, \( \sigma^2 \) - variance

One equation, two use cases:

- \( d = 2 \) image domain \( \Omega \in \mathbb{R}^2 \)
- \( d = 3 \) 3D human body model
Model of Spatial Similarity

\[ B(x) = \exp \left( -\frac{||x - \mu||^2}{2\sigma^2} \right) \]

\( \mu \in \mathbb{R}^d \) - mean, \( \sigma^2 \) - variance

One equation, two use cases:

- \( d = 2 \) image domain \( \Omega \in \mathbb{R}^2 \)
- \( d = 3 \) 3D human body model
Spatial Similarity through Sum of Gaussians

Combination of several spatial Gaussians into a Sum of Gaussians model

\[ \mathcal{K}(x) = \sum_{i=1}^{n} B_i(x) \]

Again, two use cases:

- \( d = 2 \) spatial extent of super-pixels within a similar-colored cluster
- \( d = 3 \) infinite spatial support, but influence weights decrease fast

Additionally, color model \( C = \{c_i\}_{i=1}^{n} \)

Associate with every Gaussian \( B_i \) its respective color value \( c_i \in \mathbb{R}^3 \). Used HSV color scheme.
Image Domain \( (d = 2) \) - Approximating Images using SoG

**Have:** Image \( I \)

**Want:** The image approximation \( K_I \) in terms of SoG

Consistent pixel regions

**Problem:** Having a separate Gaussian (and color value) for every image pixel is too wasteful on resources. Performance is important, so cluster image pixels based on color into regions.

**Idea:** Use a quad-tree
Quad-tree

- partition a 2D space by recursively subdividing it into 4 quadrants (regions)
- for 3D - oct-tree
- application in gaming / rendering
Quad-tree in SoG

- threshold $\epsilon_{col}$ for the standard deviation of colors in a quad-tree node
- 8 – max quad-tree depth
- quadratic cluster $\approx$ Gaussian $B_i$ such that:
  - $\mu$ is cluster center
  - $\sigma = \frac{\text{side length of node}}{2}$
  - $c_i$ is the average color of the cluster
Human Body Model \((d = 3)\)

- default human model - 58 joints
- attach to each parent joint of the skeleton a Gaussian
- get a 3D SoG model \(\mathcal{K}_m\)
  - parameterized by the kinematic skeleton pose parameters \(\Theta\)
- “personalize” the model - adapt it to match the shape and color statistics of the actor
Objective Function

**Want:** From the set of input multiview images $I$, estimate the pose-parameters $\Theta$ of the kinematic skeleton

Define an energy function:

$$E(\Theta) = E(\Theta) - w_I E_{lim}(M\Theta) - w_a E_{acc}(\Theta)$$

- **Similarity function** $E$
- **Skeleton term** $E_{lim}$
- **Motion-specific term** $E_{acc}$
Objective Function

Want: From the set of input multiview images I, estimate the pose-parameters \( \Theta \) of the kinematic skeleton

Define an energy function:

\[
\mathcal{E}(\Theta) = E(\Theta) - w_l E_{lim}(M\Theta) - w_a E_{acc}(\Theta)
\]

- **Similarity function** \( E \)
- **Skeleton term** \( E_{lim} \)
- **Motion-specific term** \( E_{acc} \)
Objective Function – Similarity

Define an energy function:

\[ \mathcal{E}(\Theta) = E(\Theta) - w_l E_{lim}(M) - w_a E_{acc}(\Theta) \]

Similarity function \( E \) – match between:

- body model in this particular pose
  - parameterized by \( \Theta \)
- all input images from all the cameras at the current moment
Objective Function – Joint Constraints

Define an energy function:

\[ \mathcal{E}(\Theta) = E(\Theta) - w_I E_{lim}(\mathcal{M}(\Theta)) - w_a E_{acc}(\Theta) \]

Skeleton term \( E_{lim} \)

- soft constraints on the range of motion of the joints
- prevents physically implausible movements
- upper and lower joint limits \( l_l, l_h \) that we associated with the kinematic skeleton; \( \Lambda \)
Objective Function – Smoothness of Motion

Define an energy function:

\[ \mathcal{E}(\Theta) = E(\Theta) - w_1 E_{lim}(\mathcal{M}\Theta) - w_a E_{acc}(\Theta) \]

Motion-specific term \( E_{acc}(\Theta_t) \)

- smoothness constraint to penalize high acceleration in parameter space
- tradeoff between jittered motion and decreased tracking accuracy
- take into account the pose estimates of the previous two frames \( \Theta_{t-1}, \Theta_{t-2} \)
Initialization

1. rough manual initialization of pose parameters $\Theta$
2. joint optimization – maximize the similarity function $E$ of actor’s silhouette
   - use gradient ascent, as in tracking later
3. back-project the color images onto the 3D Gaussian body model

**Insight:** Initialization is just a special case of tracking!
**Tracking**

**Have:** Video sequence consisting of \( m \) frames  
**Want:** Estimate pose parameters for every time-step \( \Theta_t \)  
**Solution:**

1. extrapolate the motion, taking into account the results from the previous 2 time-steps  
2. optimize the pose parameter by maximizing the Energy function \( E(\Theta) \)
   - for gradient ascent to be efficient, objective function has to have *nice* properties  
   - ours has them - similarity measure is* continuous and differentiable
Tracking (Continued)

2. optimize the pose parameter by maximizing the Energy function $\mathcal{E}(\Theta)$
   - further speed-up performance by using *conditioned* gradient ascent
     at time step $t$:
     \[
     \Theta_{i+1} = \Theta_i + \nabla \mathcal{E}(\Theta_i) \circ \sigma_i \\
     \text{conditioning vector}
     \]
   - update $\sigma_i$ on every iteration step ($i$):
     - increase step-size where gradient sign is constant
     - decrease it if the ascent is “zig-zagging”
   - reduce significantly number of iterations until convergence
     - c.f. back-propagation in *resilient* Neural Networks
SoG Similarity

- **2D-2D Similarity**
  - two SoG models $\mathcal{K}_a, \mathcal{K}_b$ and their respective color models $C_a, C_b$
  - define the similarity measure between models $E(\mathcal{K}_a, \mathcal{K}_b, C_a, C_b)$
    - similarity measure between color models
    - 2D Gaussians

- **3D-2D Similarity**
  - project a 3D Gaussian to a 2D Gaussian
  - use the perspective projection matrix (known for the respective camera)
  - just an approximation of the true projection but works and is efficient
3D-2D SoG Similarity

**Problem:** Projection function ignores possible self-occlusions. Overlapping Gaussians might contribute several times to the energy function

**Solution:**
- limit total energy contribution from single image Gaussian
- approximation but handles occlusions
- allows to calculate analytic derivatives
- parallelizable:
  - GPU implementation
  - multi-processor system
Advantages

- robust
- fast
- no training data
- relatively uncontrolled setting:
  - markerless
  - no background-subtraction
  - handles occlusions
  - actors interacting with each other
- parallelizable
- applicable in real-time
Drawbacks

Limitations:

- constant color model fails on highly textured regions
- simple body model hinders tracking of twisting motions
- < 5 cameras – stuck in local minima; fail to recover from incorrect limb detections

Proposed solutions:

- more complex appearance models
- more sophisticated optimization
- detect the tracking errors and run a global optimizer for the misaligned limbs
Real-Time Human Pose Recognition in Parts from a Single Depth Image

In a nutshell:
- single depth image
- novel human body part representation
- data-driven, learn on lots of training data
- very efficient, 200 fps, real-time
- used in Kinect gaming platform
Method Overview

**Input:** single input depth image, indicating calibrated depth in the scene

**Output:** small set of 3D joint proposals
Method Overview

1. input video = a sequence of individual input depth images
2. object recognition approach: estimate the body part through a per-pixel classification
3. reproject the classification results to predict 3D joint positions
Cameras


Kinect camera [www.ubergizmo.com]

Depth imaging technology:
- structured light sensor
- calibrated depth in the scene
- color and texture invariant
- depth resolution precision of a few cm
- synthesize
Synthetic and Real Data

**Want:** realism and variety

Ranging through:

- body shape and size
- pose
- clothing
- crop
Data Acquisition

*It is not who has the best algorithm that wins. It is who has the most data.*

— Andrew Ng

- iterative process
  - MoCap
  - sampling the model
  - training classifier
  - testing accuracy of joint prediction

...to learn to generalize from 100 000 poses

- furthest neighbor clustering
Synthetic and Real Data

several base character models with random skinning of hair and clothing

Synthesis pipeline:

- CG to render depth and body part images from 3D meshes
- 15 base meshes ← retargetted to

Note: The synthesized data turned out to be much more challenging than the real one.
Body Part Labeling

- 31 localized part labels that **densely** cover the body
  - texture map that is retargetted to skin the characters during rendering
  - distinct parts for left and right side of the body
- classification problem
Depth Image Features

- simple depth comparison features
- normalized to be depth and 3D translation invariant
- no preprocessing, 3px accesses per feature

Weak signal on its own, but combined with decision forest allows to disambiguate between body parts.
Randomized Decision Forests

forest = ensemble of trees

Tree:

- **split nodes** (feature params, threshold)
- **leaf nodes**, containing learned distribution over labels
- different **paths** that might be taken for particular input

\[ (I, x) \]

\[ P_1(c) \]

\[ P_T(c) \]

- **ML**: avoid overfitting by training on \( \sim 100k \) examples
- parallelizable \( \forall \) px on GPU
Trained Decision Tree

- propose a set of splitting candidates
- partition the set of examples into left and right subset
- Shannon’s information theory – maximize the gain in information

- depth image patch binarized to foreground/background silhouette
- avg across all pixels that reached tree node
- thickness of edge ~ number of pixels
Proposed Joint Positions

Have: per-pixel info
Want: 3D skeletal joints
Problem: accumulating centers of prob mass disrupted by outliers
Solution:
  ▶ local mode-finding
  ▶ mean shift [Comaniciu 2002] with weighted Gaussian kernel
Body Part Density Estimate

\[ f_c(x) = \sum_{i=1}^{N} w_{ci} \exp \left( - \frac{\| x - x_i \|^2}{b_c} \right) \]

- \( x \) coord in 3D world space
- \( x_i \) reprojection of image pixel into world space
- \( N \) number of pixels
- \( b_c \) learned per-part bandwidth (a smoothing parameter of Gaussian kernel)
- \( w_{ci} \) takes into account the inferred body part probability at the pixel

- mean shift efficiently finds modes in density
- detected modes lie on the surface of the body
  - push modes back into the scene by a learned z-offset
Results

Comparison:

- oracular exact nearest neighbor search
  - whole-body model
  - chamfer matching
- beats the previous state-of-the-art [Ganapathi 2010]
  - time-of-flight camera
  - tracking the skeleton with temporal and kinematic info
- $360^\circ$ rotations and multiple people

Note: experiments obeyed good ML practices – held out the original MoCap poses from the training dataset

Comparison:

- oracular exact nearest neighbor search
  - whole-body model
  - chamfer matching
- beats the previous state-of-the-art [Ganapathi 2010]
  - time-of-flight camera
  - tracking the skeleton with temporal and kinematic info
- $360^\circ$ rotations and multiple people

Note: experiments obeyed good ML practices – held out the original MoCap poses from the training dataset
Advantages

- robust, error recovery
- **super real-time** = ultra-high speed (< 5ms / frame)
- generalize - body shapes, sizes, clothing, poses
- no initialization, no temporal or kinematic information
  - but can complement any tracking algorithm
- uncontrolled setting:
  - markerless
  - cluttered background
  - works with multiple actors
- employed in real-time (Kinect for Xbox)
Drawbacks

Limitations:

- very long training times (3 trees to depth 20 on a 1000 core cluster - one day)
- fails to generalize well to unusual unseen pose
- sometimes most likely body part incorrect

Proposed future ideas:

- train yet deeper trees; additional:
  - small run-time computational cost
  - large memory penalty
- more powerful depth image features, e.g.:
  - depth integrals over regions
  - curvature
  - local descriptors
- investigate the synthesis pipeline, in particular the generative model and part definitions
This Work

- geared towards the industry
- squeeze max perf
- best results
- speed
- extensive experiments
- lots of data
More From MRC

- forest classifiers
- efficient hardware implementations
- general activity human poses
- single image

Some publications:
- **Decision Forests** for Computer Vision and Medical Image Analysis [Criminisi 2013]
- **The Vitruvian Manifold: Inferring Dense Correspondences** for One-Shot Human Pose Estimation [Taylor 2012]
- **Efficient Human Pose Estimation from Single Depth Images** [Shotton 2012]
Summary

Both papers:

- fast, efficient
- robust
- error recovery

⇒ practical for real-world applications

Toolbox: Gaussians
Introduction

1. Sum of Gaussians
   - Kinematic Skeleton
   - Body Approximation
   - Objective Function
   - Initialization and Tracking
   - SoG Similarity
   - Conclusions

2. Body Part Recognition
   - Data
   - Body Part Labeling
   - Image Features
   - Decision Forests
   - Proposed Joint Positions

Summary

Thank you
References I

[Moeslund 2006] T. Moeslund, A. Hilton, and V. Krüger

A survey of advances in vision-based human motion capture and analysis
CVIU, 2006


Vision-based human motion analysis: An overview
CVIU, 108, 2007

[Sigal 2010] L. Sigal, A. Balan, and M. Black. Humaneva

Synchronized video and motion capture dataset and baseline algorithm for evaluation of articulated human motion
IJCV, 87:4-27, 2010


Tracking people with twists and exponential maps
In Proc. CVPR, 1998
References II

Optimization and filtering for human motion capture - a multi-layer framework
IJCV, 87:75-92, 2010

Performance capture from sparse multi-view video
SIGGRAPH, 2008

Motion capture using joint skeleton tracking and surface estimation
In CVPR, 2009
References III

   *High-quality passive facial performance capture using anchor frames*
   SIGGRAPH, 2011

   *Lightweight binocular facial performance capture under uncontrolled lighting*
   SIGGRAPH Asia, 2012

   *Real time motion capture using a single time-of-flight camera*
   In Proc. CVPR, 2010
References IV

[Criminisi 2013] A. Criminisi and J. Shotton

*Decision forests for computer vision and medical image analysis*
Springer, February 2013


*The Vitruvian manifold: inferring dense correspondences for one-shot human pose estimation*
in Proc. CVPR, IEEE, June 2012


*Efficient human pose estimation from single depth images*
in Trans. PAMI, IEEE, 2012
References V

[Comaniciu 2002] D. Comaniciu and P. Meer
Mean shift: A robust approach toward feature space analysis
IEEE Trans. PAMI, 2002