Computer Vision II - Lecture 14

Articulated Tracking I

08.07.2014

Bastian Leibe
RWTH Aachen
http://www.vision.rwth-aachen.de

leibe@vision.rwth-aachen.de

Outline of This Lecture

- Single-Object Tracking
- Bayesian Filtering
  - Kalman Filters, EKF
  - Particle Filters
- Multi-Object Tracking
  - Data association
  - MHT, (JPDAF, MCMCDA)
  - Network flow optimization
- Articulated Tracking
  - GP body pose estimation
  - (Model-based tracking, AAMs)
  - Pictorial Structures

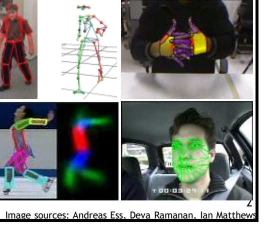
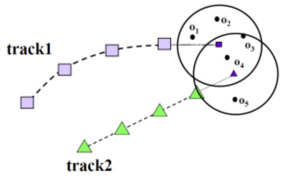


Image sources: Andreas Ess, Deva Ramanan, Ian Matthews

Recap: Linear Assignment Formulation

- Form a matrix of pairwise similarity scores
- Example: Similarity based on motion prediction
  - Predict motion for each trajectory and assign scores for each measurement based on inverse (Mahalanobis) distance, such that closer measurements get higher scores.



	ai1	ai2
1	3.0	
2	5.0	
3	6.0	1.0
4	9.0	8.0
5		3.0

  - Choose at most one match in each row and column to maximize sum of scores

Slide credit: Robert Collins

Recap: Linear Assignment Problem

- Formal definition
  - Maximize $\sum_{i=1}^{N}\sum_{j=1}^{M} w_{ij} z_{ij}$

    subject to
    $\sum_{j=1} z_{ij} = 1;\ i = 1, 2, \ldots, N$
    $\sum_{i=1} z_{ij} = 1;\ j = 1, 2, \ldots, M$
    $z_{ij} \in \{0, 1\}$

    Those constraints ensure that Z is a permutation matrix

  - The permutation matrix constraint ensures that we can only match up one object from each row and column.
  - Note: Alternatively, we can minimize cost rather than maximizing weights. $\arg\min_{z_{ij}} \sum_{i=1}^{N}\sum_{j=1}^{M} c_{ij} z_{ij}$

Slide adapted from Robert Collins

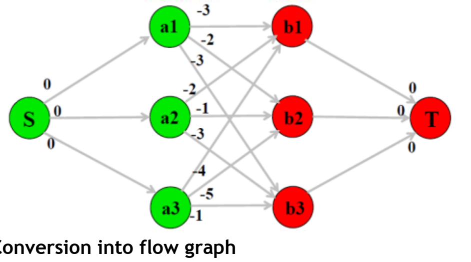
Recap: Optimal Solution

- Greedy Algorithm
  - Easy to program, quick to run, and yields "pretty good" solutions in practice.
  - But it often does not yield the optimal solution
- Hungarian Algorithm
  - There is an algorithm called Kuhn-Munkres or "Hungarian" algorithm specifically developed to efficiently solve the linear assignment problem.
  - Reduces assignment problem to bipartite graph matching.
  - When starting from an $N \times N$ matrix, it runs in $\mathcal{O}(N^3)$.

  ⇒ If you need LAP, you should use it.

Slide credit: Robert Collins

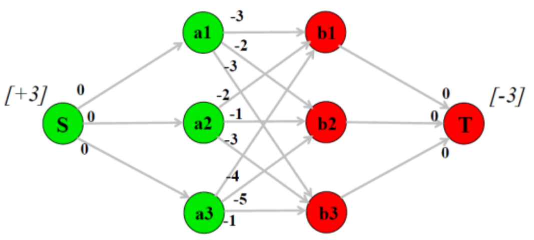
Recap: Min-Cost Flow



- Conversion into flow graph
  - Transform weights into costs $c_{ij} \quad \alpha \quad w_{ij}$
  - Add source/sink nodes with 0 cost.
  - Directed edges with a capacity of 1.

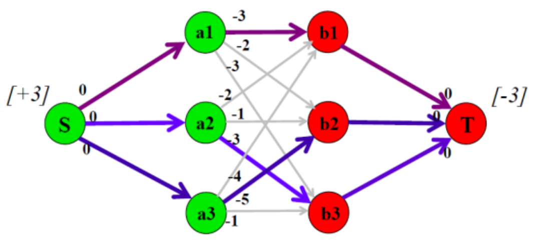
Slide credit: Robert Collins

Recap: Min-Cost Flow



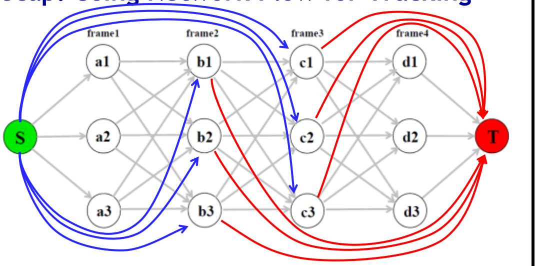
- Conversion into flow graph
  - Pump N units of flow from source to sink.
  - Internal nodes pass on flow ($\sum$ flow in $=\sum$ flow out).
  - $\Rightarrow$ Find the optimal paths along which to ship the flow.

Recap: Min-Cost Flow



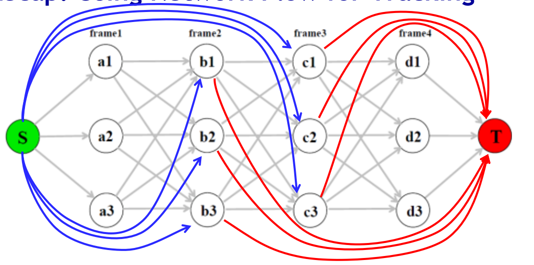
- Conversion into flow graph
  - Pump N units of flow from source to sink.
  - Internal nodes pass on flow ($\sum$ flow in $=\sum$ flow out).
  - $\Rightarrow$ Find the optimal paths along which to ship the flow.

Recap: Using Network Flow for Tracking



- Complication 1
  - Tracks can start later than frame1 (and end earlier than frame4)
  - $\Rightarrow$ Connect the source and sink nodes to all intermediate nodes.

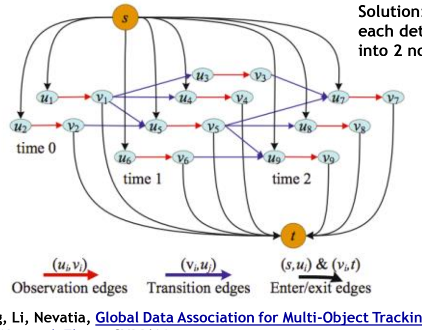
Recap: Using Network Flow for Tracking



- Complication 2
  - Trivial solution: zero cost flow!

Recap: Network Flow Approach

Solution: Divide each detection into 2 nodes



Zhang, Li, Nevatia, Global Data Association for Multi-Object Tracking using Network Flows, CVPR'08.

Recap: Min-Cost Formulation

- Objective Function

$$\mathcal{T}* = \underset{\mathcal{T}}{\operatorname{argmin}} \sum_i C_{in,i} f_{in,i} + \sum_i C_{i,out} f_{i,out}$$

$$+ \sum_{i,j} C_{i,j} f_{i,j} + \sum_i C_i f_i$$

- subject to
  - Flow conservation at all nodes

$$f_{in,i} + \sum_j f_{j,i} = f_i = f_{out,i} + \sum_j f_{i,j} \ \forall i$$

  - Edge capacities

$$f_i \leq 1$$

Topics of This Lecture

- Articulated Tracking
  - Motivation
  - Classes of Approaches
- Body Pose Estimation as High-Dimensional Regression
  - Representations
  - Training data generation
  - Latent variable space
  - Learning a mapping between pose and appearance
- Review: Gaussian Processes
  - Formulation
  - GP Prediction
  - Algorithm
- Applications
  - Articulated Tracking under Egomotion

Articulated Tracking

- Examples
  - Recover a person's body articulation
  - Track facial expressions
  - Track detailed hand motion
  - ...
- Common properties
  - Detailed parameterization in terms of joint locations or joint angles
  - Two steps
    - Pose estimation (in single frame)
    - Tracking (using dynamics model)
  - Challenging problem
    - High-dimensional
    - Hitting the limits of sensor data

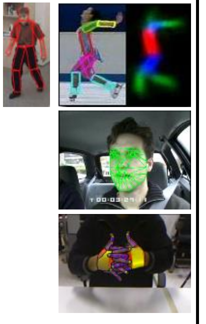
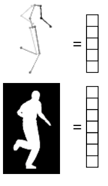


image sources: T. Svoboda, D. Ramanan, I. Matthews, J. Oikonomidis

Basic Classes of Approaches

- Global methods
  - Entire body configuration is treated as a point in some high-dimensional space.
  - Observations are also global feature vectors.
  - ⇒ View of pose estimation as a high-dimensional regression problem.
  - ⇒ Often in a subspace of "typical" motions...



- Part-based methods
  - Body configuration is modeled as an assembly of movable parts with kinematic constraints.
  - Local search for part configurations that provide a good explanation for the observed appearance under the kinematic constraints.
  - ⇒ View of pose estimation as probabilistic inference in a dynamic Graphical Model.

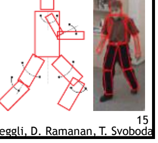
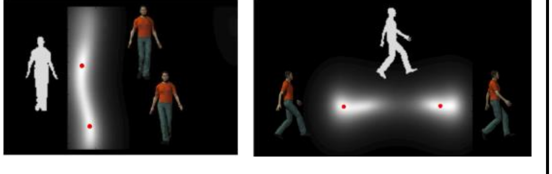


image sources: T. Jaeggli, D. Ramanan, T. Svoboda

Why Is It Difficult?



- Challenges
  - Poor imaging, motion blur, occlusions, etc.
  - Difficult to extract sufficiently good figure-ground information
  - Mapping is generally multi-modal: an image observation can represent more than one pose!

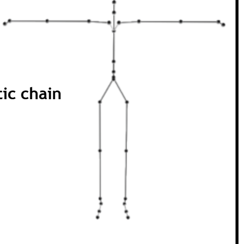
Slide credit: Raquel Urtasun

Topics of This Lecture

- Articulated Tracking
  - Motivation
  - Classes of Approaches
- Body Pose Estimation as High-Dimensional Regression
  - Representations
  - Training data generation
  - Latent variable space
  - Learning a mapping between pose and appearance
- Review: Gaussian Processes
  - Formulation
  - GP Prediction
  - Algorithm
- Applications
  - Articulated Tracking under Egomotion

Body Representation

- The body can be approximated as kinematic tree
- Parametrization via
  - Joint locations
  - Joint angles
  - Relative joint angles along kinematic chain
  - ...



- Example using in the following
  - 3D joint locations of 20 joints
  - ⇒ 60-dimensional space

Slide adapted from Raquel Urtasun

image source: R. Urtasun

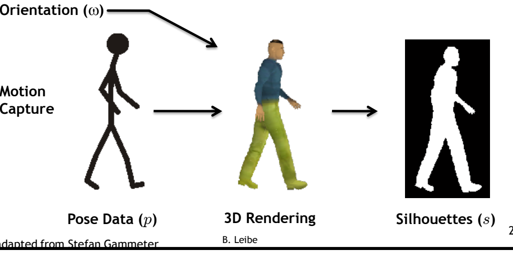
Image Representation

- Many possibilities...
- Popular choice: Silhouettes
  - Easy to extract using background modeling techniques.
  - Capture important information about body shape.
  - ⇒ We will use them as an example for today's lecture...

Video source: Hedvig Sidhenbladh

Another Advantage of Silhouette Data

- Synthetic training data generation possible!
  - Create sequences of „Pose + Silhouette“ pairs
  - Poses recorded with Mocap, used to animate 3D model
  - Silhouette via 3D rendering pipeline

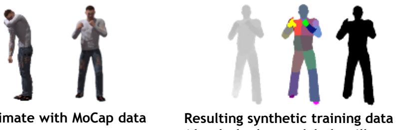


Slide adapted from Stefan Gammeter

Synthetic Training Data Generation

Varying body proportions

Different clothes models

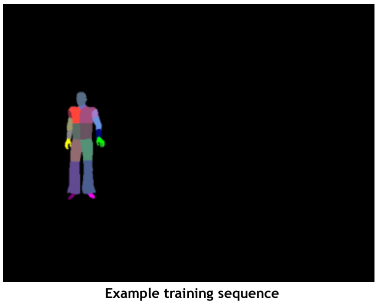


Animate with MoCap data

Resulting synthetic training data (depth, body part labels, silhouette)

Image source: Umer Rafi

Synthetic Training Data Generation

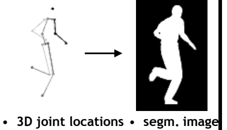


Example training sequence

Video source: Umer Rafi

Learning a Mapping b/w Pose and Appearance

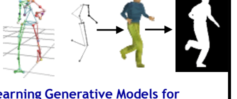
- Appearance prediction
  - Regression problem
  - High-dimensional data on both sides
  - ⇒ Low-dim. representation needed for learning!



- 3D joint locations
- ~60-dim.

- segm. image
- ~2500-dim.

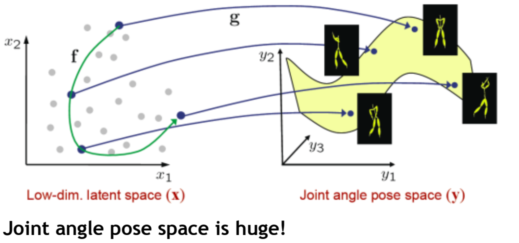
- Training with Motion-capture stimuli
  - Real dynamics from human actors
  - Synthesized silhouettes for training
  - Background subtraction for test



T. Jaeggli, E. Koller-Meier, L. Van Gool, "Learning Generative Models for Monocular Body Pose Estimation", ACCV 2007.

image source: T. Jaeggli

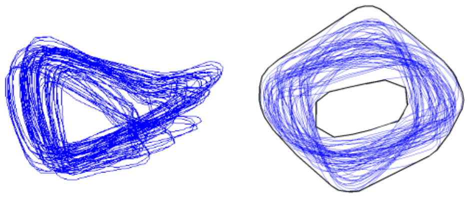
Latent Variable Models



- Joint angle pose space is huge!
  - Only a small portion contains valid body poses.
  - ⇒ Restrict estimation to the subspace of valid poses for the task
  - Latent variable models: PCA, FA, GPLVM, etc.

image source: R. Urtasun

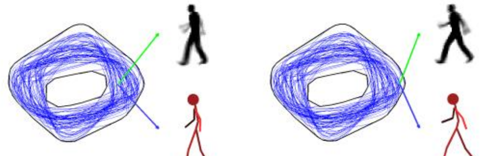
Example: Subspace of Walking Motion



- Pose modeling in a subspace
  - Pose model has 60 (highly dependent) DoF
  - But gait is cyclic, can be represented by a 2D latent space
  - Capture the dependency by dimensionality reduction (PCA, FA, CCA, LLE, GPLVM, ...)

image sources: S. Gammeter, T. Jaeggli

Articulated Motion in the Latent Space



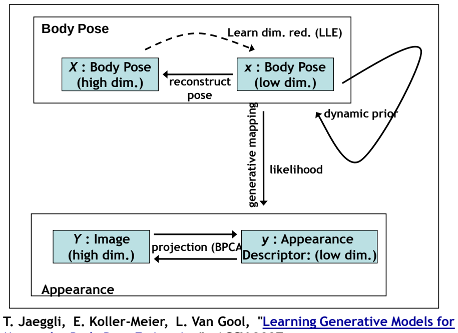
walking cycles have one main (periodic) DOF

additional DOF encode „walking style“

- Regression from latent space to
  - Pose $p(pose \mid \mathbf{z})$
  - Silhouette $p(silhouette \mid \mathbf{z})$
- Regressors need to be learned from training data.

Slide adapted from Stefan Gammeter

Learning a Generative Mapping

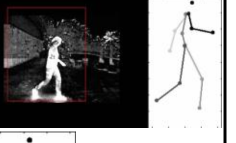


T. Jaeggli, E. Koller-Meier, L. Van Gool, "Learning Generative Models for Monocular Body Pose Estimation", ACCV 2007.

Slide credit: Tobias Jaeggli

Example Results

- Difficulties
  - Changing viewpoints
  - Low resolution (50 px)
  - Compression artifacts
  - Disturbing objects (umbrella, bag)

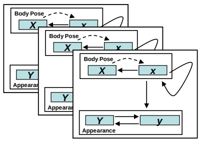


Original video

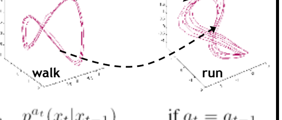
Video sources: Hedvig Sidhenbladh, Tobias Jaeggli

Representing Multiple Activities

- Learn multiple models
  - One model per activity
  - Separate LLE embedding
  - Separate dynamics



- Learn transition function
  - Link the LLE spaces
  - Find similar pose pairs
  - Learn smooth transition



$$p(x_t, a_t | x_{t-1}, a_{t-1}) \propto \begin{cases} p_{noswitch} & p^{a_t}(x_t|x_{t-1}) & \text{if } a_t = a_{t-1} \\ p_{switch} & p^{a_{t-1} \to a_t}(x_t|x_{t-1}) & \text{else} \end{cases}$$

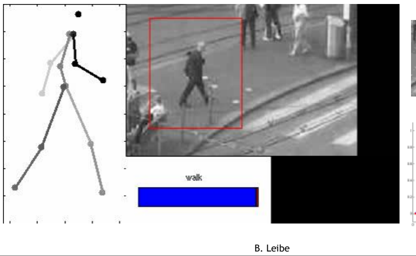
Slide credit: Tobias Jaeggli

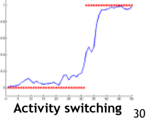
Switching b/w Multiple Activities

- Activity switching
  - Low-res. traffic scene
  - Transition from Walking to Running

Original video

Pose Estim. Input





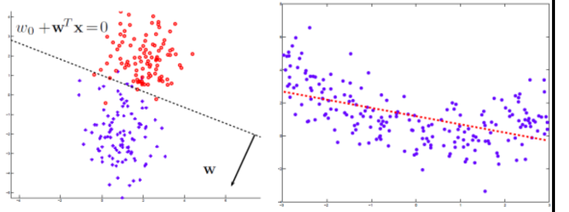
Activity switching

Videos by Tobias Jaeggli

Topics of This Lecture

- Articulated Tracking
  - Motivation
  - Classes of Approaches
- Body Pose Estimation as High-Dimensional Regression
  - Representations
  - Training data generation
  - Latent variable space
  - Learning a mapping between pose and appearance
- **Review: Gaussian Processes**
  - Formulation
  - GP Prediction
  - Algorithm
- Applications
  - Articulated Tracking under Egomotion

Classification vs. Regression

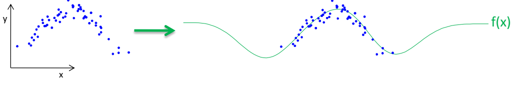


In classification: $y \in \{-1, 1\}$

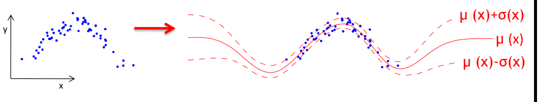
In regression: $y \in \mathbb{R}$

Gaussian Process Regression

- "Regular" regression: $y = f(\mathbf{x})$



- GP regression: $p(y|\mathbf{x}) \sim \mathcal{N}(\mu(\mathbf{x}), \sigma(\mathbf{x}))$



Gaussian Process Regression

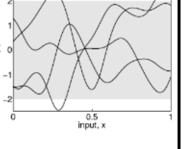
- GP Regression
  - Very easy to apply
  - Automatic confidence estimate of the result
  - Well-suited for pose regression tasks
- In the following, I will give a quick intro to GPs
  - Focus on main concepts and results
  - A far more detailed discussion will be given in the Advanced Machine Learning lecture (next semester).

Gaussian Process

- Gaussian distribution
  - Probability distribution over scalars / vectors.
- Gaussian process (generalization of Gaussian distrib.)
  - Describes properties of functions.
  - Function: Think of a function as a long vector where each entry specifies the function value $f(\mathbf{x}_i)$ at a particular point $\mathbf{x}_i$.
  - Issue: How to deal with infinite number of points?
    - If you ask only for properties of the function at a finite number of points...
    - Then inference in Gaussian Process gives you the same answer if you ignore the infinitely many other points.
- Definition
  - A Gaussian process (GP) is a collection of random variables any finite number of which has a joint Gaussian distribution.

Gaussian Process

- Example prior over functions $p(f)$
  - Represents our prior belief about functions before seeing any data.
  - Although specific functions don't have mean of zero, the mean of $f(\mathbf{x})$ values for any fixed $\mathbf{x}$ is zero (here).



  - Favors smooth functions
    - I.e. functions cannot vary too rapidly
    - Smoothness is induced by the covariance function of the Gaussian Process.
  - Learning in Gaussian processes
    - Is mainly defined by finding suitable properties of the covariance function.

Gaussian Process

- A Gaussian process is completely defined by
  - Mean function $m(\mathbf{x})$ and

$$m(\mathbf{x}) = \mathbb{E}[f(\mathbf{x})]$$

  - Covariance function $k(\mathbf{x},\mathbf{x}')$

$$k(\mathbf{x},\mathbf{x}') = \mathbb{E}[(f(\mathbf{x}) - m(\mathbf{x})(f(\mathbf{x}') - m(\mathbf{x}'))]$$

  - We write the Gaussian process (GP)

$$f(\mathbf{x}) \sim \mathcal{GP}(m(\mathbf{x}), k(\mathbf{x},\mathbf{x}'))$$

Gaussian Process: Squared Exponential

- Typical covariance function
  - Squared exponential (SE)
    - Covariance function specifies the covariance between pairs of random variables

$$\mathrm{cov}[f(\mathbf{x}_p), f(\mathbf{x}_q)] = k(\mathbf{x}_p, \mathbf{x}_q) = \exp\left\{-\frac{1}{2}|\mathbf{x}_p - \mathbf{x}_q|^2\right\}$$

- Remarks
  - Covariance between the outputs is written as a function between the inputs.
  - The squared exponential covariance function corresponds to a Bayesian linear regression model with an infinite number of basis functions.
  - For any positive definite covariance function $k(.,.)$, there exists a (possibly infinite) expansion in terms of basis functions.

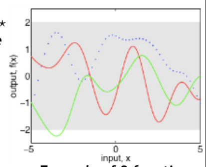
Gaussian Process: Prior over Functions

- Distribution over functions:
  - Specification of covariance function implies distribution over functions.
  - I.e. we can draw samples from the distribution of functions evaluated at a (finite) number of points.
  - Procedure
    - We choose a number of input points $X_\star$
    - We write the corresponding covariance matrix (e.g. using SE) element-wise:

$$K(X_\star, X_\star)$$

    - Then we generate a random Gaussian vector with this covariance matrix:

$$f_\star \sim \mathcal{N}(\mathbf{0}, K(X_\star, X_\star))$$



Example of 3 functions sampled

Image source: Rasmussen & Williams, 2006

GP Prediction with Noisy Observations

- Assume we have a set of observations: $\{(\mathbf{x}_n, f_n) \mid n = 1, \ldots, N\}$ with noise σ_n
- Joint distribution of the training outputs $\mathbf{f}$ and test outputs $\mathbf{f}_\star$ according to the prior:

$$\begin{bmatrix} \mathbf{t} \\ \mathbf{f}_\star \end{bmatrix} \sim \mathcal{N}\left(\mathbf{0}, \begin{bmatrix} K(X,X)+\sigma_n^2 I & K(X,X_\star) \\ K(X_\star,X) & K(X_\star,X_\star) \end{bmatrix}\right)$$

  - $K(X, X_\star)$ contains covariances for all pairs of training and test points.
- To get the posterior (after including the observations)
  - We need to restrict the above prior to contain only those functions which agree with the observed values.
  - Think of generating functions from the prior and rejecting those that disagree with the observations (obviously prohibitive).

Result: Prediction with Noisy Observations

- Calculation of posterior:
  - Corresponds to conditioning the joint Gaussian prior distribution on the observations:

$$\mathbf{f}_\star | X_\star, X, \mathbf{t} \sim \mathcal{N}(\bar{\mathbf{f}}_\star, \mathrm{cov}[\mathbf{f}_\star]) \qquad \bar{\mathbf{f}}_\star = \mathbb{E}[\mathbf{f}_\star | X, X_\star, \mathbf{t}]$$

  - with:

$$\bar{\mathbf{f}}_\star = K(X_\star, X)\left(K(X,X)+\sigma_n^2 I\right)^{-1}\mathbf{t}$$

$$\mathrm{cov}[\mathbf{f}_\star] = K(X_\star, X_\star) - K(X_\star, X)\left(K(X,X)+\sigma_n^2 I\right)^{-1} K(X, X_\star)$$

⇒ This is the key result that defines Gaussian process regression!
  - The predictive distribution is a Gaussian whose mean and variance depend on the test points $X_\star$ and on the kernel $k(\mathbf{x},\mathbf{x}')$, evaluated on the training data X.

GP Regression Algorithm

- Very simple algorithm

input: X (inputs), $\mathbf{y}$ (targets), k (covariance function), σ_n^2 (noise level), $\mathbf{x}_*$ (test input)

$L := \mathrm{cholesky}(K + \sigma_n^2 I)$
$\boldsymbol{\alpha} := L^\top \backslash (L \backslash \mathbf{y})$
$\bar{f}_* := \mathbf{k}_*^\top \boldsymbol{\alpha}$ } predictive mean eq. (2.25)
$\mathbf{v} := L \backslash \mathbf{k}_*$
$\mathbb{V}[f_*] := k(\mathbf{x}_*, \mathbf{x}_*) - \mathbf{v}^\top \mathbf{v}$ } predictive variance eq. (2.26)
$\log p(\mathbf{y}|X) := -\frac{1}{2}\mathbf{y}^\top \boldsymbol{\alpha} - \sum_i \log L_{ii} - \frac{n}{2}\log 2\pi$ eq. (2.30)
return: $\bar{f}_*$ (mean), $\mathbb{V}[f_*]$ (variance), $\log p(\mathbf{y}|X)$ (log marginal likelihood)

  - Based on the following equations (Matrix inv. ↔ Cholesky fact.)

$$\bar{f}_\star = \mathbf{k}_\star^T \left(K + \sigma_n^2 I\right)^{-1} \mathbf{t}$$

$$\mathrm{cov}[f_\star] = k(\mathbf{x}_\star, \mathbf{x}_\star) - \mathbf{k}_\star^T \left(K + \sigma_n^2 I\right)^{-1} \mathbf{k}_\star$$

$$\log p(\mathbf{t}|X) = -\frac{1}{2}\mathbf{t}^T \left(K + \sigma_n^2 I\right)^{-1} \mathbf{t} - \frac{1}{2}\log|K + \sigma_n^2 I| - \frac{N}{2}\log 2\pi$$

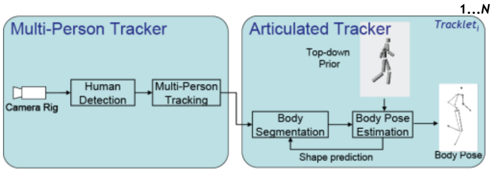
Image source: Rasmussen & Williams, 2006

Computational Complexity

- Complexity of GP model
  - Training effort: $\mathcal{O}(N^3)$ through matrix inversion
  - Test effort: $\mathcal{O}(N^2)$ through vector-matrix multiplication
- Complexity of basis function model
  - Training effort: $\mathcal{O}(M^3)$
  - Test effort: $\mathcal{O}(M^2)$
- Discussion
  - Exact GP methods become infeasible for large training sets.
  - ⇒ Need to use approximate techniques whenever #training examples > 2500-3000.

Topics of This Lecture

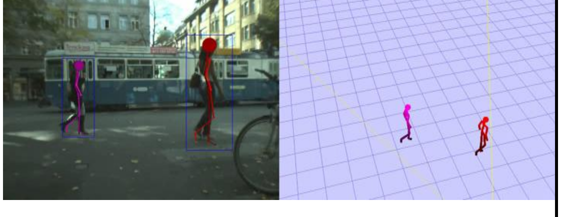
Articulated Multi-Person Tracking using GP



- Idea: Only perform articulated tracking where it's easy!
- Multi-person tracking
  - Solves hard data association problem
- Articulated tracking
  - Only on individual "tracklets" between occlusions
  - GP regression on full-body pose

[Gammeter, Ess, Jaeggli, Schindler, Leibe, Van Gool, ECCV'08]

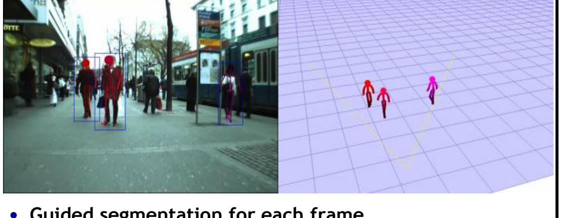
Articulated Multi-Person Tracking



- Multi-Person tracking
  - Recovers trajectories and solves data association
- Articulated Tracking
  - Estimates detailed body pose for each tracked person

[Gammeter, Ess, Jaeggli, Schindler, Leibe, Van Gool, ECCV'08]

Articulated Tracking under Egomotion



- Guided segmentation for each frame
  - No reliance on background modeling
  - Approach applicable to scenarios with moving camera
  - Feedback from body pose estimate to improve segmentation

[Gammeter, Ess, Jaeggli, Schindler, Leibe, Van Gool, ECCV'08]

Summary: Articulated Tracking with Global Models

- Pros:
  - View as regression problem (pose ↔ appearance)
  - Lots of machine learning techniques available
  - Research focus on handling the ambiguities
  - Training on MoCap data possible
    - Accurate models for human dynamics
- Cons:
  - High-dimensional problem
  - Global model
    - Can handle only those articulations it has previously seen
    - Not robust against partial occlusion
  - Difficult to get good appearance representation
    - MoCap data ⇒ Can synthesize silhouettes, but not appearance
    - Restricted to background subtraction