Skeleton Tracking

using range images

Norman Link

nlink@uni-koblenz.de

Institute for Computational Visualistics
Universität Koblenz-Landau

13. Dec. 2011

Overview

COMPUTERVISUALISTIK

Motiviation

- parent - infant experiments
- adult showing ”cup nesting“ task to infant
- detecting relations between parents and infants limb motions
- necessarity to detect skeleton topology for both parent and infant
- *OpenNI* is currently not capable of detecting infants as blobs of interest, thus skeleton tracking is not possible
- *NITE* skeleton tracking framework is closed source, modifying source code is not possible

COMPUTERVISUALISTIK

Motiviation

- currently tracking colored markers for infant skeleton detection
- using OpenNI for parent skeleton detection
- markerless tracking algorithm for both parent and infant desirable
- $\Rightarrow$ implementing skeleton tracking from scratch

COMPUTERVISUALISTIK

Framework

- simple module-based framework
- every module can define different implementations (controlled by factory pattern)
- implementations have to define the same interface
- user can define parameters from the outside before initialization
- module implementations easy exchangeable for testing purposes

COMPUTERVISUALISTIK

General Pipeline

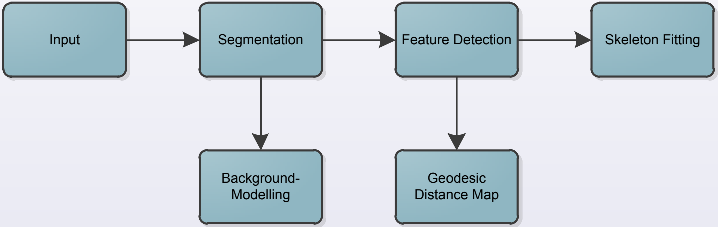


Figure: Pipeline

Input

- Retrieving input data
- Generating depth map and 3D map
- Reconstructing projection matrix and intrinsic parameters using correspondences from depth and 3D map. $\Rightarrow$ Mapping 3D points to 2D image plane.
- Saving image streams to disk

COMPUTERVISUALISTIK

Segmentation

- Depth map has to be separated into foreground and background objects
- Segmentation of foreground image into homogenious regions
- Finding relevant user blobs to analyze
- Simplest approach: assuming background to be constant. Taking first image as static background image, subtracting running image and detecting the biggest region as user blob.
- Later:
  - spatial clustering
  - detection and deletion of planar objects
  - detection and segmentation of moving objects

COMPUTERVISUALISTIK

Segmentation

Figure: Segmentation

COMPUTERVISUALISTIK

Feature Detection

- Detect outstanding feature points on the segmented foreground image
- important feature points:
  - Torso (center of gravity of user blob)
  - Head
  - Left & Right Hands
  - Left & Right Feet
- once every feature point has been found and labeled correctly, it is possible to fit a skeleton inside
- automatic labeling needs heuristics or training a classifier. Here: No implicit labeling, but trying to let skeleton tracking decide automatically which feature point corresponds to which joint (automatic labeling).

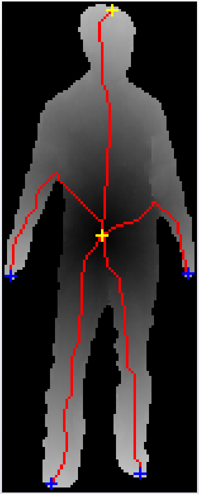
Geodesic Distance Map

General idea:

- 3d point cloud as a graph
- every graph node is connected with each of its 4 (cross) or 8 (full) neighbouring nodes
- edge weight = metric distance in 3D space
- compute the shortest distance from every graph node to the center of gravity
- shortest path algorithm: modified Dijkstra-Algorithm

COMPUTERVISUALISTIK

Geodesic Distance Map



Geodesic Distance Map

Advantages:

- features correspond to local maxima in the distance map (points farthest away from the center of gravity)
- feature distances almost invariant to pose
- taking user topology into account
- predecessor map allows for topology skeleton reconstruction, once all feature points have been detected

Disadvantages:

- necessarity to remove edges from the graph to avoid wrong geodesic paths
- local maxima detection subject to uncertanities

COMPUTERVISUALISTIK

Geodesic Distance Map

Approaches to local maxima detection

- non-maxima suppression: finding local maxima inside a mask by supressing non-maxima and emphasizing maxima
- finding maxima in the derivative by analizing the function and looking for zero crossings
- clustering approaches
- geodesic isolines / isopatches

COMPUTERVISUALISTIK

Geodesic Distance Map

Current feature detection:

- *Analogy: Consider the persons outline as a pond filled with water. Throw a small stone in the middle and find the regions that the wavefront reaches at last.*
- Undersample geodesic distance map
- respresent distances as iso patches, i.e. regions with approximately the same distance to the center of gravity (appropriate values: 10 - 30 cm)
- $\Rightarrow$ Local maxima detection: find patches that don't have neighbouring patches with a higher average distance

COMPUTERVISUALISTIK

Geodesic Distance Map

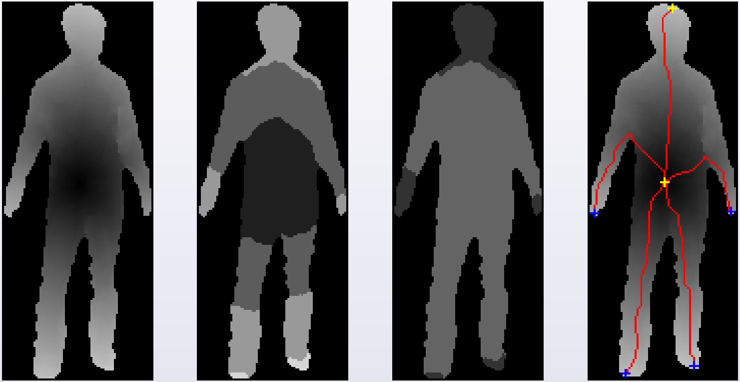
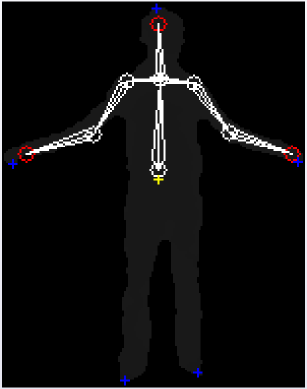


Figure: Feature-Detection

COMPUTERVISUALISTIK

Skeleton Fitting

- Goal: Matching a skeleton topology inside the users point cloud and detected features
- skeleton has to be inside the body



COMPUTERVISUALISTIK

Energy-Minimization

- Problem: Given a skeleton topology, find a joint configuration that best fits the users point cloud
- Upper body skeleton: 9 joints, connected with 8 bones.
- Every bone has 3 DOF (unconstrained)
- $\Rightarrow$ 24 + 3 DOF (XYZ-Position of root joint) $=$ 27 DOF
- Approximation by the use of energy minimization

COMPUTERVISUALISTIK

Energy-Minimization

- Every joint k defines an energy function E_k.
- Minimization of $E(P) = \sum_{k=0}^{n} E_k(P_k)$ with $P = \{P_1, P_2, ..., P_k\}$, P_k being the set of parameters that joint k relates upon.
- define E in such a way that it has one strong minimum and optimally no side minima (or only significantly weaker ones)

Energy-Minimization

- Finding the minimum by the means of local optimization (Newton method, Levenberg-Marquard algorithm, and more)
- Approximating the functions derivative at a starting point
- stepping ”downhill“ in the direction to the assumed minimum
- stopping, when the change in energy between two consecutive steps is below a threshold

Energy Function for End-Affectors

- distance to nearest feature point inside a search radius
- $\Rightarrow$ set this joint as *assigned* and assign the feature point with the label of this joint
- if there is no feature point nearby, use mean distance to k nearest neighbours from 3D point cloud
- $\Rightarrow$ set this joint as *unassigned*.

COMPUTERVISUALISTIK

Energy Function for Non-Affectors

- if the next joint in skeleton hierarchy is assigned, let this joint attract to the path from the assigned feature point to the center of gravity. $\Rightarrow$ distance from joint position to nearest point on geodesic path
- if the next joint is unassigned, use mean distance from k nearest neighbours from 3D point cloud

COMPUTERVISUALISTIK

Skeleton Constraints

- defining minimum and maximum parameter ranges for every joint in every DOF
- important for the algorithm
- restricting joint motions to only possible poses
- need to be defined accurately for every parameter dimension

COMPUTERVISUALISTIK

Demonstration

Demonstration

Current stage of development

Problems

- Tracking approach $\Rightarrow$ can loose tracking from time to time
- possibility to get stuck at local minima
- algorithm highly dependent on feature detection and labeling
- minimization algorithm computational complex

Future Prospects

- speed up minimization process
- improve energy functions for each joint
- improve automatic joint labeling for better assignments
- geodesic distance map needs to handle special cases more robustly
- better user segmentation

COMPUTERVISUALISTIK

- SHOTTON, Jamie ; FITZGIBBON, Andrew W. ; COOK, Mat ; SHARP, Toby ; FINOCCHIO, Mark ; MOORE, Richard ; KIPMAN, Alex ; BLAKE, Andrew:
  Real-time human pose recognition in parts from single depth images.
  In: *CVPR*, IEEE, 2011, 1297-1304
- SCHWARZ, Loren ; MKHYTARYAN, Artashes ; MATEUS, Diana ; NAVAB, Nassir:
  Estimating Human 3D Pose from Time-of-Flight Images Based on Geodesic Distances and Optical Flow.
  In: *IEEE Conference on Automatic Face and Gesture Recognition (FG)* (2011), March

COMPUTERVISUALISTIK