Real-time 3D Pose Estimation of Hundreds of Objects

Karl Pauwels
University of Granada, Spain

www.karlpauwels.com
www.youtube.com/user/karlpauwels
Objective

- 6DOF object pose
  - 3D position
  - 3D orientation
- Model-based
  - 3D geometry
  - appearance
Motivation and Strengths

• Motivation for **real-time** pose estimation
  – closed-loop control (e.g. visual servoing)
  – augmented reality
  – interactive exploration (speed-up discovery)

• Strengths of our approach
  – speed (>> real-time on discrete GPUs)
  – accuracy (precision)
  – robustness (noise and occlusions)
General Approach

- Continuous real-time interaction between visual simulation (GPU graphics) and visual perception (GPU compute)

- Object poses are updated using **dense** visual cues (requiring massive parallelism), and these poses are fed back to enable/facilitate the cue extraction itself
Low-level Vision
Low-level Vision

• Dense motion and stereo exploiting model feedback
• Lightweight and suitable for mobile
  – 3x optical flow and 1x dense stereo at 640x480
  – < 10ms using one GTX590 core
• SIFT keypoints
  – SiftGPU (http://cs.unc.edu/~ccwu/siftgpu/)
  – 50 ms on the other GTX590 core

6DOF Object Pose Estimation

- SIFT keypoints for pose detection
- Motion and depth cues for tracking
  - Optical flow
  - Augmented Reality flow
  - Stereo disparity (or Kinect depth)
- Jointly optimized
  - Structure-From-Motion for motion cues
  - Iterative Closest Point for depth cues

*Pauwels, K. et al. Real-time model-based rigid object pose estimation and tracking combining dense and sparse visual cues. IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2013*
Multi-object Performance

A  4 objects

B  36 objects

C  144 objects
Multi-object Performance

pre-processing (pre), absolute residuals and scale (scale), composition and reduction of the normal equations (normeq), solving the normal equations (solve) and rendering (render)

Table 2: Tracking frame rates (in Hz)

<table>
<thead>
<tr>
<th># objects</th>
<th>50,000</th>
<th>500,000</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>62</td>
<td>57</td>
</tr>
<tr>
<td>20</td>
<td>55</td>
<td>44</td>
</tr>
<tr>
<td>150</td>
<td>47</td>
<td>38</td>
</tr>
</tbody>
</table>
Multi-object Demo (Video)

object detection / 3D pose estimation

arbitrary view rendered with estimated 3D pose
Articulated Objects
Articulated Objects

**INPUT**
- left video
- right video

**CUES**
- stereo optical flow
- AR flow keypoints

**CON-STRANTS**
- type
- frame
- screw

**MODELS**
- 6D pose
- shape
- texture
- keypoints
Articulated Objects

- Different objects (parts) considered separate
- 6DOF pose update / part
- Post-impose hard constraints (Lagrange multipliers) on velocity updates, while minimizing increase in original problem’s least-square error
- Extended to include pose **detection** and to allow for occluded parts
- GPU-friendly (parts can be processed in parallel as before)

Articulated Objects
Kinematic Structure
Articulated Objects Demo (Video)
Articulated Box Folding (Video)
Incorporating the Robot

Incorporating the Robot

- **Sensor Data**: RGB, depth, encoder values
- **Visual Features**: SIFT, AR flow
- **Object Detection**: shape, texture, keypoints
- **Robot Model**: shape
- **Internal State**: base pose + offset, hand pose, grasped object pose, target object pose
- **Tracking**: position-based, endpoint closed-loop
- **Robot Control**: initialization + recovery

**Camera Motion**

**Target Object Motion**
Visual Servoing Demo (Video)
System Development

- Ubuntu 12.04 with QtCreator as IDE and CMake as build system
- CUDA
  - extensive use of textures and OpenGL interoperability
  - CUDPP for stream compaction
  - SiftGPU for feature extraction and matching
- OpenCV for camera/Kinect input and color conversion
- Eigen for linear system solving
- Matlab-prototype-driven development using small binaries with MAT-file I/O rather than MEX-files
  - stability (prevents Matlab crashes)
  - simplicity (with OpenGL)
  - IDE debugging, profiling, valgrinding, CUDA-memchecking, …
Acknowledgments

• University of Granada
  – Leonardo Rubio
  – Prof. Javier Diaz
  – Prof. Eduardo Ros

• Royal Institute of Technology, KTH, Stockholm
  – Alessandro Pieropan
  – Puren Guler
  – Prof. Danica Kragic

• University of Edinburgh
  – Vladimir Ivan
  – Peter Sandilands
  – Prof. Sethu Vijayakumar

• King’s College London
  – Emmanouil Evangelos
  – Dr. Ketao Zhang
  – Prof. Jian S Dai