3D Computer Vision: from Points to Concepts

Luís A. Alexandre

ICPRAM – January 12, 2015
Motivation
Games

Video game industry has an estimated market value in 2015 of 111 billion USD (Gartner).
Motivation

3D facial and expression recognition
Motivation

Robotic vision

DARPA Robotics Challenge

RobotCup@Home: Cosero
Online shopping

1. Create a MeModel of yourself in under 30 seconds
2. Try on clothes and create outfits from online stores
3. Buy clothes with confidence in how they look and fit
3D Sensors
Common technologies for getting 3D images

- Stereo vision: “home made” systems, Bumblebee2

Larger disparity (ALx-ARx), closer point (A)
Common technologies for getting 3D images

- Structured light: Microsoft Kinect, Asus Xtion Live, IDS Ensenso N20
Common technologies for getting 3D images

- Time-of-flight: Microsoft Kinect 2, DepthSense, Creative Senz3D, Intel RealSense F200, Fotonic, PMD CamBoard pico
Common technologies for getting 3D images

- Size and weight have been falling:
Point Clouds

- These sensors eventually produce a point cloud, typically at 30 fps:

- For 300k points with RGB at 30 fps: more than 30 MB/s.
Keypoints

Keypoints
Keypoints

What, why

- What are keypoints? A set of points considered representative of the point cloud.
- Why? Too much data for real-time processing.
- Keypoints are a way to do sub-sampling, hopefully an intelligent one!
Humans don’t process every “input pixel”, but focus their attention on salient points.

We have proposed recently a 3D keypoint detector based on a computational model of the HVS: BIK-BUS (Biologically Inspired 3D Keypoint based on Bottom-Up Saliency).
BIK-BUS: some HVS mechanisms

- **Center-surround cells**: sensitive to the center of their receptive fields and are inhibited by stimuli in its surroundings.

- **Color double-opponency**: neurons are excited in the center of their receptive field by one color and inhibited by the opponent color (red-green or blue-yellow) while the opposite takes place in the surround.

- Impulse response of **orientation-selective neurons** is approximated by Gabor filters.

- **Lateral inhibition**: neighboring cells inhibit each other through lateral connections.
BIK-BUS

- Performance evaluated against 8 state-of-the-art detectors.
- We performed around 1.6 million comparisons for each pair keypoint detector/descriptor for a total of 135 pairs (9 keypoint detectors \( \times \) 15 descriptors).

**Counting the number of times a keypoint detector has the best result in Table III in case of a tie both methods score.**

<table>
<thead>
<tr>
<th>Keypoint</th>
<th>Category AUC</th>
<th>Category DEC</th>
<th>Object AUC</th>
<th>Object DEC</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>BIK-BUS</td>
<td>7</td>
<td>9</td>
<td>7</td>
<td>9</td>
<td>32</td>
</tr>
<tr>
<td>Curvature</td>
<td>3</td>
<td>2</td>
<td>2</td>
<td>1</td>
<td>7</td>
</tr>
<tr>
<td>Harris3D</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>2</td>
</tr>
<tr>
<td>ISS3D</td>
<td>2</td>
<td>0</td>
<td>4</td>
<td>2</td>
<td>8</td>
</tr>
<tr>
<td>KLT</td>
<td>2</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>4</td>
</tr>
<tr>
<td>Lowe</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>Noble</td>
<td>0</td>
<td>1</td>
<td>2</td>
<td>1</td>
<td>4</td>
</tr>
<tr>
<td>SIFT3D</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>SUSAN</td>
<td>2</td>
<td>1</td>
<td>2</td>
<td>1</td>
<td>6</td>
</tr>
</tbody>
</table>
Descriptors
Descriptors

- What is a descriptor? Measure extracted from the input data that represents or describes an input data region in a concise manner.
- They are used to discard most input data and keep only a representation (are the “features”).
- There is a wide choice of descriptors: which should one use?
- We made an evaluation of 13 available in PCL.
Descriptors

Recall vs. 1-Precision for various descriptors:
- PFHRGB
- PFH
- FPFH
- PPF
- VFH
- ESF
- CVFH
- RIFT
- PCE
- USC
- 3DSC
- SHOTCOLOR
- SHOT
Accurate descriptors are very computationally intensive.

Faster descriptors use quite a bit of storing space.

We developed a trainable descriptor that is both fast and has a small space footprint, while maintaining an acceptable accuracy.
Genetic Algorithm-Evolved 3D Point Cloud Descriptor

- Create a keypoint cloud by sub-sampling with leaf size of 2cm.
- Consider 2 regions: disk ($R_1$) + ring ($R_2 - R_1$)

- Shape: histogram of angles between normals at keypoint and at each neighbor in region.
- Color: (Hue, Saturation) histogram of points in each region.
Genetic Algorithm-Evolved 3D Point Cloud Descriptor

- Distance between 2 point clouds: \( d = w \cdot d_{shape} + (1 - w) \cdot d_{color} \).
- Genetic algorithm: 5 parameters (\#shape bins, \#color bins, \( R_1 \), \( R_2 \), \( w \))

<table>
<thead>
<tr>
<th>Descriptor</th>
<th>Object (%)</th>
<th>Category (%)</th>
<th>Time (s)</th>
<th>Size</th>
</tr>
</thead>
<tbody>
<tr>
<td>PFHRGB</td>
<td>20.25</td>
<td>5.27</td>
<td>2992</td>
<td>250</td>
</tr>
<tr>
<td>SHOTCOLOR</td>
<td>26.58</td>
<td>9.28</td>
<td>178</td>
<td>1353</td>
</tr>
<tr>
<td>Our</td>
<td>27.43</td>
<td>10.34</td>
<td>72</td>
<td>248</td>
</tr>
</tbody>
</table>
3D Object Recognition
Typical 3D object recognition pipeline
Point cloud matching

- Each point cloud is represented by a set of descriptors.
- Each descriptor is $n$-dimensional.
- Variable number of descriptors in each set.
- To find the closest object DB match to the input point cloud we need a set distance.
Set distances

- Set distances are usually built around point distances.
- Three common point distances $x, y \in \mathbb{R}^n$:
  - City-block:
    \[ L_1(x, y) = \|x - y\|_1 = \sum_{i=1}^{n} |x(i) - y(i)| \]
  - Euclidian:
    \[ L_2(x, y) = \|x - y\|_2 = \sqrt{\sum_{i=1}^{n} (x(i) - y(i))^2} \]
  - Chi-squared:
    \[ d_{\chi^2}(x, y) = \frac{1}{2} \sum_{i=1}^{n} \frac{(x(i) - y(i))^2}{x(i) + y(i)} \]
Set distances

- $a, b$ are points; $A, B$ are sets.
- $D_1(A, B) = \max\{\sup\{f(a, B) \mid a \in A\}, \sup\{f(b, A) \mid b \in B\}\}$ with $f(a, B) = \inf\{L_1(a, b), \ b \in B\}$
- $D_2 = \text{Pyramid Match Kernel distance}$
- $D_3(A, B) = L_1(\min_A, \min_B) + L_1(\max_A, \max_B)$ with
  \[
  \min_A(i) = \min_{j=1, \ldots, |A|}\{a_j(i)\}, \ i = 1, \ldots, n \\
  \max_A(i) = \max_{j=1, \ldots, |A|}\{a_j(i)\}, \ i = 1, \ldots, n
  \]
  and similarly for $\min_B(i)$ e $\max_B(i)$.
- $D_4(A, B) = L_1(c_A, c_B)$ where $c_A, c_B$ are cloud centroids
- $D_5(A, B) = L_2(c_A, c_B)$
- $D_6(A, B) = D_4(A, B) + L_1(\text{std}_A, \text{std}_B)$ with
  \[
  \text{std}_A(i) = \sqrt{\frac{1}{|A|-1} \sum_{j=1}^{|A|}(a_j(i) - c_A(i))^2}, \ i = 1, \ldots, n
  \]
  and similarly for $\text{std}_B$.
- $D_7(A, B) = d_{\chi^2}(c_A, c_B) + d_{\chi^2}(\text{std}_A, \text{std}_B)$
- $D_8(A, B) = \frac{1}{|A||B|} \sum_{i=1}^{|A|} \sum_{j=1}^{|B|} L_1(a_i, b_j)$
Distance evaluation

- We evaluated 8 distances using 2 descriptors (PFHRBG and SHOTCOLOR).
- Data set with 48 objects from 10 categories and 1421 point clouds.
- Keypoint detector used: Harris3D.
Distance evaluation using PFHRGB

![Graph showing distance evaluation using PFHRGB]

Luís A. Alexandre (UBI and IT) ICPRAM 2015 31 / 43
3D Object recognition

Distance evaluation using SHOTCOLOR

Recall vs. 1-Precision for different datasets (D1 to D8). The graph shows the performance of the SHOTCOLOR method in recognizing 3D objects, with each dataset represented by a different marker and color.

Luís A. Alexandre (UBI and IT) ICPRAM 2015 32 / 43
Distance evaluation

- Time in seconds for test set evaluation (12 threads).

<table>
<thead>
<tr>
<th>Distance</th>
<th>PFHRGB</th>
<th>SHOTCOLOR</th>
</tr>
</thead>
<tbody>
<tr>
<td>$D_1$</td>
<td>1914</td>
<td>175</td>
</tr>
<tr>
<td>$D_2$</td>
<td>2197</td>
<td>1510</td>
</tr>
<tr>
<td>$D_3$</td>
<td>1889</td>
<td>132</td>
</tr>
<tr>
<td>$D_4$</td>
<td>1876</td>
<td>137</td>
</tr>
<tr>
<td>$D_5$</td>
<td>1886</td>
<td>134</td>
</tr>
<tr>
<td>$D_6$</td>
<td>1885</td>
<td>132</td>
</tr>
<tr>
<td>$D_7$</td>
<td>1883</td>
<td>113</td>
</tr>
<tr>
<td>$D_8$</td>
<td>1914</td>
<td>174</td>
</tr>
</tbody>
</table>

- Simple distances like $D_6$ and $D_7$ are a good choice (accurate and fast) better than more common distances such as $D_1$ and $D_2$.
- Additionally, simple distances don’t need any parameter choosing.
Object recognition system
Deep learning is showing great potential in pattern recognition.

The idea of transfer learning is also a very appealing one: learn in one problem and reuse (at least part of) the knowledge in other problems.

Use both in a work where a convolutional neural network learns to recognize objects from 3D data.

Transfer learning is used from one color channel to the others and also to the depth channel.

Decision fusion is used to merge each nets predictions'.
3D Object recognition

Deep transfer learning for 3D object recognition

- Average error and time used on 10 repetitions.

<table>
<thead>
<tr>
<th>Approach</th>
<th>Error [%]</th>
<th>Time[s]</th>
</tr>
</thead>
<tbody>
<tr>
<td>RGBD</td>
<td>29.87</td>
<td>714.60</td>
</tr>
<tr>
<td>Channel R</td>
<td>32.15</td>
<td>136.50</td>
</tr>
<tr>
<td>Channel G</td>
<td>44.02</td>
<td>131.60</td>
</tr>
<tr>
<td>Channel B</td>
<td>55.62</td>
<td>110.10</td>
</tr>
<tr>
<td>Channel D</td>
<td>65.85</td>
<td>126.30</td>
</tr>
<tr>
<td>R,G,B,D maj</td>
<td>36.72</td>
<td>504.50</td>
</tr>
<tr>
<td>R,G,B,D mean</td>
<td>29.58</td>
<td>504.50</td>
</tr>
<tr>
<td>Channel G + TL</td>
<td>37.47</td>
<td>166.60</td>
</tr>
<tr>
<td>Channel B + TL</td>
<td>43.58</td>
<td>95.10</td>
</tr>
<tr>
<td>Channel D + TL</td>
<td>66.32</td>
<td>157.70</td>
</tr>
<tr>
<td>R,G+TL,B+TL,D+TL maj</td>
<td>33.45</td>
<td>555.90</td>
</tr>
<tr>
<td>R,G+TL,B+TL,D+TL mean</td>
<td><strong>28.80</strong></td>
<td>555.90</td>
</tr>
<tr>
<td>R,G+TL,B+TL,D maj</td>
<td>32.63</td>
<td>524.50</td>
</tr>
<tr>
<td>R,G+TL,B+TL,D mean</td>
<td>29.01</td>
<td>524.50</td>
</tr>
</tbody>
</table>
3D Object Tracking
Why track?

- The world is dynamic: another step to understanding it is to follow objects as they move.
- Many different ways to track: most used are particle filter variants.
PFBIK 3D tracking

- We used a biologically-inspired keypoint extractor to initialize and maintain particles for particle filter-based tracking from 3D.
PFBIK tracking results

- We compared our tracker against the OpenNI tracker available in PCL.
- 10 different moving objects, 3300 point clouds.

<table>
<thead>
<tr>
<th></th>
<th>Number of Points</th>
<th>Tracking</th>
<th>Distance between Centroids</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Keypoints</td>
<td>Initialization</td>
<td>Tracking</td>
</tr>
<tr>
<td>PFBIK-Tracking</td>
<td>116.973 ± 76.251</td>
<td>0.203 ± 0.162</td>
<td>0.036 ± 0.032</td>
</tr>
<tr>
<td>OpenniTracker</td>
<td>2502.536 ± 1325.807</td>
<td>0.173 ± 0.187</td>
<td>0.038 ± 0.023</td>
</tr>
</tbody>
</table>
Challenges
Challenges

- **object representation** set of descriptors of partial object views (2.5D) but other possibilities might be better (represent an object using a fused view representation).

- **non-rigid object recognition** current keypoint + descriptor approach is not a good solution: more complex models are needed.

- **activity recognition** what are the best approaches to understand human activities from 3D video?

- **real-time processing** GPU-based implementations of most algorithms can bring us here but still a problem with embedded devices (cloud-based processing requires high bandwidth and permanent connection).
Acknowledgments

▶ Students:

Sílvio Filipe

Dominik Wegrzyn

▶ Colleague: Laurent Itti

▶ Funding:

▸ Student grant: FCT grant SFRH/BD/72575/2010
▸ Project: PTDC/EIA-EIA/119004/2010
▸ Institutions: UBI and IT (PEst-OE/EEI/LA0008/2013)

Thank you for your time!

Luís A. Alexandre (UBI and IT)