Realtime Object Pose Estimation and Detection for Manipulation

Zhe Cao, Natasha Kholgade , Yaser Sheikh

THE ROBOTICS INSTITUTE

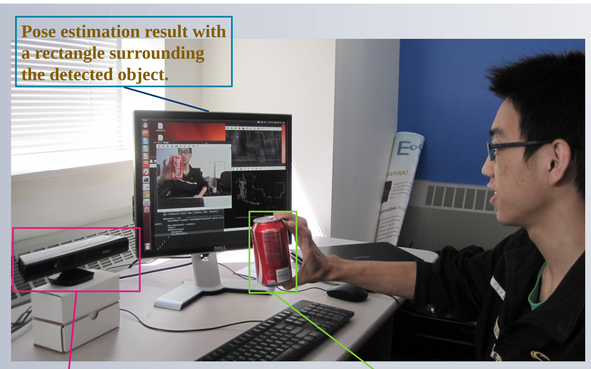
Introduction

Real-time recognition of the precise pose of an object is essential for a robotic system to perform complex manipulation tasks. In this work, we use a 3D model to render example poses of an object, and find the nearest match to the input image using a GPU implementation. Our goal is to perform efficient matching, we need to address changes in illumination and appearance across an object.

Pictures of HERB from CMU Personal Robotic Lab

Real-time System

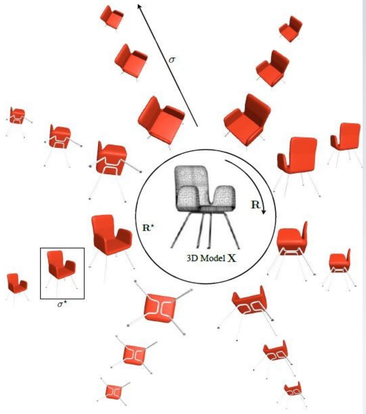
We build a system to perform real-time estimation of the viewpoint, scale, and translation of an object from a real-time camera.



Imagine that I am a robot, the Kinect is my eye and the computer is my brain. At first I use the Kinect to percept both the depth information and the color image. Computer programs in my brain can help me find the position and the angle of the object. This enables me to do manipulation in real-time.

METHODS

We achieve invariance to illumination and appearance by matching in the Laplacian of Gaussian space and by using normalized cross-correlation.



Step 1: 3D Model Rendering

We pre-render the 3D model from several viewpoints that span the space of 288 poses and 10 scales, like the right image.

Step 2: Laplacian of Gaussian

Laplacian filters are derivative filters used to find areas of rapid change in images. Since they are very sensitive to noise, Gaussian filter is used to smooth the image before applying the Laplacian. Using edge information to do matching can solve the problem about changes in illumination and texture.

Step 3: Normalized cross correlation

Normalized cross correlation is used widely in object detection area. We run normalized cross correlation in different scales and poses.

The normalized cross correlation equation:

$$\frac{\sum_{x,y}[f(x,y)-\bar{f}_{u,v}][t(x-u,y-v)-\bar{t}]}{\left\{\sum_{x,y}[f(x,y)-\bar{f}_{u,v}]^2\sum_{x,y}[t(x-u,y-v)-\bar{t}]^2\right\}^{0.5}}$$

- where f represents the image, t represents the template.

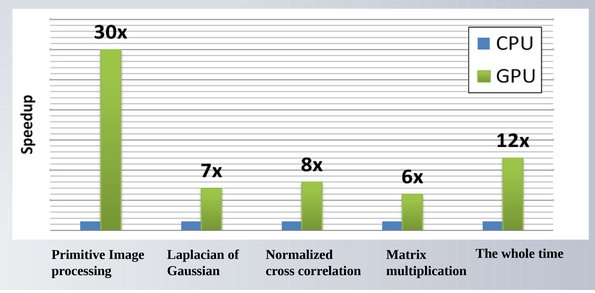
GPU Implementation

Performing normalized cross-correlation based matching of rendered poses for 288 views and 10 scales in real-time is computationally prohibitive. To address this, we parallelized the matching using CUDA programming with 2880 cores in a Nvidia Quadro K6000 GPU .

Quadro K6000

RESULTS

GPU and CPU runtime comparison



In our experiments, the method can deal with different scale and poses with low latency. We achieve 12X speed-up over a CPU. For example, when dealing with a 640*480 image and a 360*360 template, our program uses 35 ms to do 6 normalized cross correlation.

Matching results

The model rendering images and real-time pose estimation

REFERENCES

1. Natasha K et al, 3D Object Manipulation in a Single Image using Stock 3D Models.. ACM Transactions on Graphics (TOG) 2014

2. Lewis J P. Fast normalized cross-correlation[C]//Vision interface. 1995, 10(1): 120-123.

3. Collet A, et al. The MOPED framework: Object recognition and pose estimation for manipulation[J].