AAAI Technical Report SS-12-06
Wisdom of the Crowd

Kinect@Home: Crowdsourcing a Large 3D Dataset of Real Environments

Alper Aydemir, Daniel Henell and **Patric Jensfet**
CVAP, KTH, Stockholm, Sweden
aydemir, dhenell,patric@kth.se

Roy Shilkrot
Fluid Interfaces Group, Media Lab, MIT
roy.shil@gmail.com

Abstract

We present Kinect@Home, aimed at collecting a vast RGB-D dataset from real everyday living spaces. This dataset is planned to be the largest real world image collection of everyday environments to date, making use of the availability of a widely adopted robotics sensor which is also in the homes of millions of users, the Microsoft Kinect camera.

Introduction

Robotics has a long-standing aim to build robots that can function in complex man-made environments. The long term vision (which is rapidly becoming a short term goal) of robotics is to help humans with tedious and hard tasks, e.g. assisting elderly in everyday tasks, providing care for disabled persons for increased ability or performing hard, hazardous and tedious tasks that are unfit for human health.

In order to determine and accomplish such tasks, the robotics researcher usually *guesses* the tasks needed or the environments used by a typical user of such robots in the real world and tries to come up with various problems and solutions regarding perception, action and planning in robotics. The proposed solutions generally lacks the basis for the robustness as they are not tested in complex real environments with the intended end user. This leads a mismatch between what is promised in publications and their actual performance which is a growing concern as the pressure on robotics as a field to provide working products increases. For this reason, we present the Kinect@Home project.

Kinect@Home

The Kinect@Home project is aimed at collecting a vast dataset of Microsoft Kinect images of real everyday living spaces such as offices, homes and alike. The project location is at http://www.kinectathome.com. We have chosen the Microsoft Kinect camera because it provides both an RGB image and a depth value for each pixel of the image. Thanks to its high quality 3D data for its low price, the Kinect camera has been rapidly adopted as a robotics sensor. Most importantly, it has since entered the homes of some 20 million users therefore fit for a crowdsourcing task. The significance of this is being, never before a highly used robotics sensor was at the home of millions of people, therefore it presents ample opportunity for a crowdsourcing application.

Datasets in computer vision and robotics are widely used for testing and benchmarking various algorithms such as object recognition and detection, mapping and image segmentation. Already there exists several Kinect datasets (Lai 2011, Sun 2010, Silberman and Fergus 2011) mainly on the topic of object recognition and detection in scenes. We welcome these efforts and find them very encouraging. Closest to our approach is (Janoch et al.) where individual images of indoor scenes are being collected. However none of these datasets aims to capture the challenging real world scenes that a robot shipped to a home today might face. We believe we can make a big impact by collecting a large dataset of real world environments for developing better methods.

In robotics, various research groups have opted to recreate the man-made environments that these robots are intended to work in by building mock versions of living spaces such as kitchens and living rooms in their laboratories. These environments certainly serve as an initial testbed for algorithms and methods as a way of validating the plausibility of the proposed approach. However, there are several shortcomings regarding evaluating robot performances in simulated of living spaces. First, since only a few instances of the said home environments can be built, the evaluation of the proposed methods tends to include only a few cases of a general problem. Second, the environments tend not to be realistic and instead become over simplified, as no human lives and uses these spaces on a daily basis. We therefore propose the Kinect@Home project as a way to collect large amounts of 3D data from ordinary people's everyday environments. With this project, we will amass a large dataset of everyday indoor environments such as offices, kitchens, living room spaces. This data will be used for various applications such as object detection, recognition, 3D mapping and various other robotic applications. The dataset will be available publicly at the interest of all interested researchers.

In order to construct such a dataset, the software implementation should have certain specifications. We will continue by briefly describing our software architecture.

Software architecture and usage

The software architecture consists of two parts: clients which are ordinary people uploading Kinect frames and the server which collects the uploaded data. There are several considerations for building the software implementation that realizes the dataset. First of all, we want to minimize the number of steps a user has to take in order to accomplish the task. Therefore we avoid hefty downloads, installation guides or tedious tutorials. This means we cannot simply ask the user to download and install a program, record the Kinect frames to file (which would take a few gigabytes of data) and send over to us.

We have chosen a browser plug-in as the client since it provides a much more light-weight installation compared to a stand alone program both technically and in the minds of regular internet user. Furthermore by doing this the user interface will be HTML-based and by default cross platform. The plug-in is programmed using the FireBreath cross platform browser plugin framework (FireBreath 2012). The OpenNI framework is chosen to read frames from the device. Microsoft Kinect SDK (Microsoft 2012) is also considered however it only works on Windows.

We want the server to be as simple as possible and general enough to accept any type of client that may be realized in the future. Furthermore, the bandwidth and heavy hard disk file operations involving receiving large amounts of images need to be considered. For this reason, we have opted for an HTTP RESTful API using the Django web framework. We have considered frameworks such as ZeroMQ, Apache thrift, (Arslan 2012, Zeromq 2012). We will skip over the detailed discussion for the lack of space in this extended abstract, however they all seemed to need a significant amount of infrastructure, front-end code and a complete user-interface. Instead, HTTP REST calls are a fairly basic and almost ubiquitous standard used throughout the internet.

The raw Kinect data is too big to be uploaded without compression, we assume the typical user would not wait for the whole upload period. Therefore we compress the data stream with near-lossless video encoding. We compress and upload the data in chunks. This way the amount of HTTP calls and computational overhead is reduced compared to uploading every frame individually.

Upon reaching the website, the user will be prompted to connect their Kinect devices and install the plug-in. Once this is done, the website starts showing the live Kinect images on the browser as a confirmation that the software is working accordingly. This also helps to display the user the currently captured data. A *Record* button and an optional email adress text box is also displayed the purpose of which we will explain in more detail. Once the button is pressed, the plug-in starts uploading captured frames to server. After a set period of time or when the user hits the *Stop* button, the recording stops and the user is prompted with an optional text box for metadata about the video. A progress bar indicates how much of the data is sent to the server.

Privacy and control of the data

In order to alleviate any user trust and user related problems we give full control to the data uploader. If the user provides an email address, we email the participant with a PIN code after each recording and the unique identification number of the specific upload. With these credentials, the user can view or delete the uploaded files anytime, with no questions asked. Our code base is entirely open source. As part of addressing the privacy concerns, we don't keep any user-related data whatsoever. The users however need to agree a terms of service agreement, which basically states that the data uploaded will be used for scientific purposes.

Conclusion

We have presented a crowdsourcing platform for collecting Kinect camera images. We will share our findings about the software architecture and the wider public's reactions in the coming months during the symposium. The system is open source and the data will be completely anonymous and publicly available. We expect a high participation.

Acknowledgements

The authors thank Javier Romero and Burak Arslan for their suggestions.

References

Arslan, B. 2012. Remote procedure call library.

Sun, M. S. et al. 2010. Depth-encoded hough voting for joint object detection and shape recovery. In *ECCV*.

Lai, Keven, et al. 2011. A large-scale hierarchical multi-view rgb-d object dataset. In *ICRA*, 1817–1824.

FireBreath. 2012. Firebreath, cross-platform browser plugin framework.

Janoch, A., et al. A category-level 3-d object dataset: Putting the kinect to work. In *ICCV Workshop on Consumer Depth Cameras for Computer Vision*.

Microsoft. 2012. Microsoft kinect sdk.

Silberman, N., and Fergus, R. 2011. Indoor scene segmentation using a structured light sensor. In *ICCV Workshop on 3D Representation and Recognition*.

ZeroMQ. 2012. Zeromq: The intelligent transport layer.