3D Object Recognition and Neural Networks

Literature Review #2 - COMP.5460

Adam Gaudreau

21 March 2018

3D OBJECT RECOGNITION

When we think of computer graphics, we typically think of them as a representation of numerical data, a user interface, a photo or image, a video game, or other image processing and rendering applications. However, graphics are important to computers themselves, especially for image and object recognition. These image recognition methods are fundamental for machine learning models such as neural networks, since a computer needs to be able to understand what it's ”looking at”.

As technology progresses, there is a greater demand for more advanced object recognition methods—more specifically—for three-dimensional objects. (By advanced I mean more powerful, fast, and efficient.) This type of recognition would have a vast array of applications, including but not limited to military use, Google searches, artificial intelligence, self-driving cars, and much more. There are a handful of existing methods out there, but they differ in their approach and consequently their efficiency. Since the research field trivially strives to find more efficient solutions, there is a current focus on real-time object recognition.

What methods are used today to recognize 3D objects you ask? Machine/Deep learning algorithms, mostly. There are a plethora of different approaches used, but this review will analyze two articles that look at convolutional neural networks (CNN) to take a step toward real-time 3D object recognition.

CONVOLUTIONAL NEURAL NETWORKS

A neural network is modelled after the human brain. Consequently, it is made up of neurons that have learnable weights and biases that process a given input and produce a result from it's past learning. In the case of image processing, it takes each individual pixel of that image as input, and produces a result based off that.

CNNs work the same way as a normal neural network with one difference: they always assume that the input is an image. Simply put, this greatly reduces some of the functions of the neural network architecture which makes it more efficient to implement. Now that we know what a CNN is, let's see how it can be useful to computer graphics.

PROCESSING RGB-D IMAGES

Most images we see on the computer are RGB images. That is, each pixel consists of a value for red, blue, and green from zero to 255 each. When these pixels are arranged in a specified order with a certain color, it creates an image for the end user to see. RGB-D adds another aspect to RGB images called depth, thus in doing so, adds another dimension to the image. This new value, D, is only used by the computer, since it really doesn't concern the viewer. For instance, if you are viewing an image of a car on the screen, you don't need to see the values for depth at each pixel. Your brain will automatically process that image and add depth based on lighting, size, etc. Though a computer may be able to make an estimate of the depth in an image, it would be too resource and time consuming to be useful. Instead, RGB-D capture devices record depth from the source. An example of such a device is the Kinect for XBox consoles.

RGB-D is important to object recognition because depth is invariant to factors like lighting and color. This makes the detection more powerful, since it is better able to distinguish between background and foreground as well as the shapes and contours of an object. Socher et al discovered a method to use the power of RGB-D raw images and apply it to a machine learning algorithm to recognize objects within the images.

This methods begins by extracting the data from raw RGB-D images and sends that information to a CNN. This will provide information on low level features like edges and overall shape. The results from the first pass-through will go to a recursive neural network (RNN) which will learn other features about the image and how those features relate to each other. From there, the neural network will be able to make a reasonable guess as to what the image is, and remember the knowledge gained to make more accurate predictions in the future.[1]

This method works both in theory and in practice, but it has it's limitations. For instance, RGB-D images can potentially miss out on a lot of detail on an object, since it is only approaching it from one angle. Furthermore, the process of arranging and comparing pixels to an image (i.e. translational invariance) over and over is quite resource consuming, and there are many implementations of object recognition that will have strict memory and computing limits. Not surprisingly, this method is therefore too slow (at least, not reliably fast) to be feasible in real-time applications. A more recent method has been shown that uses volumetric CNNs and multitask learning to make a significant step toward reducing these limitations.

VOLUMETRIC CNNS AND MULTITASK LEARNING

To understand what volumetric CNNs we first need to understand why they're useful. 3D objects in the real world were first represented via RGB-D images, just to reiterate. Due to it's limitations, a new approach was taken named multi-view rendering. This rendering consisted of several 2D representations of the object from multiple angles. This solves the problem with RGB-D images that it was only one angle, but it still causes an issue since the angles and contours of the object are represented in 2D, not 3D. Volumetric CNNs were then introduced, which represent the object with 3D voxels, or "spots" on a grid, which are then sent to a convolutional deep belief network (CDBN) to perform classification and retrieval, as well as the prediction. This method is still commonly used today (even as a benchmark for other methods), but it still requires a lot of computational resources.[2] Since volumetric CNNs produce some of the most accurate results, an effort was made to improve efficiency. This is where multitask learning comes in.

Research has shown that multitask learning can greatly increase the efficiency of a network for a relatively low cost.[2] The paper goes in-depth as to how exactly this method works, but I will be focusing on how they numerically represent the object model since it better relates to graphics.

At first, an object must be virtually represented. There are several ways to do this with special cameras and similar devices, but the end result is usually a graphical mesh. A mesh is a system of unstructured edges and vertices that form many polygons who's faces take the form of the object. This type of abstract representation doesn't work well in neural networks, so voxels are used instead. Voxels look like cubes, and stack on each other on the x-, y-, and z-axis. Voxels stack in a way that closest fits the mesh of the object. Furthermore, each voxel contains a value of random probability that represents its likelihood of occupancy. You can sort of imagine them as a 3D pixel.

Once the object is represented by voxels, the machine learning model needs to learn. This is done by taking snapshots of the voxels while rotating it along the z-axis. To increase robustness of this new method, they introduce random jittering that will mirror the object on the x- and y-axis. By getting as many random angles of the object as possible, the network is able to learn more efficiently as well as produce more accurate results.

CONCLUSION

3D object recognition involves machine learning and computer graphics alike. An improvement in the methods to represent real-world images in virtual spaces will help improve the efficiency of the neural networks (and vice-versa).

From these articles, we saw how 3D object recognition began with simply adding another property to RGB images: depth. It had it's limitations, but that technology was used to build more efficient methods like the ones we see today, like multi-view and volumetric rendering. It's easy to understand how important 3D object recognition can be in the graphics field, since it has so many real-world applications. It has proven to be one of the hardest topics to research too, since improving either the graphical or machine learning aspect may be beyond our current capabilities. Without a doubt, we will continue to see this aspect of graphics improve over the near future.

Bibliography

[1] R. Socher, B. Huval, B. Bhat, C. D. Manning, and A. Y. Ng, "Convolutional-recursive deep learning for 3d object classification," in *Proceedings of the 25th International Conference on Neural Information Processing Systems - Volume 1*, NIPS'12, (USA), pp. 656–664, Curran Associates Inc., 2012.

[2] S. Zhi, Y. Liu, X. Li, and Y. Guo, "Toward real-time 3d object recognition: A lightweight volumetric cnn framework using multitask learning," *Computers & Graphics*, vol. 71, pp. 199 – 207, 2018.