June 07, 2019

PERCEPTION FIELD FOR A MOBILE DEVICE TO PROVIDE REAL-TIME DEPTH ESTIMATION FOR DETECTED OBJECTS

MingHung Hsieh

Shih-Yen Ku

I-Ray Chang

Yu-Sung LEE

Jhao-Yin Li

See next page for additional authors

Inventor(s)
MingHung Hsieh, Shih-Yen Ku, I-Ray Chang, Yu-Sung LEE, Jhao-Yin Li, YuKai Hung, and Wen-Yu Weng

PERCEPTION FIELD FOR A MOBILE DEVICE TO PROVIDE REAL-TIME DEPTH ESTIMATION FOR DETECTED OBJECTS

ABSTRACT

A mobile computing device (e.g., a mobile phone, camera, tablet computer, wearable and/or headset device) may include an integrated display device (e.g., a presence-sensitive screen) at which a user interface is presented to provide perception field functionality, which enables real-time depth estimation for static or moving objects that are detected by the mobile computing device based on sensory input from an onboard camera. In various examples, this functionality may be embodied in a portable and flexible library (e.g. Android library) that is installed on the mobile computing device. The purpose of perception field monitoring is to provide fast and efficient algorithms for spatial object mapping to enable real-time distance estimation of static and moving objects on a mobile computing device. The implementation of these algorithms may provide spatial location information of targeted objects, as well as distance information associated with objects that are detected by the device. In certain cases, mobile applications executing on the device may utilize such information to provide assistance to visually impaired users by creating audible alerts.

DESCRIPTION

As noted above, the purpose of perception field monitoring is to provide fast and efficient algorithms for spatial object mapping to enable real-time distance estimation of static and moving objects on a mobile computing device. In some cases, mobile applications executing on the device may utilize such information to create audible alerts that are associated with detected objects in the vicinity of the mobile computing device (e.g., for use by visually impaired users).

As a result, these applications may provide real-time collision prevention functionality to protect a mobile computing device, or users of the device, from colliding with objects. In addition, these applications may provide a real-time perceptive understanding of objects in the user's local environment. For example, as the device and/or user moves through a particular environment, the mobile application may output the name or identifiers of any detected or approaching objects (e.g., in left-to-right or near-to-far order). The mobile application may provide spatial location and depth information of objects that are detected by the device.

In general, the techniques described herein may be implemented on a mobile computing device (e.g., a mobile phone, camera, tablet computer, drone, Internet-of-Things enabled device, wearable and/or headset device), which may include an integrated display device (e.g., a presence-sensitive screen) at which a user interface is presented to provide perception field functionality. In various examples, the described functionality may be embodied in a portable and flexible library (e.g. Android library) that is installed on the mobile computing device. The use of this library by an application may provide real-time object distance estimation on various software and/or hardware frameworks.

Perception Field Library Overview

Figure 1, which is shown below, illustrates an overview of the perception field library, which may be implemented on a mobile computing device.

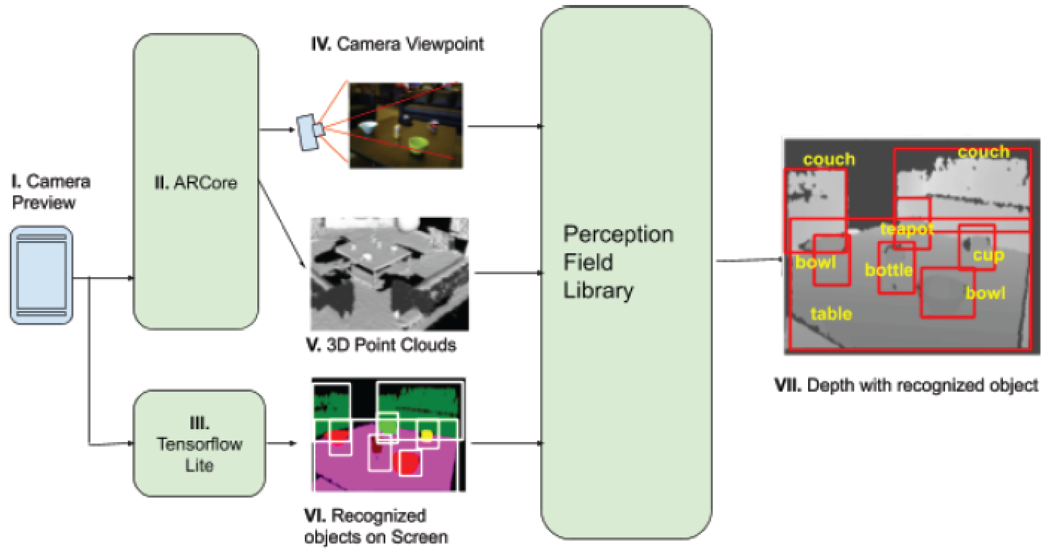

Figure 1: Overview of Perception Field Library

The perception field library of Figure 1 receives the matrices of the camera viewpoint and three-dimensional (3D) Point Clouds from a software library or development toolkit for building augmented reality applications to integrate virtual content with real-world content as detected via the camera of the mobile computing device (e.g., ARCore). ARCore is one non-limiting example of such a toolkit. ARCore is an open source software development toolkit for developing augmented reality applications that integrate virtual content with real-world content as detected via the camera of the mobile computing device. The toolkit utilizes motion tracking that allows the device to track its position within an environment, and it also provides environmental understanding functionality to allow the device to detect the size and location of surfaces. The perception field library of Figure 1 also receives the location of recognized objects on screen from a software library for dataflow and differentiable programming across a range of tasks (e.g., TensorFlow Lite), based on the camera preview provided by the mobile computing device, which may be a device that supports a dual camera. TensorFlow Lite is one non-limiting example of such a library. TensorFlow Lite is an open source software for dataflow and differentiable programming across a range of tasks, and in some cases, it may be used for machine-learning applications. Although ARCore and TensorFlow Lite are shown in Figure 1 and other figures, and are referenced herein, other libraries or modules that provide similar functionality may also be used in other examples to provide similar input into the perception field library.

Based on the received camera viewpoint, 3D point clouds, and locations of recognized objects on the screen, the perception field library estimates and reports the depth of recognized objects, as illustrated in Figure 1 and as described in further detail below.

Module Overview

In various examples, the perception field library may include multiple software modules. Figure 2 below illustrates examples of these software modules, as well as example data flows and dependencies between modules.

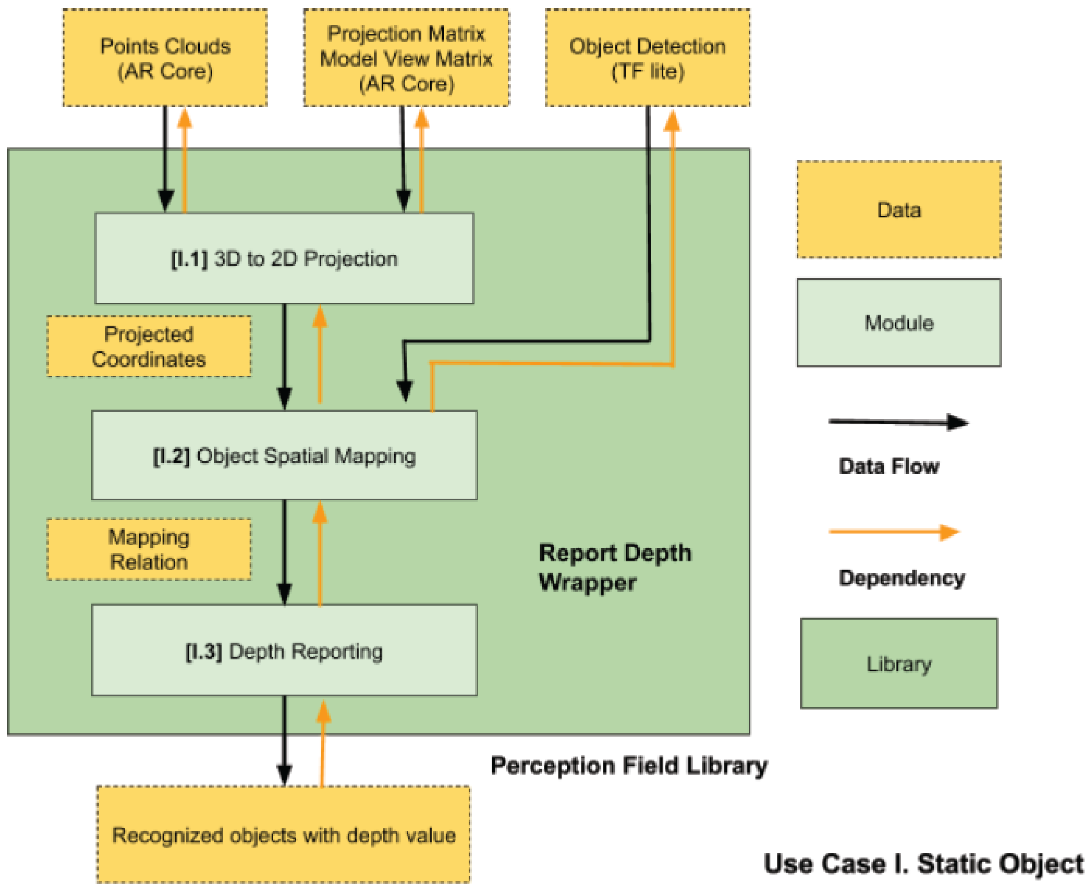

Figure 2: Data Flow and Dependencies of Modules For Static Objects

Case I. Static Objects

The perception field library of Figure 2 includes a Report Depth Wrapper that is capable of performing real-time or near-real-time distance and depth reporting for detected static (i.e., non-moving) objects in the environment. As illustrated in Figure 2, the 3D-to-2D Projection Module receives 3D Point Clouds from ARCore, which are the real-world 3D normalized coordinates of the environment as reported by ARCore. The 3D-to-2D Projection Module also receives a Projection Matrix and a Model View Matrix from ARCore. The Model View Matrix is a matrix that transforms real-world 3D coordinates to camera view 3D coordinates. The Projection

Matrix is a matrix that projects the camera view 3D coordinates onto the screen. The 3D-to-2D Projection Module uses the received inputs to project real-world 3D coordinates into 2D coordinates on the screen of the mobile computing device. This module may return or output the projected 2D coordinates of Point Clouds.

The Object Spatial Mapping Module of the perception field library, as shown in Figure 2, receives object detection information from TensorFlow Lite, which may include the location of recognized objects on the screen. Recognized object information may include rectangles that comprise the recognized objects. The Object Spatial Mapping Module also receives the projected 2D coordinates calculated from the 3D-to-2D Projection Module, and then returns or outputs the mapping relation between Point Clouds and recognized objects.

The Depth Reporting Module shown in Figure 2 receives the mapping relation from the Object Spatial Mapping Module and is configured to return depth, distance, and/or location information for recognized objects. This information may include information for regions or points of interest associated with the recognized objects. For example, the Depth Reporting Module may output statistical measurements of depth distribution on regions of interest, such as a mean depth value, maximum-minimum depth values, and/or standard deviation depth value associated with a respective region of interest.

Figure 3, as illustrated below, shows an example of object depth mapping that may be performed by the 3D-to-2D Projection Module and/or the Object Spatial Mapping Module for static objects, using a projection function to project real-world objects (3D point clouds) to 2D objects on the screen, and also using an inverse projection function to project the 2D objects on the screen to the real-world objects.

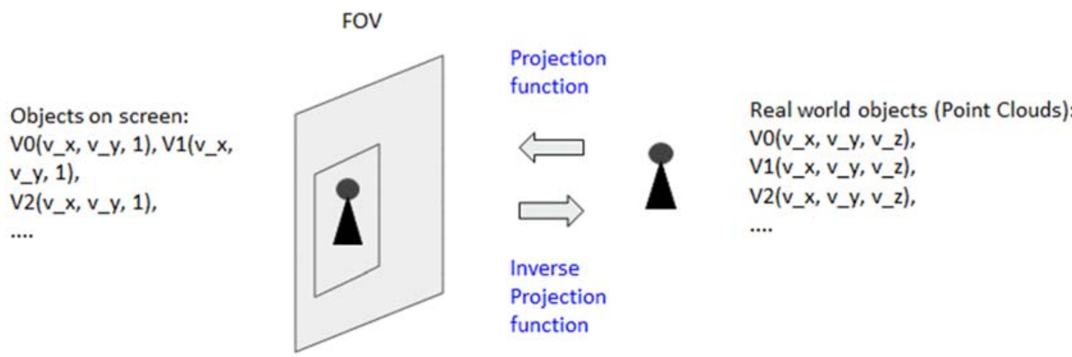

Figure 3: Object Depth Mapping For Static Objects

Each 3D point cloud may comprise a number of vertices having X, Y, and Z values, and each 2D object on the screen may comprise a number of vertices having X, Y, and Z values (where Z=1). As noted above, the 3D-to-2D Projection Module receives a Projection Matrix and a Model View Matrix from ARCore, and uses these inputs to project real-world 3D coordinates into 2D coordinates on the screen of the mobile computing device. This module may output projected 2D coordinates of Point Clouds. The Object Spatial Mapping Module receives objection detection information from TensorFlow Lite, and also receives the projected 2D coordinates calculated from the 3D-to-2D Projection Module. The Object Spatial Mapping Module then returns or outputs the mapping relation between Point Clouds and recognized objects, which is then used by the Depth Reporting Module, as described above.

Case II. Moving Objects

The perception field library is also capable of performing real-time distance and depth reporting for detected moving objects in the environment. Figure 4, shown below, illustrates an estimate depth wrapper that may be included in the perception field library for providing such functionality.

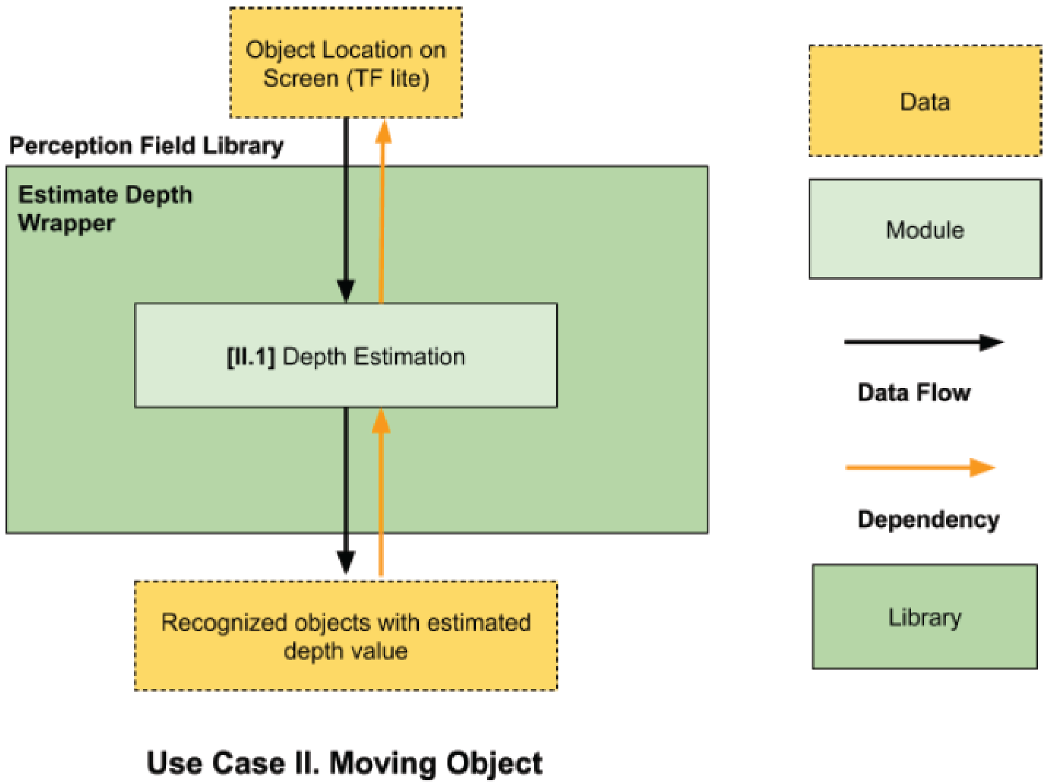

Figure 4: Data Flow and Module Dependency For Moving Objects

As shown in Figure 4, an Estimate Depth Wrapper may include a Depth Estimation Module. This module receives input regarding object detection and location information from TensorFlow Lite, which may include the location of recognized objects on the screen. The recognized object information may include rectangles that comprise the recognized objects. The Depth Estimation Module then predicts the depth of recognized objects according to the specific object type (e.g., humans, cats, dogs, cars, etc.). The Depth Estimation Module may utilize one or more different types of models, such as heuristic and/or machine-learning models, in providing the predicted or estimated depth and location values, which may be associated with corresponding regions or points of interest for the recognized objects.

Figure 5, as shown below, illustrates an example of object depth mapping that may be performed by the Depth Estimation Module.

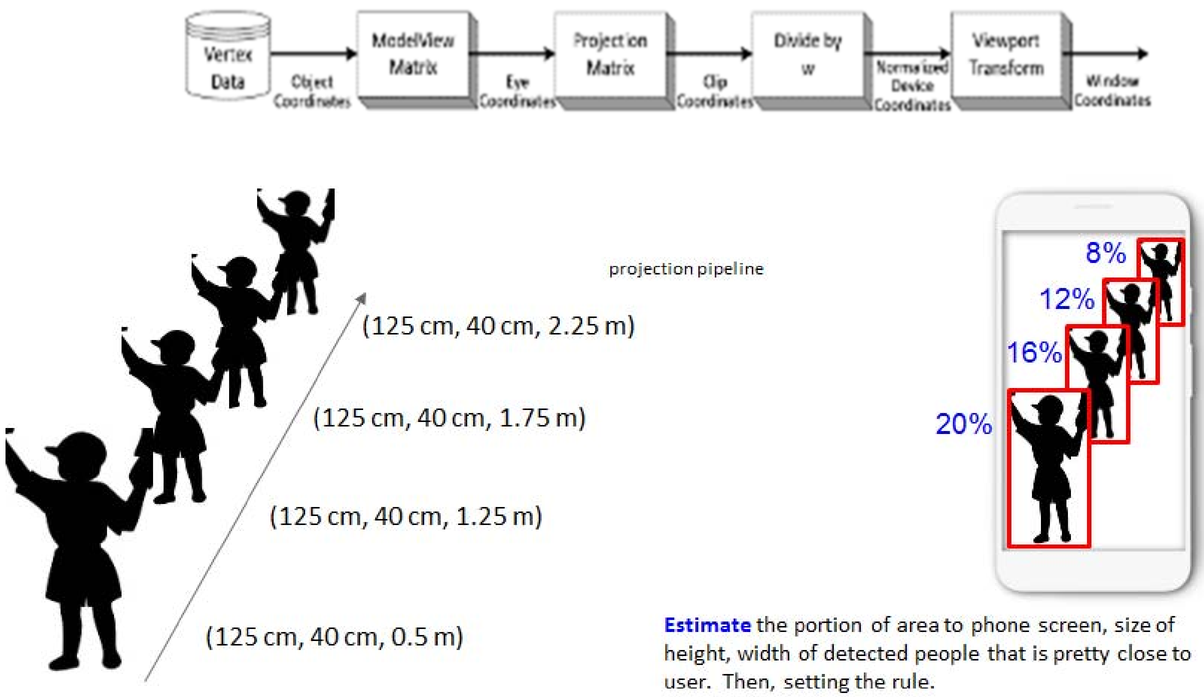

Figure 5: Object Depth Mapping For Moving Objects

As shown in Figure 5, the Depth Estimation Module may implement a 3D projection pipeline, similar to that described above with respect to the 3D-to-2D Projection Module, which is configured to project real-world 3D coordinates on 2D coordinates on the screen of the mobile computing device. As is shown on the left-hand side of Figure 5, a detected object (e.g., child) may, over time, move farther away from the mobile computing device. The height (i.e., 125 cm) and the width (i.e., 40 cm) of the detected object may not change, but the distance or depth of the object, relative to the mobile computing device, does change as the object moves farther away over time (i.e., 0.5 m to 1.25 m to 1.75 m to 2.25m). The Depth Estimation Module utilizes the recognized object information provided by TensorFlow Lite to predict the depth of recognized objects according to the specific object type (e.g., human child), and can output depth and location information of recognized objects for use in updating the 2D display on the screen of the mobile computing device.

As shown in Figure 5, as the object moves farther away from the mobile computing device, the Depth Estimation Module estimates the portion or area of the screen that will be occupied by a display of a 2D representation of the object at different points in time. As illustrated in Figure 5, when the object is closest to the mobile computing device, its 2D display occupies 20% of the screen real estate. However, as the object moves farther away over time, its corresponding 2D display occupies less and less of the screen real estate (i.e., 16%, 12%, 8%), as both the height and the width of the displayed object decreases. In some cases, the Estimate Depth Wrapper of Figure 4 may further include modules similar to the 3D-to-2D Projection Module and/or the Object Spatial Mapping Module described previously.

Figure 6, as illustrated below, shows an example of object depth mapping and tracking that may be performed by the Depth Estimation Module (and/or, in some cases, modules similar to the 3D-to-2D Projection Module and/or the Object Spatial Mapping Module described previously), using a projection function to project real-world objects (i.e., 3D point clouds) to 2D objects on the screen, and by also using an inverse projection function to project the 2D objects on the screen to the real-world objects. The perception field library may be configured to calculate and store all of the 2D and 3D projection information, such as that shown in Figure 3 above, but for any number of N preview frames, as illustrated in Figure 6. This functionality provides corresponding 2D and 3D object tracking over time, as an object may move in space relative to the mobile computing device.

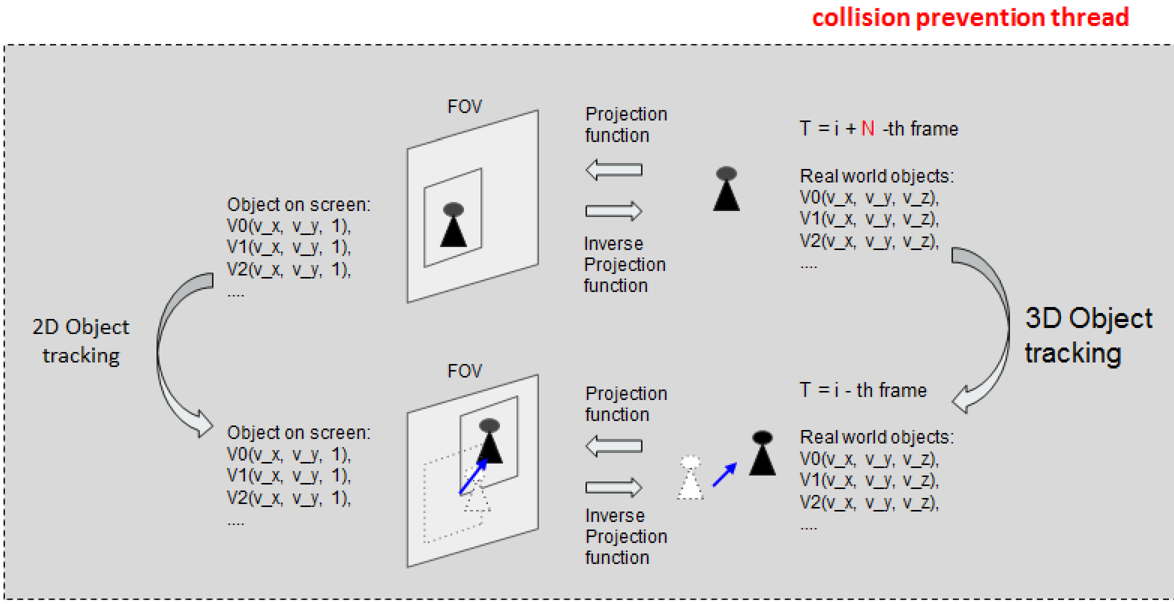

Figure 6: Object Tracking

As the 3D object moves in space, the 2D display and tracking of the object can be updated for depth and location purposes, and the representation of the object may be updated on the screen, such as shown in Figure 5. As described previously, in some cases, mobile applications executing on the device may utilize such information to assist with users who are visually impaired, such as by creating audible alerts associated with detected objects in the vicinity of these users. As a result, these applications may enable real-time collision prevention functionality, to protect the device or user from colliding with objects in real time. In addition, these applications may provide a real-time perceptual understanding of the objects in the environment in which the device or user operates. For example, as the user moves through an environment, the mobile application may output or otherwise report the name or identifiers of any detected or approaching objects (e.g., in left-to-right or near-to-far order). The mobile application may therefore provide spatial location and depth information of targeted objects.

REFERENCES

1. P. Tosteberg, “Semantic Segmentation of Point Clouds Using Deep Learning,” Master of Science Thesis in Electrical Engineering, pages 1-58, 2017.

2. A. Angelova, “A Structured Approach To Unsupervised Depth Learning from Monocular Videos,” Google AI Blog, https://ai.googleblog.com/2018/11/a-structured-approach-to-unsupervised.html, November 27, 2018.