OCCLUSION DETECTION USING DEEP LEARNING

author1

Coauthor
Affiliation
Address
email

ABSTRACT

Occlusion edges in images which correspond to range discontinuity in the scene from the point of view of the observer are an important prerequisite for many vision and mobile robot tasks. They can be extracted from range data however extracting them from image and videos would be extremely beneficial. We develop an unsupervised deep learning technique to identify occlusion edges in images and videos. Training data for the deep learning is generated from the depth values of an RGBD sensor, a color camera which also gives pixel ranges for short distances (¡ 5m).

1 INTRODUCTION

1.1 MOTIVATION

Occlusion edge detection is a fundamental capability of computer vision systems as is evident from the number of applications and significant attention it has received **?????**. Occlusion edges are useful for a wide array of tasks including object recognition, feature selection, grasping, obstacle avoidance, navigating, path-planning, localisation, mapping, stereo-vision and optic flow. In addition to numerous applications the concept of occlusions edges is support by the human visual perception research where it is referred to as figure/ground determination. Common fate motion **?** of occlusions edges and the internal texture is one of the primary mechanisms for the occluding edge determination in humans.

Once occlusion boundaries have been established depth order of regions become possible **??** which aids navigation, slam and path planning.

Occlusion edges help image feature selection by rejecting features generated from regions that span an occlusion edge. Because these are dependent on viewpoint position. Removing these variant feature saves on further processing and increases recognition accuracy. Interest point invariance under observer pose is an essential component of feature performance **?**.

Predominately objects are delineated by their spatial boundaries and particularly for rigid objects their shapes are intrinsically invariant as well as often being tied to their function. This form-function link means form as opposed to appearance is often more invariant which aids recognition, especially of object class rather than recognising particular object instances. Object recognition by matching shape of object means it may have any colour or pattern but will still be recognised. This is not the case for state of art sift based object recognition. In some situations the shape the object is better for recognition rather than its appearance, which can be easily dramatically altered e.g. painted objects, camouflage and people wearing different clothes.

Knowing the occlusion edges helps with stereo vision and optic flow algorithms Stereo vision and depth discontinuities. Stereo vision approaches often ignore information which is absent in either image. In **?** they instead take advantage of this unilateral information to as a strong clue to depth discontinuities. Optic flow is often a precursor as in **?** with good motion edge determination accomplished by bi-directional frame differencing for sequential video frames. **?** also points out that unlike color and stereopsis all visual species use motion as a cue, implying its importance.

?

The geometric edges of objects demarcate their spatial extent helping with grasping, manipulation as well as maneuvering through the world without collision. Knowledge of the occlusion edges is essential for effective moving rigid and articulated object tracking. By selecting regions of the object that do not span occlusion edges they are more likely to be completely on the object and hence more reliably tracked through methods such as template matching.

2 Background Material

Throughout this work it is assumed that appearance edges are a necessary but not sufficient condition for occlusion edges. This assumption is rarely violated in real world environments but when it is then even the human visual system fails.

There is utility of geometric edges for localisation and mapping (SLAM) for mobile robots. Many textureless environments are not suitable for feature based SLAM techniques despite being relatively common in indoor environments. However, maps based on occlusion and geometric edges will still allow localisation even in these low texture regions. For indoor mapping, planes seem a natural consideration for landmarks, and indeed numerous researchers have explored plane based mapping **?**. Although less common in robotic experiments, there are places not suitable for planar mapping including buildings with curved walls, natural outdoor environments and extremely cluttered scenes such as those found in search and rescue scenarios. Although planes can be a good way of compressing map information, an observed planar surface is not as constraining to robot pose as feature points and edges.

Another strong motivation for using the geometric edges is that they allow localisation by both range and image sensors. For many environments, considering only geometric edges, removes floors, walls and ceilings leaving the elements that lie along the intersection of planes or in cluttered regions, resulting in significant compression of the map data.

A rigorous definition of edge pixels is difficult. Edges manifest along paths of high contrast in images, and are due to four main reasons.

1. Texture change — Abrupt change in surface color.
2. Lighting change — Sharp shadows.
3. Range discontinuity — Abrupt change in distance from the observer.
4. Surface normal change — E.g. intersection of two planes.

It is important to appreciate the distinction in the causes of image edges. Texture change and illumination edges are not observed by 3D sensors. So the remaining geometric edge types are range discontinuities and abrupt surface normal changes. Surface normal changes are pose invariant, however edges due to range discontinuities can vary with observer position. These surface normal and range discontinuities are illustrated in the last image of Fig. **??**. The cylinder sides in Fig. **??** are examples of range discontinuities. The position of these edges varies in 3D space as the position of the observer shifts whereas the cylinder rim edge position is consistent regardless of observer position.

For use in mapping, we desire the following characteristics from extracted edge voxels: they should be generally invariant to rotation and translation, and they should be helpful in terms of constraining pose. We hence directly seek the third and fourth type of edges.

3 Related work

There are many motivations for occlusion edge determination, more than can be satisfactorily listed here, good reasons are included in numerous papers **?????**.

At this juncture a distinction should be made between maps containing the positions of landmarks or features and denser maps. Feature maps aid localisation but do not allow obstacle detection. Dense maps consisting of point clouds or occupancy grids enable localisation, path planning and obstacle avoidance.

Figures/edgevoxels_blend_image.jpg

Figure 1: A voxel map and the corresponding geometric edges for the *mason hallway* dataset.

Figures/cylinder.png
Figures/cylinder_image_edges.png
Figures/cylinder_range_edges.png

Figure 2: Image with associated edges due to appearance and due to geometry.

The following list summarises these maps in descending sparsity and places the mapping of edges into the context of the broader research. Sparser maps are smaller, easier to store and quicker to process however they do not work with as wide a range of robot tasks.

- 3D occupancy grid — Extension of 2D occupancy grids
- Occupied voxel list — Occupied voxels only
- Geometric edge map — Occlusion and/or surface normal edges
- Feature map — List of point features and their covariances

In **?** they extend conventional landmark based SLAM to incorporate edge information by the extraction of edgelets from the scene image.

The building of maps whilst considering all occupied voxels has proved successful for both indoor and outdoor environments **?**. There is a continuum in sparsity ranging from full 3D occupancy to feature maps. Feature extraction, whilst extremely helpful, comes at a price, namely reduced generalisation where mapping will fail in environments without the requisite features. It is observed that for indoor environments, while reliable point features can sometimes be absent, there are usually edge features. Edge mapping is faster and the associated maps are smaller and therefore require less memory. In the worst case, if there are not enough edges available it is possible to resort to full matching of the occupied voxels. For environments with planar surfaces there are far fewer edge voxels than occupied voxels. For conventional appearance edge extraction the structure tensor has been widely applied in image **?** and video analysis **??**.

The benefits of geometric edges for simultaneous localization and mapping (SLAM) has been established in **?**. Other approaches for detect geometric edges in 3D data are a keypoint detector based on a 3D extension of the Harris corner operator in the Point Cloud Library **?**. This detector operates on local normals of points A related approach for selecting interest points on 3D meshes was introduced in **?**.

4 Generating the Training data

To create an unsupervised training procedure we train with the depth data from an RGB-D sensor.

```
D = xyz_rgbs[:, :, 2]
RGB = xyz_rgbs[:, :, 3:].astype(np.uint8)
D[D > max_range] = 0

# simple edge detect on D to provide classification
E = ndimage.gaussian_gradient_magnitude(D, edge_sigma)
# filter out bad measurements e.g. too far or absorbing surface
C = np.zeros(E.shape)
C[E > edge_strength_threshold] = 1
structure = np.ones((dilate_size, dilate_size), dtype=np.bool)
```

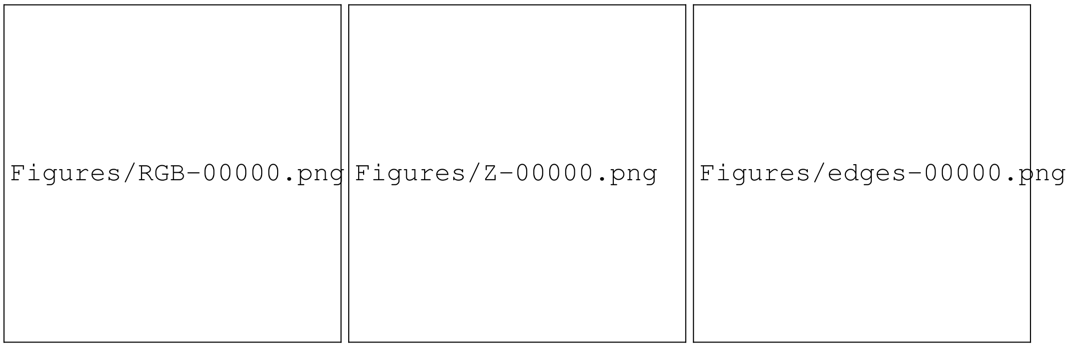

Figure 3: Example RGB, depth and classification frames from the training data generation procedure. In the classification frame gray signifies no edge, occlusion edges are white and black is for no or unreliable data.

Figure 4: Plot of occlusion edge recognition accuracy as a function of patch size

```
bad_inds = ndimage.binary_dilation(D==0, structure=structure, iterations=1)
C[bad_inds] = -1
```

5 RESULTS

ACKNOWLEDGMENTS