To appear in IEEE Int. Conf. on Advanced Video and Signal Based Surveillance (AVSS'14), Seoul, Korea, August 2014.

Scene Geometry from Moving Objects*

Eitan Richardson
BriefCam Ltd.
Israel

Shmuel Peleg
The Hebrew University
Jerusalem, Israel

Michael Werman
The Hebrew University
Jerusalem, Israel

Abstract

It has been observed that in most videos recorded by surveillance cameras the image size of an object is a linear function of the y coordinate of its image location. This simple linear relationship holds in the most common surveillance camera configurations, where objects move on a planar surface and the camera's X axis is parallel to that plane. This linear relationship enables us to easily perform and enhance several geometric tasks based on tracking an object over a few frames: (i) computing the horizon; (ii) computing the relative real world sizes of objects in the scene based on their image appearance; (iii) improving tracking by constraining an object's location and size. When the the camera's X axis is not parallel to the ground plane, after tracking a couple of objects it is possible to find the rotation which rectifies the video so that its new X axis is parallel to the ground plane.

1. Introduction

It has been established [2] that most recorded video comes from stationary surveillance cameras. Surveillance cameras are normally installed higher than the ground plane, observing this plane in an oblique view, with their X axis parallel to the ground plane. Under these conditions objects closer to the camera are located lower in the picture, and objects further from the camera are located higher up in the picture. This implies that the image of a moving object will be larger at the bottom of the picture compared to its image closer to the top of the image. See Fig. 1 (a).

Under assumptions that are valid for most surveillance scenarios, it has been observed [8, 3, 4, 6] that the relationship between the object size in the image and its vertical image position (y coordinate) is in fact linear. The coefficients of the linear function depend on the camera: its internal parameters, its external parameters (height and tilt angle), and on the object's size. Since in most cases these parameters are not known, most surveillance applications do not utilize the correlation between object size and its image location. Object detection is usually performed at all scale levels on the entire image, and tracking models initially assume that objects have a fixed size and a constant velocity in the image plane.

Several papers suggested using the linear relationship in a single image or in video to improve detection and tracking [4, 6], but required prior knowledge about the camera height and viewpoint or about objects' world sizes (for example, that the camera is held at eye level or that all detected objects are people). See Sec. 1.1.

In this paper, we propose a simple method to estimate the coefficients of the linear relationship between the image size of an object and its vertical image position, without requiring prior knowledge about the camera or the object size. The only requirement is that the camera's X axis is parallel to the ground plane. In fact, we show that tracking a single object enables to discover the horizon in the image. Furthermore, when the horizon is known, the size of an object in one video frame suffices in order to predict the object's image size at all image locations. We propose a robust method to estimate the horizon from multiple tracked objects. For the case where the surveillance camera's X axis is not parallel to the ground plane, we propose a method to find the angle between the camera's X axis and the ground plane, and rectify the video by rotating it with this angle.

The results presented in this paper can be used to improve video analysis for most surveillance videos:

- As a first step, after tracking a couple of objects, the video should be rotated so that its X axis is parallel to the ground plane. This step also computes the horizon.
- The world size of any pair of detected objects can be compared regardless of their image locations, and without computing the 3D scene. This can help filter objects of consistent sizes, such as people, cars, buses, etc.
- Object detection and tracking can be done much more reliably. With the change of an object's y coordinate we know exactly how its image size should change. We can therefore resize the search template according to the object's y location.

*This research has been supported by the Israel Science Foundation.

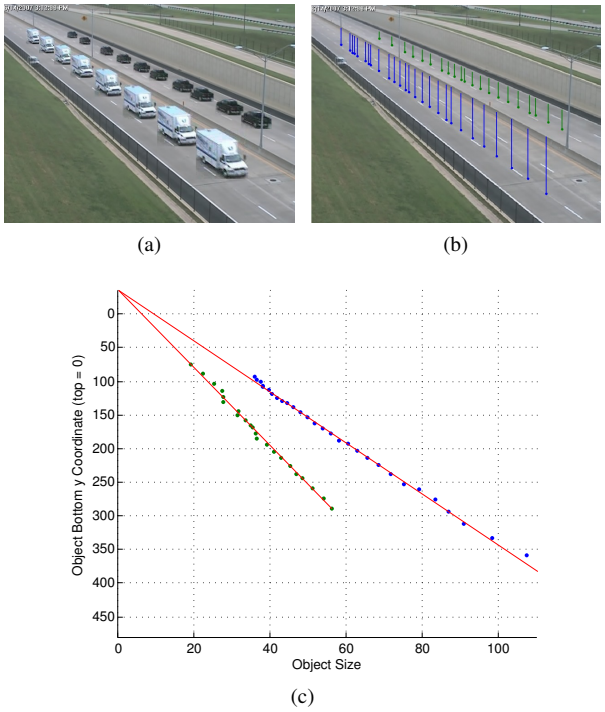


Figure 1. (a) Multiple instances of two tracked objects in a surveillance video. (b) A vertical bar is plotted for each instance, whose height is the size of the object and whose lowest point is the y coordinate of the bottom of the object. (c) A $size \times y$ graph showing the relation between object size (horizontal axis) and its y image location (vertical axis) for these two objects. All points belonging to a single object approximately lie on a straight line.

The rest of the paper is organized as follows: In Sec. 2 we present the linear function between the image y coordinate and the perspective scaling factor, and discuss when the measured object image size will be linear with y. Sections 3 and 4 present our robust method for finding the horizon and discuss the relationship between the height of the horizon to the size of objects in the image. In Sec. 5 we propose a method to rectify the video in the case that the camera is not horizontally levelled. We provide our conclusions in Sec. 6.

1.1. Related work

Traditional methods for computing the horizon in the image are based on vanishing points analysis, for example [10, 11]. If we extend the images of a set of lines which are parallel in the scene and which are also parallel to the ground plane, they intersect at the horizon. See Fig. 5. The method presented in this paper computes the horizon using tracked moving objects, without relying on background clues, which might be missing in the recorded scene.

Most of the work on image geometry involved the analysis of a stationary scene from multiple viewpoints [5]. An approach for finding scene structure from a single stationary video camera is presented in [9, 7], where information of scene structure is obtained from occlusion analysis.

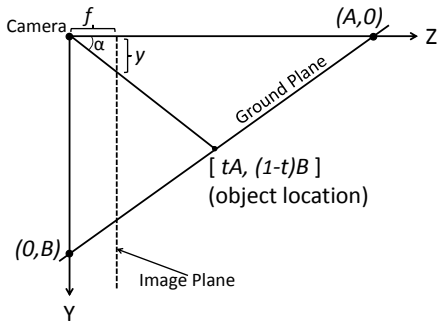


Figure 2. Imaging Geometry in camera coordinates. The ground plane intersects the camera's Z axis at $Z = A$, and intersects the camera's Y axis at $Y = B$. An object moving on the ground plane can be at locations $[tA, (1-t)B]$ for any real number t.

In [6], the authors use the linear relationship between object size and its image y coordinate to improve single-image object detection. A dependency tree between the camera viewpoint, object sizes and surface geometry is defined, and belief propagation is used for inference. The method assumes prior knowledge about the camera height and viewpoint (a camera is typically held at eye level and the horizon is at the center of the image) and about the object sizes. The prior probabilities are refined by inference between the camera height, viewpoint and object sizes.

The linear relationship is used in paper [4] to construct a scene depth map, in order to discover static occlusion and improve tracking. The object size in the image is defined as a function of its vertical image position (y coordinate), the vertical position of the horizon in the image and the *height expansion rate (HER)*. The two unknown parameters (horizon and HER) are estimated by accumulating a 2D histogram of all objects in the video, again assuming that all objects have a similar height.

In surveillance applications, cameras are installed at a variety of different heights and tilt angles. In addition, in outdoor scenes, the variance in object sizes can be high (a surveillance video can include large vehicles and pedestrians). The method presented in this paper does not assume any prior knowledge about camera height, tilt angle or object sizes.

2. Object's image size and its y image location

Fig. 2 shows the imaging geometry, where the camera is located at the origin pointing towards the positive Z axis. It is assumed that the X axis of the camera is parallel to the ground plane. Since in this case the image size of an object does not depend on its x coordinate, the camera's X axis

can be omitted.

In a perspective projection a world point (X, Y, Z) is mapped to the image point $(X\frac{f}{Z}; Y\frac{f}{Z})$. The projection is calculated by multiplying X and Y with the scale $\frac{f}{Z}$. As an object moves on the ground plane, it's depth Z changes. In Fig. 2 the ground plane is represented by the line $[t(A,0)+(1-t)(0,B)]$, and we derive the dependence of the scale of each point on the ground plane based solely on its vertical image location y.

Using similar triangles for the object located on the ground plane at location point $[t(A,0)+(1-t)(0,B)]$ on the segment from $(A,0)$ to $(0,B)$ gives:

$$\frac{f}{y} = \frac{tA}{(1-t)B} \tag{1}$$

solving for t we get:

$$t = \frac{fB}{Ay + fB}. \tag{2}$$

As the Z location of our object is tA, its scale according to the perspective rule is $\frac{f}{tA}$. Substituting Eq. 2 for t we get:

$$Scale = \frac{f}{tA} = \frac{1}{B}y + \frac{f}{A}. \tag{3}$$

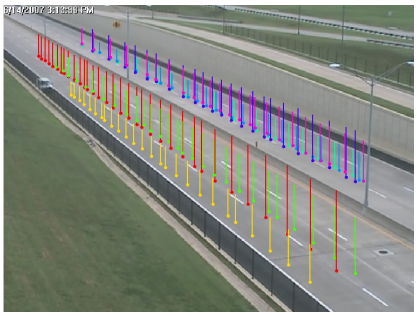
As f, A, and B are constants, the scaling of an object at depth Z is therefore a linear function of its vertical image location y.

If we track an object moving across the image, the width of the object will normally scale linearly with y according to Eq. 3. While the distances from the camera plane to the top and bottom of objects are not exactly identical in down looking cameras, in most cases these distances are similar enough so that the height of objects can also be assumed to be linear with y. Additional reasons for small deviations from the linear relationship are changes in the viewing angle of the 3D object along its trajectory and partial occlusions. In our experiments we use the square root of the area of an object as the linear measure of image size. We found this measure to be more robust than either height or width, giving a good approximation to the linear relationship.

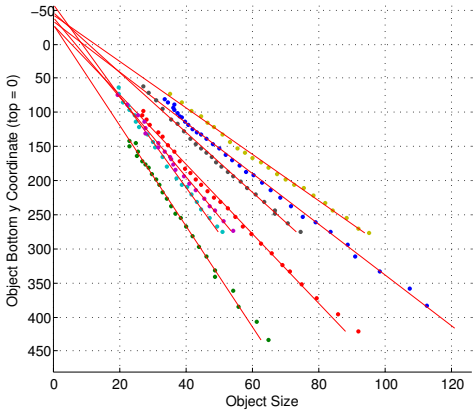
A plot showing the object size in the image, s, as a function of the vertical image location y is shown for two objects in Fig. 1, and for more objects in the same scene in Fig. 3. An additional scene is shown in Fig. 4. The linear relationship between object size and its y location is not limited to objects moving along a straight line, it is valid to any trajectory.

3. Finding the horizon

As objects move on the ground plane, their image size gets smaller as they get closer to the horizon, since their distance from the camera increases. Objects become infinitesimally small at the horizon. When the X axis of the camera



(a)



(b)

Figure 3. More objects in same scene of Fig. 1. (a) Object tracks are shown on the background image, with a different color for each object. (b) A $size \times y$ graph showing the relation between object size (X axis) and its y image location (Y axis) for all objects. All points coming from a single object approximate a straight line.

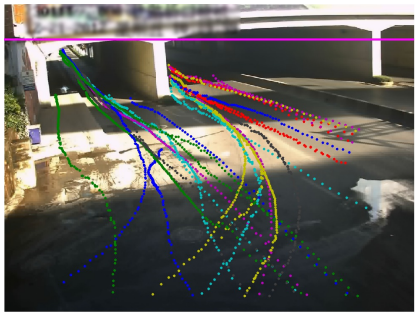
is parallel to the ground plane, the horizon in the image is horizontal, and can therefore be uniquely defined by its vertical image location, i.e. its y image coordinate. We can find the y coordinate of the horizon by computing the value of y where the object size becomes zero.

From Eq. 3 and because objects are infinitesimally small when $Scale = 0$, the location of the horizon y_h is:

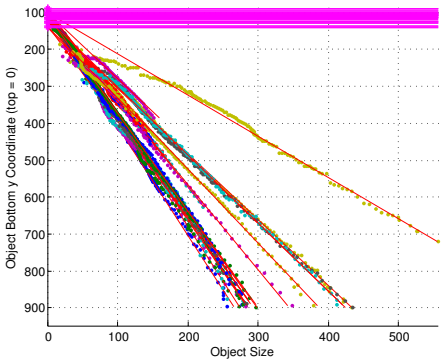
$$y_h = \frac{-fB}{A} \tag{4}$$

For each tracked object in the previous figures, the line fitting its points in the $size \times y$ graph intersects the y axis at the location of the horizon, y_h. It is especially interesting to observe this feature when the objects move on a curved path, such as in Fig. 4.

To estimate the horizon using multiple tracked objects, we present a robust method based on the Hough transform [1]. As can be seen in Fig. 6, each pair of detections of the same object points to the horizon y_h (the intersection of the line passing between the two points with the y axis). Since the tracked object can be partially occluded or the angle in



(a)



(b)

Figure 4. (a) Tracks of moving objects are drawn over a background image, with a different color for each object. (b) A $size \times y$ graph showing the relation between object size (X axis) and its y image location (Y axis) for all objects. All points coming from a single object approximate a straight line. The intersection of each line with the Y axis indicates the y location of the horizon, shown as a horizontal line in both (a) and (b)

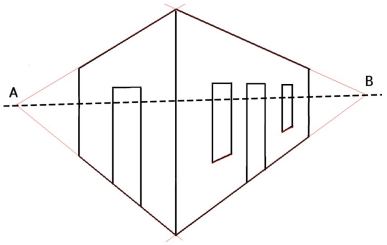


Figure 5. The horizon is the dashed line and A and B are two points on the horizon at the intersections of the lines extending the segments representing the parallel edges of the building (From Wikipedia). Each pair of parallel lines is also parallel to the ground plane.

which it is viewed by the camera changed, a robust method is required. For each tracked object we sample random pairs of observations and update the global y_h histogram for each pair. If the conditions for linearity between objects size and

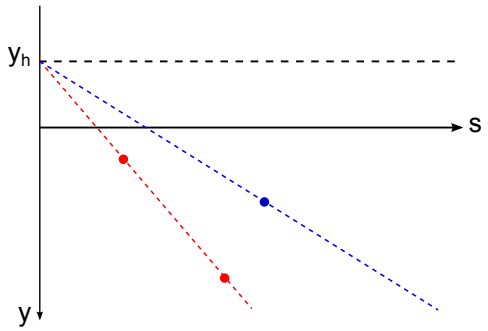


Figure 6. Two instances of a tracked object (red) point to the y location of the horizon in the image - y_h (above the image top in this example). The horizon and a single instance of a second object (blue) predict the size of this object for all y values.

Figure 7. A sample horizon probability function from our robust method for computing the horizon. Each pair of object observations votes for a horizon y value (the intersection of the line passing through the two (y, s) points with the y axis. The histogram peak (red line) matches the actual scene horizon.

y coordinate are met (all objects are positioned on a plane which is parallel to the camera X axis), the resulting horizon probability function will have a sharp peak at y_h. See Fig. 7 for an example.

It is clear that when tracking the bottom of an object that is always located on the ground plane, we get a straight line in the $size \times y$ graph, a line that intersects the Y axis at the horizon. But it should be noted that we can track other parts as well. Assume we track a person's head. Since the head moves on a plane parallel to the ground plane, its horizon is the same as the horizon of the ground plane as all planes parallel to the ground plane share the same horizon.

4. Relative size of objects

We established in Sec. 2 that the image size s of an object whose world size is S depends linearly on the scale. Using Eq. 3, we get:

$$s = \frac{S}{B}y + \frac{fS}{A}. \quad (5)$$

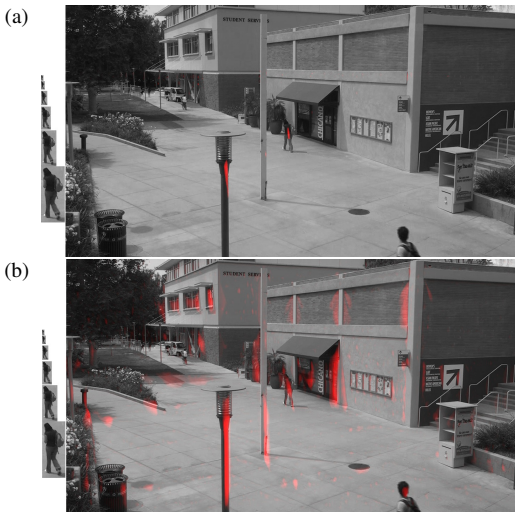


Figure 8. (a) The person on the left is scaled as a function of the y image location and is searched in the image. Matches are denoted in red. Only few false matches appear. (b) The object is searched at all scale levels on the entire image. The number of false matches is larger by a factor of 17. **This figure is best viewed in color.**

Eq. 5 indicates that the slope of the line in the $size \times y$ graph is linear with the real size S of an object. Thus, the ratio of slopes of lines in the $size \times y$ graph belonging to different objects is equal to the ratio of the objects' world sizes

$$\frac{S/B}{S'/B} = \frac{S}{S'}. \tag{6}$$

There is no need to track objects in order to determine their relative sizes if the y location of the horizon is known. Since we know that all lines in the $size \times y$ graph pass through the y axis at the location of the horizon, a single observation of an object giving its image size and its y location is needed. Connecting this point to the horizon on the y axis determines the slope of this object, and therefore is relative size. This is an immediate and easy way to determine if a small object at the top of the image may be larger in the real world than a large object at the bottom of the image.

The first step before comparing object sizes is to determine the y location of the horizon. The horizon can be found by tracking one or more "good" objects: objects that are easy to track, and whose y coordinate changes. The horizon is determined as described in Sec. 3. After finding the y location of the horizon, determining relative object sizes is straight forward.

Fig. 8 compares the result of searching (using normalized cross correlation) for an object template at all scale levels all over the frame to searching at a scale level that matches each y coordinate, according to the learned horizon. With our method, in addition to fewer computations, the number of false detections is much smaller.

5. Finding the Z rotation

The analysis described in this paper assumes that the camera is level, and its X axis is parallel to the ground plane. When the camera is not level, and its X axis is not parallel to the ground plane, we need to rectify the video and make it level. The video can be made level by rotating the video frames about the Z axis so that the X axis will be parallel to the ground plane.

We have shown in Sec. 3 that, when the camera is level, an object's trajectory is a line in the $size \times y$ graph and that all these lines intersect the Y axis in a single point, the location of the horizon. This does not hold when a camera is not level. The video can be rotated about the Z axis[1], until reaching a rotation angle α for which all estimates of the horizon have the lowest variance. An example of finding the Z rotation of the camera, and rectifying the X axis to be horizontal, is shown in Fig. 9 and Fig. 10.

6. Conclusion

This paper presented a simple and robust method to improve object detection and tracking and to calculate relative world-sizes of objects, based on the linear relationship between object image-size and its y location in the video frame. Unlike previously suggested methods, we do not require any prior knowledge about the object world-sizes or about any camera parameters. After establishing that the perspective scale factor changes linearly with the image y coordinate, we observed that all observations of the same tracked object (regardless of its size) lie on a line in the $(y, size)$ space and all these lines intersect at the horizon (at point $(y_h, 0)$).

Once a couple of objects are tracked reliably in the video, the video can be rotated so that its X axis is horizontal and the y location of the horizon can be determined. With the horizon known, we know how the size of every object in the video changes with changing y coordinate, as its object line must intersect the Y axis at the horizon. This knowledge improves tracking substantially by adjusting the object size based on its predicted y location. In addition, when real object size is a criterion for filtering, pointing to one instance of an interesting object can pull out all others of same size regardless of their different image size.

References

[1] D. H. Ballard. Generalizing the hough transform to detect arbitrary shapes. *Pattern recognition*, 13(2):111–122, 1981.

[2] J. Gantz and D. Reinsel. The digital universe. Technical report, IDC, Sponsored by EMC Corporation, December 2012.

[1]Note that it is not important around which coordinate the image is rotated as the rotation is a rigid transformation of the image and parallel lines are preserved.

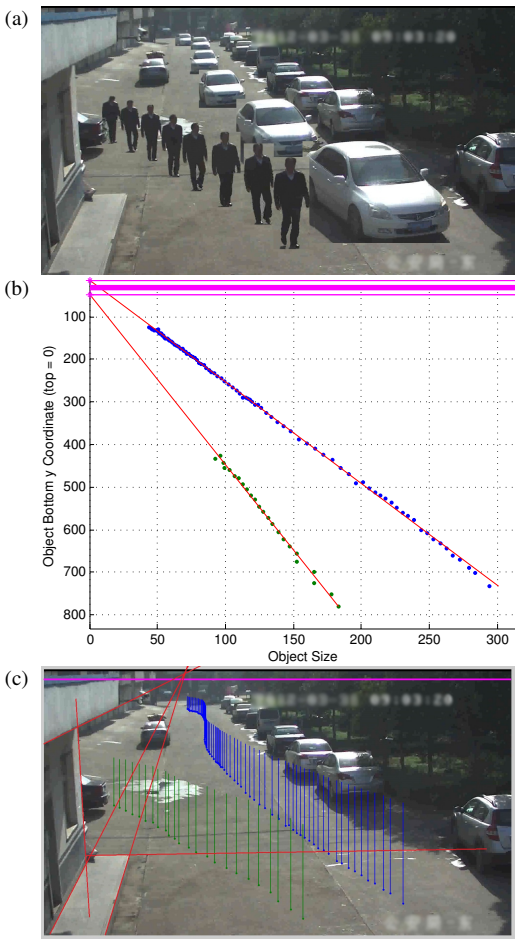


Figure 9. (a) Multiple instances of two tracked objects in a scene, a person dressed in black and a white car. (b) The $size \times y$ graph of the two objects tracked in (a). Each object line intersects the Y axis at a different location, indicating that the assumption of an horizontal X axis may be wrong. (c) The trajectory of the tracked objects with vertical bars indicating object sizes. The horizontal line near the top is the horizon predicted by (b). The horizon predicted by the vanishing points is located higher in the image.

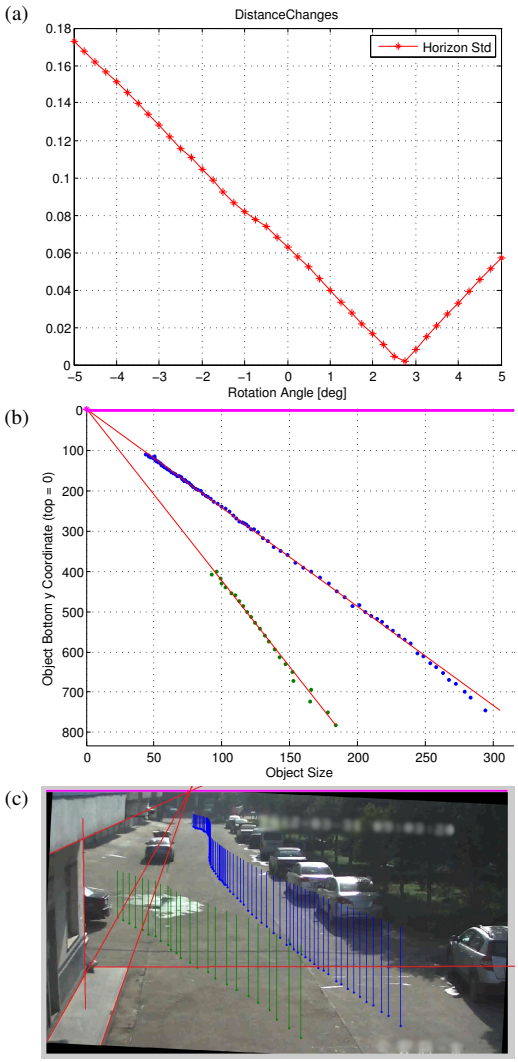


Figure 10. Rectifying the camera in Fig. 9. (a) The distance in the $size \times y$ graph between the two intersection with the Y axis of the two objects. The minimum is at 2.8 degrees. (b)Rotating the original video by 2.8 degrees found in (a), brings the two object lines to intersect the Y axis at the same point. (c) Rotating the original video by 2.8 degrees found in (a) brings the horizon found in (b) to be the exact same horizon found using vanishing point.

[3] D. Greenhill, J. Renno, J. Orwell, and G. Jones. Learning the semantic landscape: embedding scene knowledge in object tracking. *Real-Time Imaging*, 11(3):186–203, 2005.

[4] D. Greenhill, J. Renno, J. Orwell, and G. Jones. Occlusion analysis: Learning and utilising depth maps in object tracking. *Image and Vision Computing*, 26(3):430–441, 2008.

[5] R. I. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, second edition, 2004.

[6] D. Hoiem, A. A. Efros, and M. Hebert. Putting objects in perspective. In *CVPR'2006*, volume 2, pages 2137–2144, 2006.

[7] A. Kowdle, A. Gallagher, and T. Chen. Revisiting depth layers from occlusions. In *CVPR'2013*, pages 2091–2098, 2013.

[8] J. Renno, J. Orwell, and G. Jones. Learning surveillance tracking models for the self-calibrated ground plane. In *British Machine Vision Conference, BMVC'2002*, pages 607–616, 2002.

[9] A. Schodl and I. Essa. Depth layers from occlusions. In *CVPR'2001*, volume 1, pages 639–644, 2001.

[10] F. A. Van den Heuvel. Vanishing point detection for architectural photogrammetry. *International archives of photogrammetry and remote sensing*, 32:652–659, 1998.

[11] Y. Wang, E. K. Teoh, and D. Shen. Lane detection and tracking using b-snake. *Image and Vision computing*, 22(4):269–280, 2004.