Shadow Removal : Its effectiveness for images captured from real world scenarios

[1]Sagar Sahni, [2]B.Venkata Kranthi
[1]*Scholar*, [2]*Assistant Professor*
[1,2]*Department of Electronics & Communication Engineering, Lovely Professional University, Jalandhar*

Abstract— This paper presents the problem of shadow detection and removal from strong shadows on the road that confound the detection of the boundary between clear path and obstacles, making clear path detection algorithms less robust. The presence of shadows has been responsible for reducing the reliability of many computer vision algorithms, including segmentation, object detection, scene analysis, stereo, tracking, etc. Therefore, shadow detection and removal is an important pre-processing step for improving performance of such vision tasks. Decomposition of a single image into a shadow image and a shadow-free image is a difficult problem, due to complex interactions of geometry, albedo, and illumination. We present an algorithm to detect strong shadow edges & use spatial smoothing and gaussian filtering to further improve shadow edge detection.

Keywords— *Edge patch candidates, Feature extraction, Spatial smoothing, Gaussian filter.*

I. INTRODUCTION

The fundamental goal of image correction is that whenever we capture an image then there are imperfections in the image due to practical limitations of the hardware or location where we capture the image which becomes significant as the images become dimmer. However, with the use of special digital cameras such type of imperfections are greatly reduced & for remaining imperfections different software's are used to characterize the remaining image imperfections and remove them from the image. One of such imperfections that we discuss here is the shadows in the image, how they are detected & removed, with the goal of providing an image that is free from such imperfections. Shadows can be divided into two major classes i.e. Self (Form) shadow and Cast shadow.

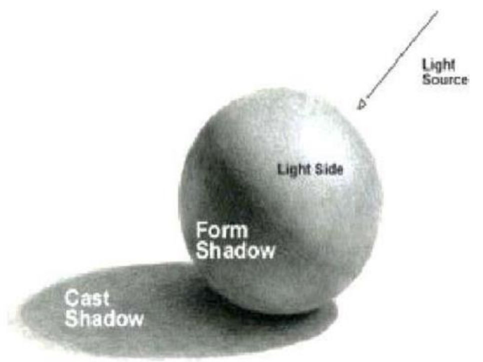

Fig.1: Shadows can be Broadly Divided into Cast and Self Shadow.

Cast shadows can be further classified into umbra and penumbra region because of multi lighting and self shadows have many sub-regions such as shading and interreflection. Usually, the self shadows are basically vague shadows and do not have clear boundaries whereas cast shadows are hard shadows and always have a violent contrast to background. Because of their different properties, the methods to handle these two kinds of shadows are different.

Shadow detection and removal over the past decades covers many specific applications & nowadays object shadow detection has been an active field of research for several decades. Most researches focus on providing a general method for various images and thereby obtaining "visually pleasing" shadow free images. Many techniques have been proposed for removing shadows from images [1]. These different techniques are used based on shadow properties such as:

- **Model based techniques-**For this technique it is assumed that we know the 3D geometry & illumination of scene. This includes the sensor/camera localization, the geometry of observed objects and the light source direction from which a priori knowledge of shadow areas is derived. To explain this, consider polygonal regions to approximate the shadows of buildings or urban elements in some simple urban scenes. But this technique has limitations that it does not give approximate result in case of quick bird images & sometimes geometry of scene & light sources are unknown.
- **Image based techniques-** This technique make use of certain image shadow properties such as color (or intensity), shadow structure (umbra and penumbra hypothesis), boundaries etc. without any assumption about the scene structure. Some common ways of exploiting image shadow characteristics are:
  - The value of shadow pixels must be low in all the RGB bands. Shadows are, in general, darker than their surrounding, thus it is delimited by noticeable borders (shadow boundaries).
  - Shadows do not change the surface texture and the surface markings tend to continue across a shadow boundary under general viewing conditions.
  - In some color components (or combination of them) no change is observed whether the region is shadowed or not, that is, this is invariant to shadows.
- **Color/ Spectrum based techniques-** This technique attempts to describe the color change of shaded pixel and find the color feature that is illumination invariant. It was study that shadows change the hue component slightly and

decrease the saturation component significantly. The pixels of shadow region cluster in a small region that has distinct distribution compared with foreground pixels. These shadows are further discriminated from foreground objects by using empirical thresholds on HSV color space.

- **Texture based techniques-** Texture has been described by five different properties in the psychology of perception i.e. coarseness, contrast, directionality, line-likeness & roughness. The principle behind the textural model is that the texture of foreground objects is different from that of the background whereas the texture of shaded area remains the same as that of the background. Texture synthesis can be used to fill in holes in images & create large non-repeative background images & expand small pictures.
- **Geometry based techniques-** This method makes use of the camera location, the ground surface, & the object geometry, to detect the moving cast shadows. The model using this technique is parameterized with several features including the orientation, center position and mean intensity of a shadow region with the orientation and centroid position being estimated from the properties of object moments.

The limitations of presently available shadow detection techniques in still images are given below:-

- Needs of some prior knowledge, for example, human's interaction.
- Effective in specific application.
- Failing when working on some complex scenes.

II. SHADOW EDGE DETECTION

The detection of shadow and shading edges is a first step towards reducing the imaging effects that are caused by interactions of the light source with surfaces that are in the scene. Shadow removal relies on the classification of image edges as shadow edges or non-shadow edges. There are many methods that are used so far for shadow edge detection. Different types of geometric features can also be used to analyze possible patterns in geometry that are characteristic for shadow edges. The three features are explained as:

- **SIFT:** SIFT is an algorithm in computer vision to detect and describe local features in images [2]. The algorithm was published by David Lowe in 1999. SIFT keypoints of objects are first extracted from a set of reference images and stored in a database. This feature has proven to be very effective for object and scene recognition, and it describes the orientation of edge responses in a small region around a point. The original SIFT-feature is composed of a detector and descriptor phase. However, considering the size of the patches, only the descriptor is used to describe the content of a patch. Applications include object recognition, image stitching, robotic mapping and navigation, 3D modelling, video tracking, individual identification of wildlife and match moving.
- **Local Binary Pattern:** Local binary patterns is a type of feature used for classification in computer vision and was first described in 1994. It has since been found to be a powerful feature for texture classification [3]. It has further been determined that when it is combined with the histogram of oriented gradients classifier, it improves the detection performance considerably on some datasets. It describe a texture using a histogram of binary patterns, where every binary pattern corresponds to one pixel in a region. The binary pattern for a pixel P describes the relative values of neighbouring pixels: neighbours with lower values than pixel P are assigned the value 0, while the other neighbours (i.e. with higher or equal values) are assigned the value 1. By concatenating the values of all neighbours, a binary pattern for pixel P is obtained and the binary patterns of all pixels in a region are summarized into a histogram. The histogram of an entire patch is used as feature.
- **Grey-level Co-occurrence Matrix:** The Grey-level co-occurrence matrix is a tabulation of how often different combinations of pixel brightness values (grey levels) occur in an image [4]. It is a well-studied texture feature that captures the relationship between intensity values that occur simultaneously. The result is matrix C, where an element (i, j) indicates the number of times elements i and j co-occur in an image I at a given offset (Δx, Δy). Thus, Grey-level co-occurrence matrix texture considers the relation between two pixels at a time namely, the reference and the neighbour pixel.

Besides these geometric features there are other techniques that are used for shadow edge detection. One of such techniques that are used is patch based shadow edge detection in which we process edges using patches instead of pixels as working with individual edge pixels is prone to noise and computationally expensive [5]. In this method we analyze the image features which are able to distinguish shadow edges and non shadow edges to train a robust shadow edge classifier. Furthermore, spatial patch smoothing is applied to enforce consistency between neighboring patches. And at last we filter the shadow edge using Gaussian Filter to get accurate results and in the end we get the shadow free and filtered image.

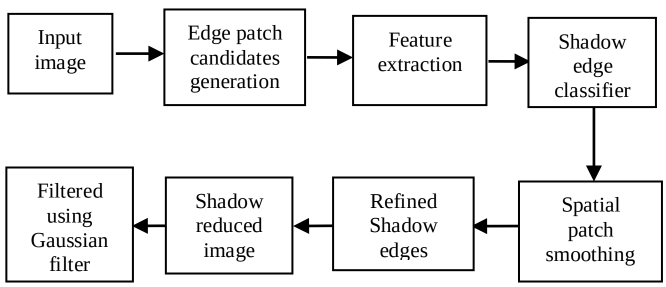

Fig.2: Overview of Shadow edge detection & removal.

III. SHADOW REMOVAL APPROACH

A. Generation Of Edge Patch Candidates

To find the adaptive gradient threshold to extract edge candidates we use linear regression model. For this a threshold is computed from the gradients obtained from an image to extract the edge candidates. Here we find the cumulative sum of the image gradient magnitudes when the gradient magnitudes are smaller than the threshold and cumulative sum of the whole image gradient magnitudes are calculated separately, and the ratio between these two cumulative sums is computed. A linear regression model is trained by the inputs of ratios and sums of whole image gradient magnitudes from different training images. Thus, given a new image, the ratio is computed by applying this regression model with sums of new image gradient magnitude as the input, and then, local shadow edge threshold for this image can be calculated based on the learned ratio. In this, we process the edges using patches instead of pixels to find all edge patches consisting of shadow edge patch and non-shadow edge patch.

B. Feature extraction

This stage distinguish the shadow edge patches from all edge patches and then determine strong shadow edges from these patches [6]. Here we propose three types of features: illuminant invariant features, illumination direction features and neighbouring similarity features.

(i) Illuminant invariant Features

For this we convert the input image from standard RGB space into two color spaces which have reduced illumination effects: illuminant invariant chromaticity space as discussed in [7] and hue and saturation as suggested in [9].

Then, for each illuminant-invariant color space, we extract two features to characterize one patch. 1) variance of colors in illuminant-invariant color spaces as pixel values from same surface in shadow edge patch have a smaller variance while pixel values from different surfaces in object patches exhibit a larger variance. 2) Entropy of gradients on illuminant invariant color spaces as in the absence of illumination effects, the texture of surface in shadow edge patch can be described by gradients with smaller entropy whereas texture of multiple surfaces in non-shadow edge patch leads to larger entropy of gradients.

(ii) Illumination Direction Features

The illuminant-invariant theory [7] shows that 2D log-chromaticity values (generated by (log(R/G), log(B/G)) from a single-color surface under a variety of illuminations form a straight line parallel to the illumination direction.

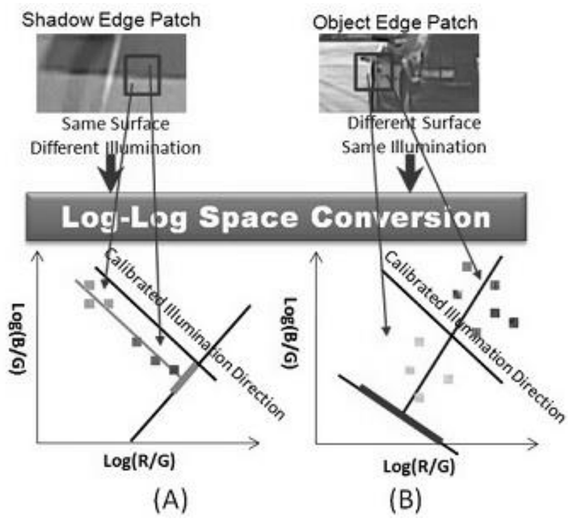

Fig.3: (A) 2D log-chromaticity of a shadow edge patch.

(B) 2D log-chromaticity of a non-shadow edge patch.

Therefore, the 2D log-chromaticity values of shadow edge patch from the same color surface fit a line parallel to the calibrated illumination direction as shown in Figure 3(A). Also they have a small variance after projecting on to the illuminant-invariant direction (perpendicular to the illumination direction). On other hand for a non-shadow edge, its 2D log-chromaticity fits a direction other than the illumination direction and generates the projection to its perpendicular direction with large variance as shown in Figure 3(B)).

(iii) Neighboring Similarity Features

The features above are extracted from individual edge patches. However, neighboring patches on both sides of an edge can also provide evidence to distinguish shadow edges from non-shadow edges. Hence, we first identify the pair of neighboring patches located on the two sides of an edge [8]. The rules for choosing neighboring patch pair are based on categorizing edge orientation into 4 zones shown in Figure 4. Then, we employ two features (gradient features & texton features [9]) which capture the texture differences between the pair of neighboring patches.

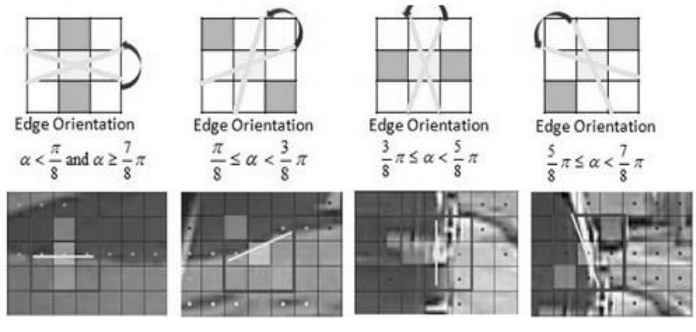

Fig.4: Rules for choosing neighbouring patch pair. The first row describes the neighbouring patch selection rules and the second row gives some examples on real images.

C. Shadow edge detection

Shadow edge detection problem act as a per-patch labeling task, where every patch is classified as either being a shadow edge patch or a non-shadow edge patch [10]. For this propose, we employ a binary Support Vector Machine (SVM) classifier. These are supervised learning models with associated learning algorithms that analyze data and recognize patterns which are used for classification and regression analysis. The basic Support Vector Machine takes a set of input data and predicts that for each input given which of two possible classes that forms the output and making it as a non probabilistic binary linear classifier. Therefore, this classification method provides a fast decision and outputs probabilities, which has been successfully used in a variety of other vision tasks for clear path detection [11]. This classifier provides the probability $P_j(c)$ of both classes ("shadow edge patch" and "non-shadow edge patch") of each patch j based on patch's features. Finally, we use maximal likelihood estimate < $\hat{c}_j$> = *arg* $\max_c P_j(c)$ to identify each patch's initial SVM classified label of "shadow edge patch" ($\hat{c}_j = 1$) or "non-shadow edge patch" ($\hat{c}_j = 0$). The probabilities and classifier decisions are used in the next section as inputs to spatial patch smoothing module for achieving improved results.

D. Spatial Patch Smoothing

It has been seen that shadow edges and non-shadow edges have neighboring connectivity which can be utilized to improve detection results. Here we propose spatial smoothing approach to exploit the consistency across neighboring edge patches to remove isolated false detection.

In the feature extraction stage, we obtain the edge orientation. Then, we render an orientation ray along with this patch's edge orientation and passing the mean location of all edges in this patch. Among 8 neighboring patches, the patches passed by this orientation ray are marked as the neighboring edge patches for the current edge patch (Fig.5). We use both neighboring edge patches for spatial patch smoothing.

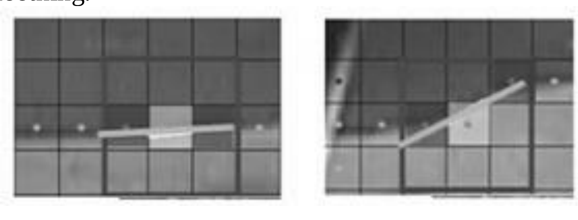

Fig 5. Neighbouring Edge Patch Identification

The edges in neighboring patches should have an influence on current patch's label if they share similar edge orientation, as shadow edges and non-shadow edges are continuously connected. We set the spatial smoothness coefficient of patch j to be $n_j(c)$. Let *l* denote one of the current patch j's neighboring patches with its associated initial probability from SVM $P_l(c)$ and maximal likelihood estimate $\hat{c}_l = \max_c P_l(c)$ obtained from SVM. We define spatial smoothness coefficient $n_j(c)$ to be:

$$n_j(c) = \prod_l N(c; \hat{c}_l, \sigma_l^2) + \varepsilon \tag{1}$$

where $N(c; \hat{c}_l, \sigma_l^2)$ is the Gaussian distribution and ε is a small constant (e.g. 10^{-10}).

To find the Gaussian Distribution, maximal likelihood estimate $\hat{c}_l = \max_c P_l(c)$ & variance is given as

$$\sigma_l^2 = \frac{g}{P_l(c)^2 b_l N(\Delta_{j,l}; 0, \sigma_\Delta^2)} \tag{2}$$

where g is scale factor of variance, σ_Δ^2 is variance of edge orientation similarity, $N(\Delta_{j,l}; 0, \sigma_\Delta^2)$ which measures the edge orientation difference i.e. $\Delta_{j,l} = |\alpha_l - \alpha_j|$ between patches j and l and b_l is the no. of edge pixels which measures the strength of edges in neighboring edge patches.

If patch j and its neighboring patch l have similar edge orientations, and patch j class is consistent with its neighbor's label estimates (they are both classified as the same type of edge), we expect spatial smoothness coefficient $n_j(c)$ of patch j to be large. Finally, the probability of the patch j is updated as follows:

$$P_j^{'}(c) = \frac{n_j(c)}{\sum_{c\varepsilon(0,1)} n_j(c)} \tag{3}$$

E. Gaussian filter

Gaussian smoothing is a fundamental process that is used in almost every computer vision application. A Gaussian filter is a filter whose impulse response is a Gaussian function and is designed to give no overshoot to a step function input while minimizing the rise and fall time. This type of behavior is closely connected to a fact that the Gaussian filter has the minimum possible group delay. The Gaussian smoothing operator is a 2-D convolution operator that is used to `blur' images and remove detail and noise. In this regards it is similar to mean filter but it uses different kernel that represent the shape of a Gaussian (`bell-shaped') hump. The degree of smoothing is determined by the standard deviation of the Gaussian. With higher resolution images one is also often wanting to use Gaussian filters with correspondingly larger standard deviations. The standard deviation of an averaging filter of width w is

$$\sigma_{av} = \sqrt{\frac{w^2 - 1}{12}} \tag{4}$$

If we perform n averaging with the same filter the variances of the filters add to produce an overall filtering effect equivalent to a standard deviation of

$$\sigma_{nav} = \sqrt{\frac{nw^2 - n}{12}} \tag{5}$$

From this equation we compute the ideal width of the averaging filter that one should use to achieve filtering that is equivalent to that obtained with a Gaussian of standard deviation σ

$$w_{ideal} = \sqrt{\frac{12\sigma^2}{n} + 1} \qquad (6)$$

The accuracy of the standard deviation can be improved by increasing n but it is worth keeping n as small as it is practical to reduce the edge effects in the final filtered result. With each averaging filter pass the edge effects propagate further into the image. If we define the 'radius' of an averaging filter as being (width-1)/2, the width of the edge affected boundary will be n×radius. For n = 5 this boundary width will be slightly greater than the 3σ that is typically allowed for in Gaussian smoothing. So, overall it is worth keeping the number of passes small and not more than 6.

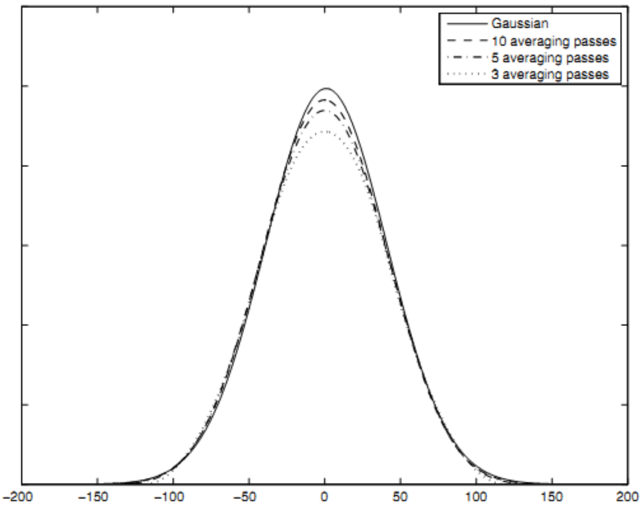

Fig.6: Approximating a Gaussian with standard deviation of 40 using 3, 5 and 10 averaging passes.

Thus, the Gaussian outputs a `weighted average' of each pixel's neighborhood with average weighted more towards the values of the central pixels. This aspect is generally in contrast to the mean filter's uniformly weighted average. Because of this a Gaussian provide gentler smoothing and preserves edges better than a similarly sized mean filter.

IV. EXPERIMENTAL RESULTS ANALYSIS

In this section, we compare the results of the Patch based strong shadow removal with the method which combines photometric features and geometric features for patch-based classification of shadow edges. We compared the performance of recent approach used in [10] with Patch based strong shadow removal in different scenarios. Table I summarizes the performance comparison in which Accuracy, False Alarm Rate (FAR) and False Reject Rate (FRR) are defined as the percentage of all correctly classified patches, the ratio of the wrongly classified non-shadow edge patches to all the non-shadow edge patches and the ratio of the wrongly classified shadow edge patches to all the shadow edge patches, respectively.

TABLE I. COMPARISON WITH GIGSENIJ'S METHOD [10] ON PATCH BASED SHADOW DETECTION.

	Gijsenij Method[10]			Patch based Method		
	Acc.	**FAR**	**FRR**	**Acc.**	**FAR**	**FRR**
Scenario 1	85.9%	13.1%	14.7%	95.1%	5.1%	5.9%
Scenario 2	80.3%	15.7%	16.9%	90.7%	8.3%	9.0%
Scenario 3	89.8%	11.2%	13.8%	95.9%	4.7%	5.5%
Scenario 4	81.1%	14.9%	18.2%	91.5%	7.3%	8.2%
Overall	**84.2%**	**12.8%**	**15.0%**	**94.1%**	**5.5%**	**6.7%**

It is notable that the patch based method achieves accuracy of 94.1%, 5.5% FAR (False Alarm Rate) and 6.7% FRR (False Rejection Rate) & thus has accuracy of 9.9% higher than the method proposed in [10] which also has high FAR & FRR which is 12.8% and 15% respectively.

(A) (B)

Fig.7: Results of Strong Shadow Removal via Shadow Detection.

(A) Without Shadow Removal; (B) With Shadow Removal.

V. CONCLUSION

Clear path detection on road is very essential and shadows can reduce the ability of computers for clear path detection. So it is essential to detect the shadows and remove them. Untill now different methods or techniques have been used to detect the shadows. In this paper we have discussed a novel approach to detect shadow edges and remove shadows in images. We used patch based edge detection technique for shadow edge detection and removal and gaussian filter for filtering the result to get a clear vision of road track. The high energy points can be easily identified and smoothed with the help of Gaussian filter. It also had been studied that gaussian filters are useful for applications requiring large bandwidth filters. Thus, after filtering the result using gaussian filter, the final image shows significant improvement to generate the shadow free image without any visual artifacts.

ACKNOWLEDGMENT

I sincerely thank the reviewers for the detailed suggestions that have significantly improved the presentation of this material.

VI. REFERENCES

[1] Sanjeev Kumar, Anupreet Kaur, “Algorithm for shadow detection in real colour images,” International Journal on Computer Science and Engineering, Vol. 02, No. 07, pp.2444-2446, 2010.

[2] D.G. Lowe, “Distinctive image features from scale invariant keypoints,” International Journal of Computer Vision, Vol. 60, No. 2, pp. 91–110,2004.

[3] T. Ojala, M. Pietikainen, and T. Maenpaa, “Multiresolution gray-scale and rotation invariant texture classification with local binary patterns,” IEEE Trans. Patt. Anal. Mach. Intel., Vol. 24, No. 7, pp.971–987, 2002.

[4] J. van de Weijer, Th. Gevers, and J.M. Geusebroek, “Edge and corner detection by photometric quasi-invariants,” IEEE Trans. Patt. Anal. Mach. Intel., Vol. 27, No. 4, pp. 625–630, 2005.

[5] Q. Wu, W. Zhang, and B.V.K. Vijaya Kumar, “Strong Shadow Removal Via Patch-Based Shadow Edge Detection,” IEEE Conference on Robotics and Automation (ICRA), 2012.

[6] J.F. Lalonde, A.A. Efros, and S.G. Narasimhan, “Detecting ground shadows in outdoor consumer photographs,” European Conference on Computer Vision (ECCV), pp. 212-218, 2010.

[7] C. Lu, M.S. Drew, and G.D. Finlayson, “On the removal of shadows from images,” IEEE Tran. on Pattern Analysis and Machine Intelligence (PAMI), pp.134-142,2006.

[8] Q. Wu, W. Zhang, T.Chen, and B.V.K. Vijaya Kumar, “Prior based vanishing point estimation through global perspective structure match,” IEEE Conference on Acoustics, Speech and Signal Processing(ICASSP), 2010.

[9] J. Winn, A. Criminisi, and T. Minka “Object Categorization by Learned Universal Visual Dictionary,” International Conference on Computer Vision (ICCV), pp.120-128,2005.

[10] A. Gijsenij and T. Gevers, “Shadow edge detection using geometric and photometric features,” IEEE International Conference on Image Processing (ICIP), 2009.

[11] Q. Wu, W. Zhang, T. Chen, and B.V.K. Vijaya Kumar, “Camera based clear path detection,” IEEE Conference on Acoustics, Speech and Signal Processing (ICASSP), 2010.