The automatic determination of atmospheric visibility from real-time images of video cameras

Qun Chen, Wenbing Chen, Jin Cheng,
Yumei Huang, Jackie Shen, Yanbo Wang*

1 Introduction

"Visibility" is defined as the greatest distance at which a black object of suitable dimensions, located near the ground, can be seen and recognized when observed against a scattering background of fog, sky, etc. This definition is based on human observation of atmospheric visibility conditions.

Nowadays, video cameras are widely used for remote monitoring and security surveillance purposes. Visibility estimation using digitized video images through identification of targets at known distance from the camera is a potential application. It has an intrinsic advantage over the instrumental light-absorption/scattered approaches in that the image acquisition processes between the camera lens system and the human eye are similar.

The use of images from video cameras for the automatic observation of visibility is non-trivial. The determination of visibility from images is affected by the size and shape of the reference targets, the contrast of the targets and their background and weather conditions. The objective of this research topic is to develop an efficient automatic algorithm to compute visibility through the processing of images captured by video cameras.

These images are taken by video cameras at fixed places so we can say that the locations of landmarks in the images are fixed. This information is very useful and we can mark the boundary of the landmarks manually. But because of wind and some other reasons, small vibrations still may happen and this has to be considered in the algorithm.

2 Briefing of the method

In this section we will propose an algorithm for determining visibility based on an image edge detection method. We will give an index value about the visibility. A landmark is considered to be visible if we can identify it from the background. This means that the color difference between the landmark and background is large enough. While edge detection is also to determine the color difference between the object and background. So image edge detection method may work for this problem.

Since our output is only a value about visibility and it is not as simple as Yes or No, one still needs a threshold to decide if the value means visible or invisible. The threshold can be determined by experience and we must keep in mind that different landmarks have different threshold values.

Here we will use a new method for edge detection([1],[2]). Though the image is a 2-D image, we will treat it as a 1-D problem(line by line) and we believe that there is a color function which is unknown to us and the color value at each pixel is the value of the color function with noise. Our target is to reconstruct the color function. Based on the theorem that the L^2 norm of the second order derivative of the reconstructed color function will blow up at the boundary of an object, we reconstruct the color

*E-mail: ybwang@fudan.edu.cn

function and calculate the L^2 norm of the second order derivative, and use the average norm of the small domain around the boundary as an index. The theorem can be found in ([1]) but here we are only interested in the procedure and algorithm.

We give the outline of our method as follows:

1. Consider line-by-line scans of an image with size $N_1 \times N_2$ where N_1 and N_2 are positive integers. For simplicity we only consider 8-bit gray level pictures, which means that the colour value will vary from 0 to $2^8 - 1 = 255$. (The formula for converting RGB color value to gray level is very easy to be found in internet. Here we use $gray = Red * 0.3 + Green * 0.6 + Blue * 0.1$.
2. The colour values of an digital image are noisy observation data at discrete points. Here we denote the colour value of a pixel at (i, j) as $c(i, j)$.
3. Let $M = 2^8$ and define $y_i^\delta(x_j) = \frac{c(i,j)}{M}$, where $x_j = \frac{j-1}{N_2-1}$, $j = 1, 2, \cdots, N_2$. Then $0 \le y_i^\delta(x_j) < 1$ and the value of the grid length h is $\frac{1}{N_2-1}$.
4. The numerical differentiation algorithm developed in [2] is then applied to find the approximation of the real colour function $y_i(x)$, $0 \le x \le 1$, from the noisy values $y_i^\delta(x_j)$ on each line. The approximation is denoted by f_{*i}. We will give the detail algorithm later.
5. We will calculate the L^2 norm of the second order derivative of f_{*i} near the boundary of landmarks. Since f_{*i} is a cubic spline function so it is very easy to get the second order derivative of f_{*i}. We will also show the formula later.
6. It usually takes two different directions of line-by-line scans to obtain an acceptable approximation result. Note that the directions are not required to be perpendicular.

If the index value is large, the landmark is considered to be visible. If the index is small, the landmark is considered to be barely visible or invisible. One advantage of this method is, it is stable for the pictures with noise, which is proved in ([1]).

3 Algorithm

Now we will give the detailed algorithm of the method.

The reconstructed color function f_* by using the method in ([1]) and ([2]) is a piecewise cubic spline function. It can be defined as follows:

$$f_*(x) = a_j + b_j(x - x_j) + c_j(x - x_j)^2 + d_j(x - x_j)^3,\ j = 0, \cdots, n-1, x \in [x_j, x_{j+1}]$$

where there are a total of $4n$ parameters whose unique values can be obtained from solving the following equations:

$$f_*^{(i)}(x_j+) - f_*^{(i)}(x_j-) = 0,\ i = 0, 1, 2;\ j = 1, 2, \cdots, n-1, \tag{3.1}$$

$$f_*^{(3)}(x_j+) - f_*^{(3)}(x_j-) = \frac{1}{\delta^2(n-1)}(\tilde{y}_j - f_*(x_j)),\ j = 1, 2, \cdots n-1, \tag{3.2}$$

$$f_*^{(2)}(0) = 0,\ f_*^{(2)}(1) = 0, \tag{3.3}$$

$$f(0) = y(0),\ f(1) = y(1). \tag{3.4}$$

For notational convenience, denote T and Q to be the following $(n-1)\times(n-1)$ matrices:

$$T = \begin{pmatrix} \frac{4h}{3} & \frac{h}{3} & 0 & \cdots \\ \frac{h}{3} & \frac{4h}{3} & \frac{h}{3} & \cdots \\ \cdots & \cdots & \cdots & \\ \cdots & \frac{h}{3} & \frac{4h}{3} & \frac{h}{3} \\ \cdots & 0 & \frac{h}{3} & \frac{4h}{3} \end{pmatrix}, \tag{3.5}$$

$$Q = \begin{pmatrix} \frac{-2}{h} & \frac{1}{h} & 0 & \cdots \\ \frac{1}{h} & \frac{-2}{h} & \frac{1}{h} & \cdots \\ \cdots & \cdots & \cdots & \\ \cdots & \frac{1}{h} & \frac{-2}{h} & \frac{1}{h} \\ \cdots & 0 & \frac{1}{h} & \frac{-2}{h} \end{pmatrix}, \tag{3.6}$$

and a, c, $\widetilde{y}$ to be the vectors

$$a = (a_1, \cdots, a_{n-1})^T, \tag{3.7}$$
$$c = (c_1, \cdots, c_{n-1})^T, \tag{3.8}$$
$$\widetilde{y} = (\widetilde{y}_1, \cdots, \widetilde{y}_{n-1})^T. \tag{3.9}$$

From equation (3.3), we have

$$c_0 = 0,\ d_{n-1} = -c_{n-1}/(3h).$$

Putting $c_n = 0$ in equation (3.1) for the case when $i = 2$, we obtain

$$d_j = (c_{j+1} - c_j)/(3h), \quad j = 0, 1, \cdots, n-1. \tag{3.10}$$

From equation (3.4), we have

$$a_0 = \widetilde{y}_0,\ a_n = a_{n-1} + b_{n-1}h + c_{n-1}h^2 + d_{n-1}h^3 = \widetilde{y}_n.$$

From equation (3.1) for the case when $i = 0$, we have

$$b_j = (a_{j+1} - a_j)/h - c_j h - d_j h^2, j = 0, 1, \cdots, n-1. \tag{3.11}$$

Now, from the case when $i = 1$ in equation (3.1), we have

$$Tc = Qa + \begin{pmatrix} \frac{\widetilde{y}_0}{h} \\ 0 \\ \vdots \\ 0 \\ \frac{\widetilde{y}_n}{h} \end{pmatrix}.$$

From equation (3.2), we have

$$Qc = \frac{1}{2\delta^2(n-1)}(\widetilde{y} - a).$$

Finally, we have

$$c = (T + 2\delta^2(n-1)Q^2)^{-1}(Q\widetilde{y} + z),$$

and

$$a = \widetilde{y} - 2\delta^2(n-1)Qc,$$

where $z = (\frac{\widetilde{y}_0}{h}, 0, \cdots, 0, \frac{\widetilde{y}_n}{h})^T$. The expression of the minimizer f_* in each interval $[x_j, x_{j+1}]$ can now be obtained by computing the parameters b and d from equations (3.10) and (3.11). Note that the square of the L^2 norm of f_*'' in each interval $[x_j, x_{j+1}]$ is given by:

$$l_j = \int_{x_j}^{x_{j+1}} (2c_j + 6d_j(x - x_j))^2 dx = 4c_j^2 h + 12c_j d_j h^2 + 12d_j^2 h^3.$$

4 Numerical Examples

In this section we will show the numerical results for some pictures with real data. We mark the boundary of the landmarks with dark black color.

First are pictures of an island with different weather condition (sunny and rainy)

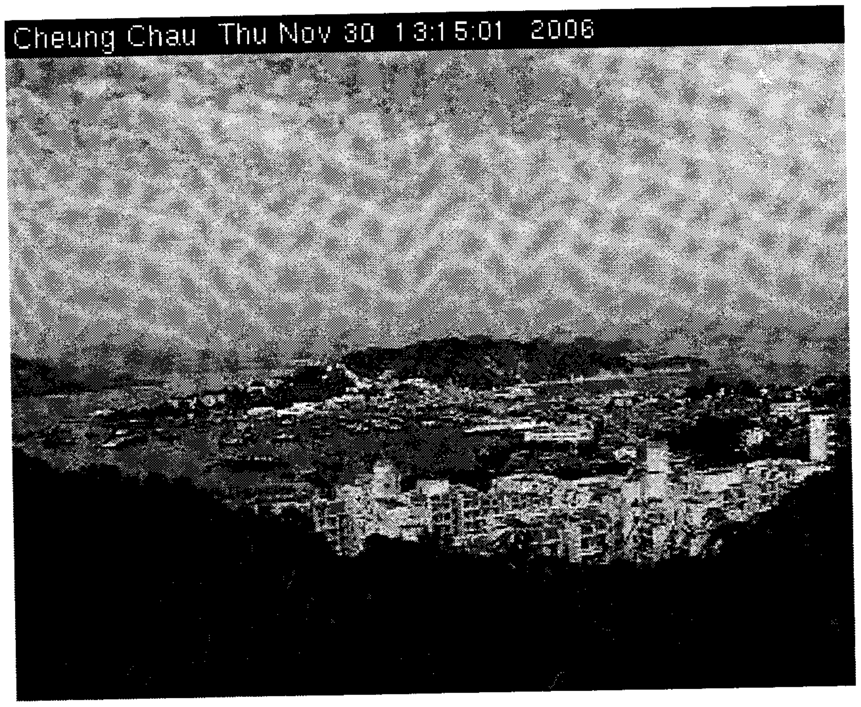

Figure 1a: Sunny

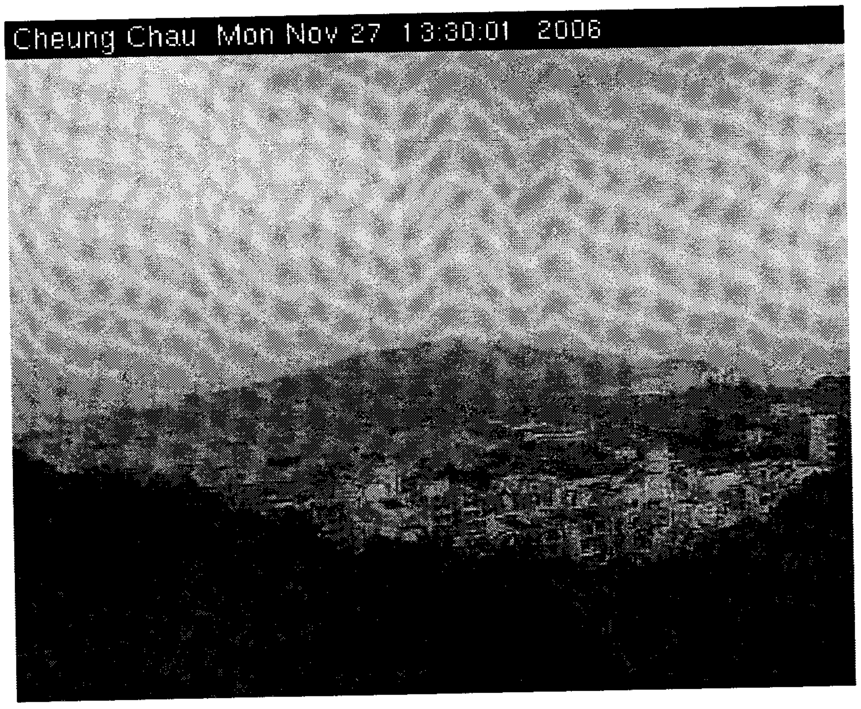

Figure 1b: Rainy

And two landmarks are chosen as follows:

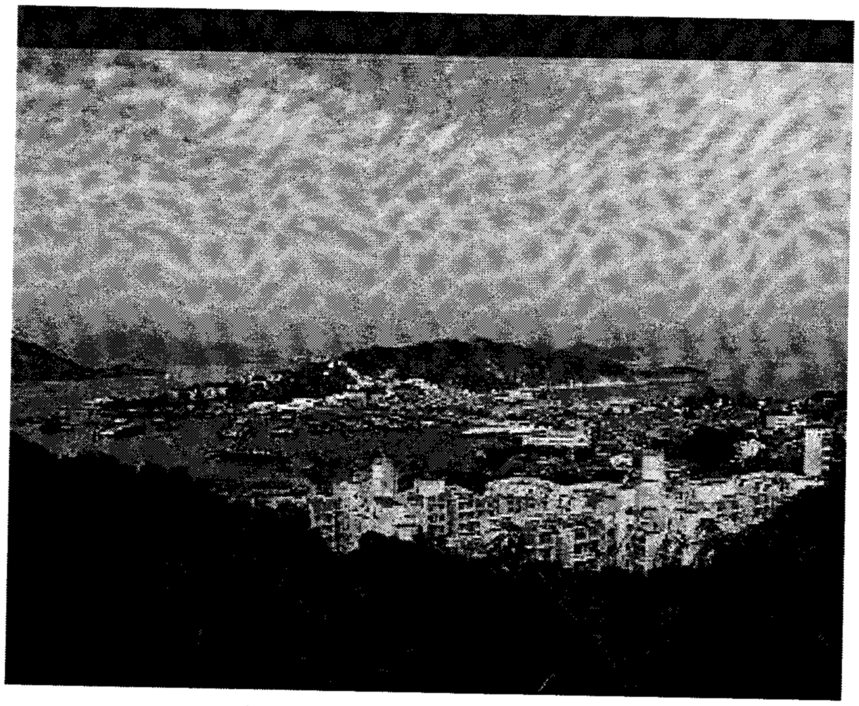

Figure 2a: small island on the left

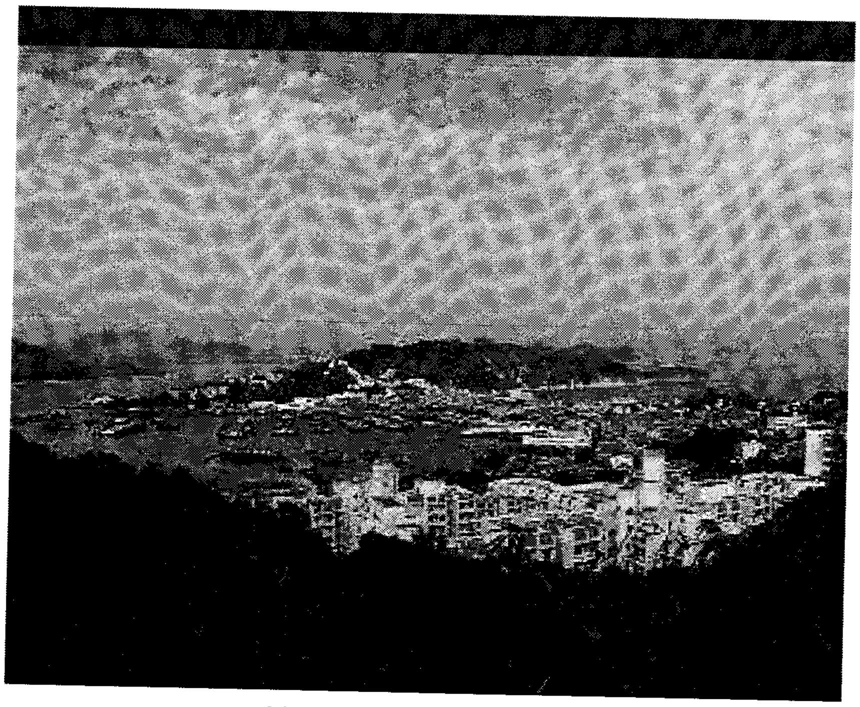

Figure 2b: top of the island

The next is another place: harbor, also with sunny and rainy pictures.

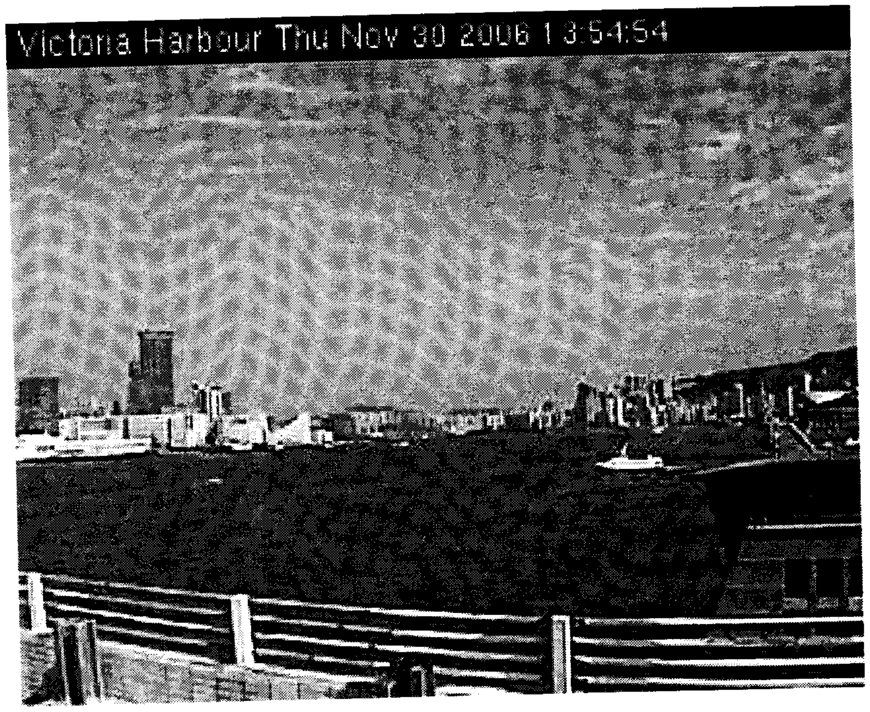

Figure 3a: Sunny

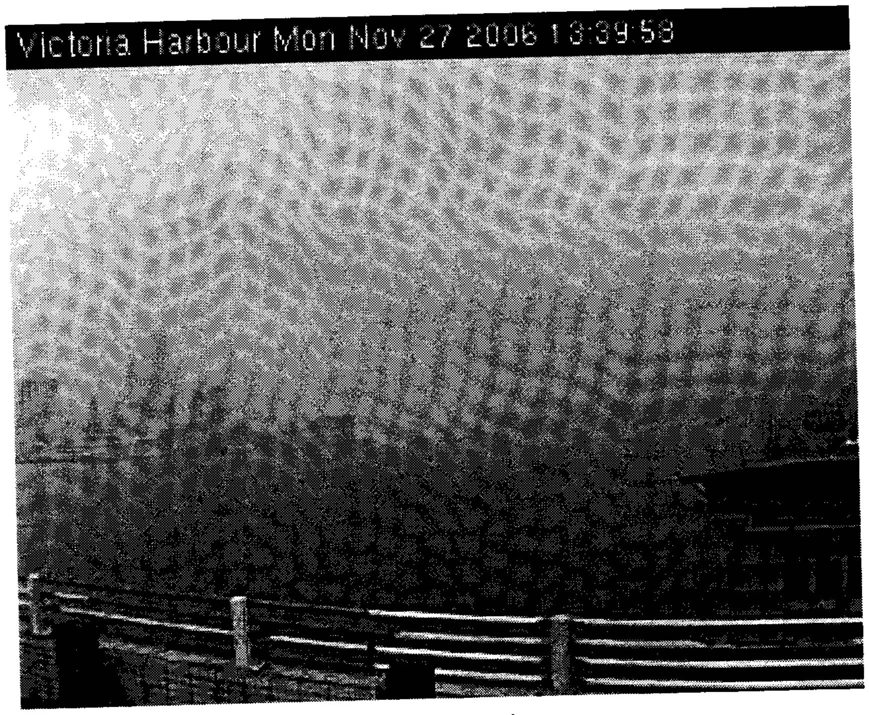

Figure 3b: Rainy

We choose 3 landmarks:

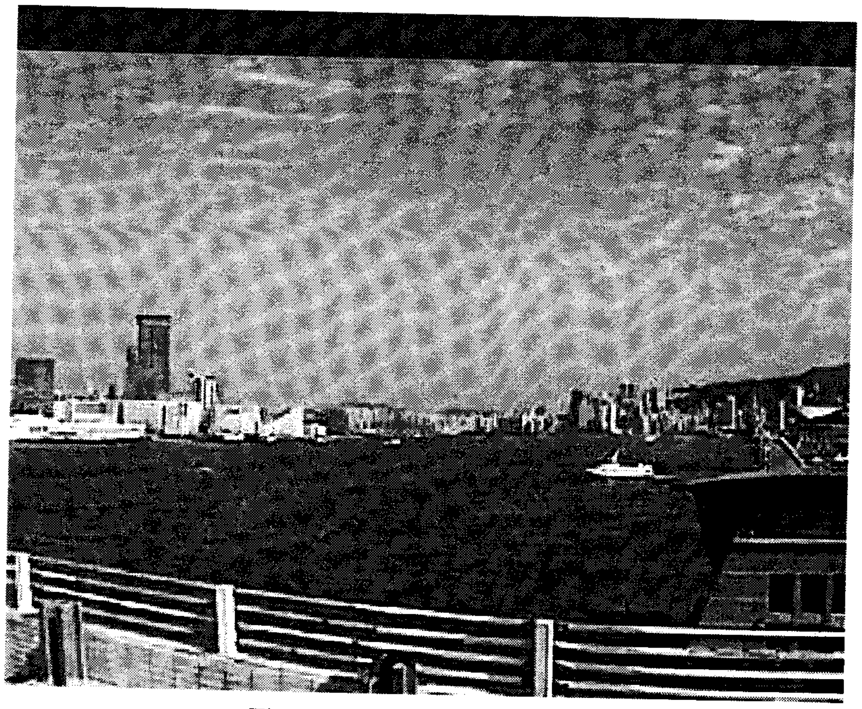

Figure 4a: building on the left

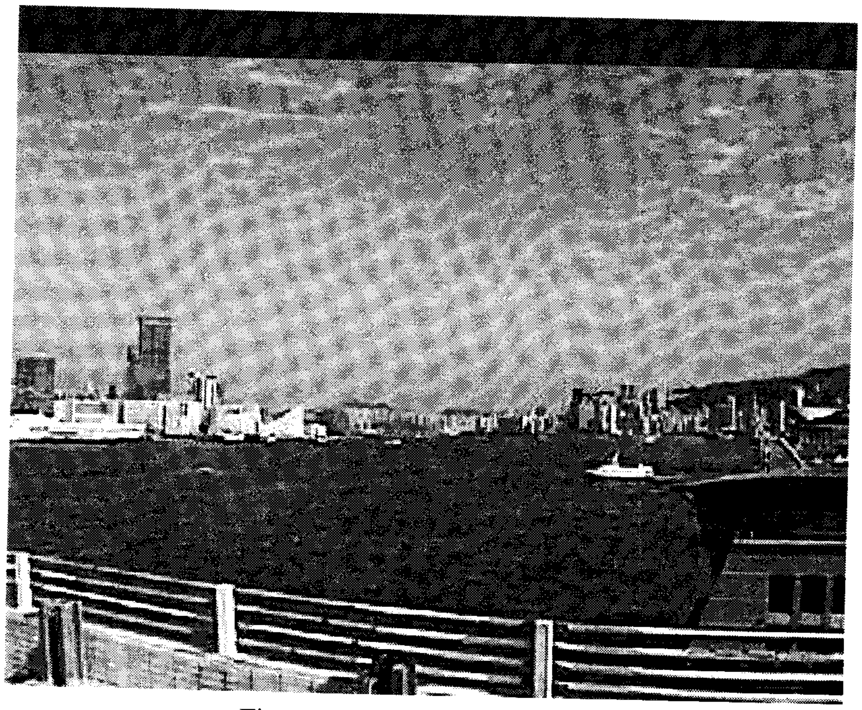

Figure 4b: building on the right

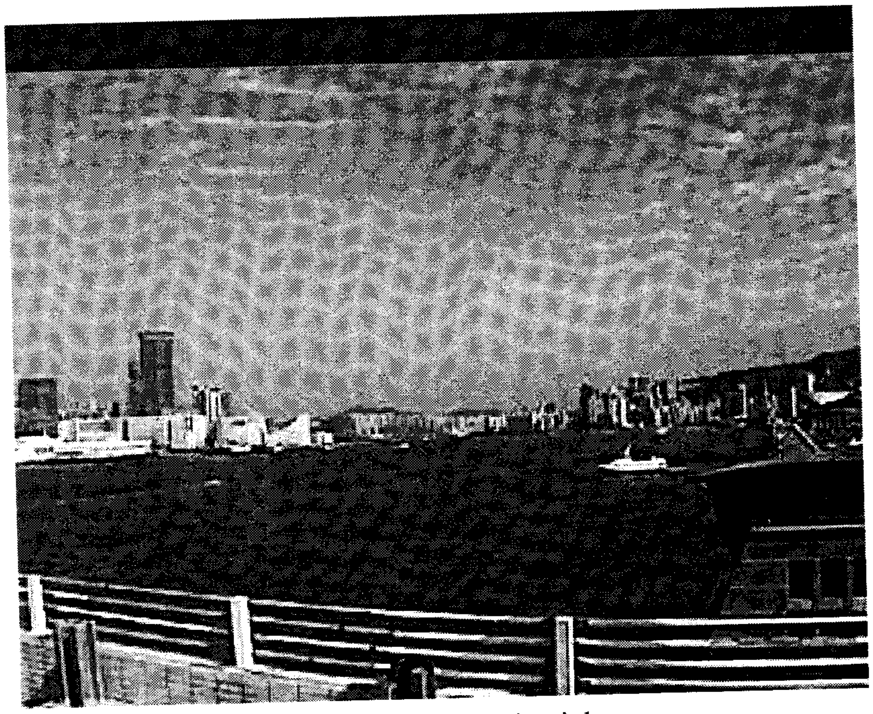
Figure 4c: hill on the right

The last example is the pictures of several buildings:

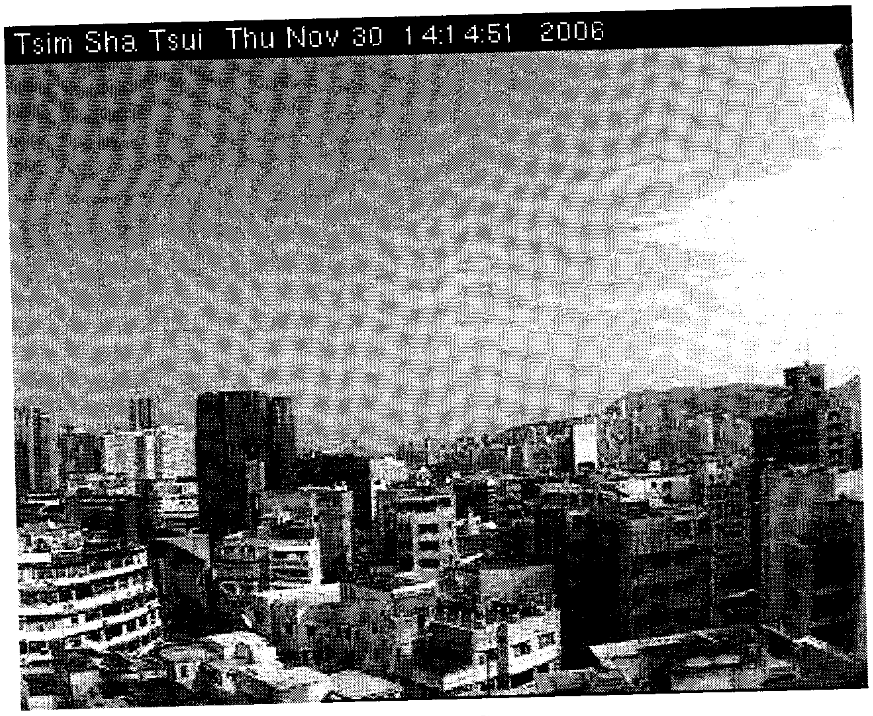

Figure 5a: Sunny

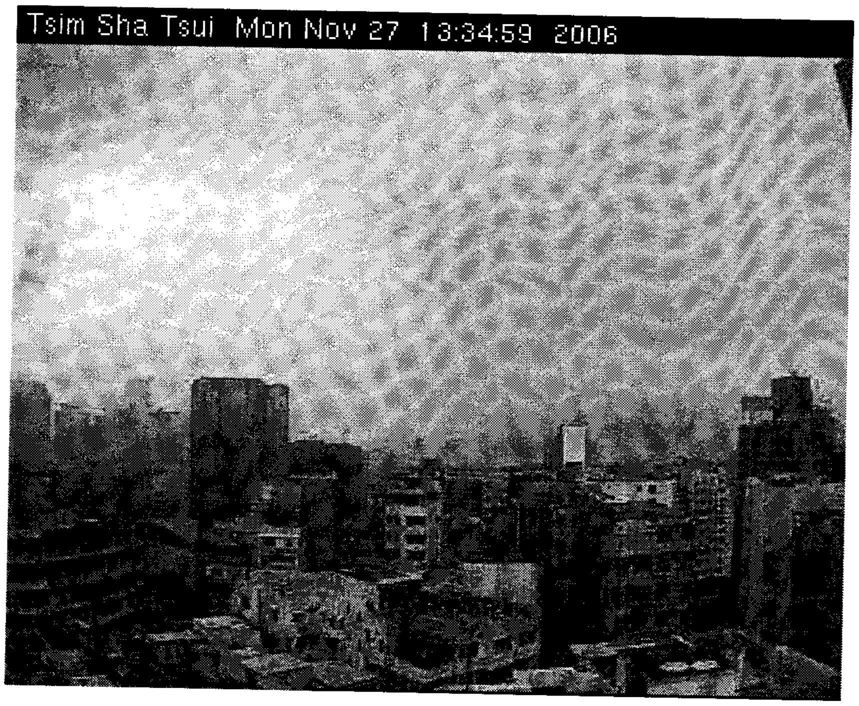

Figure 5b: Rainy

And we also choose 3 landmarks:

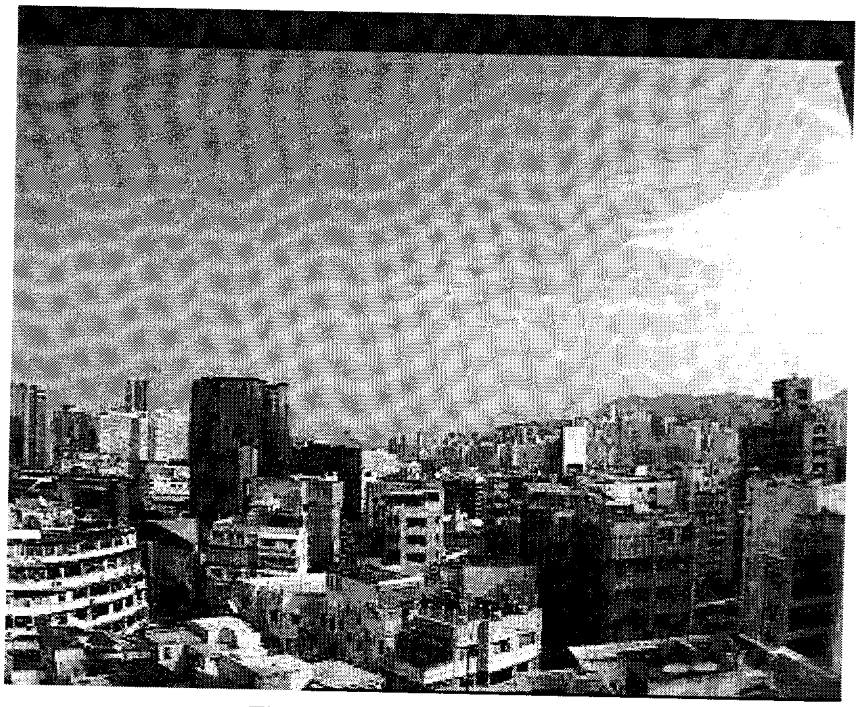

Figure 6a: building on the left

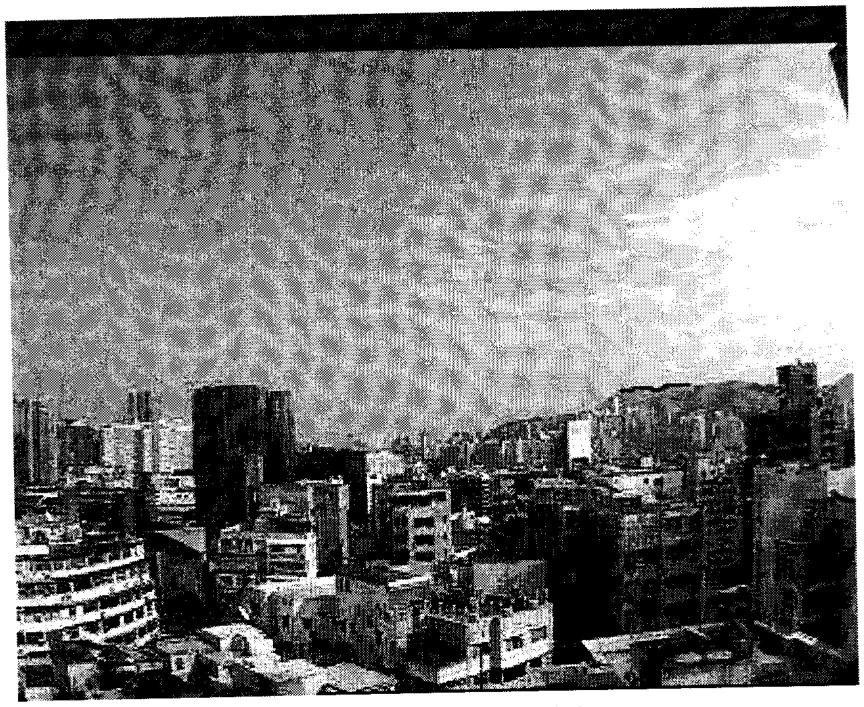
Figure 6b: hill on the right

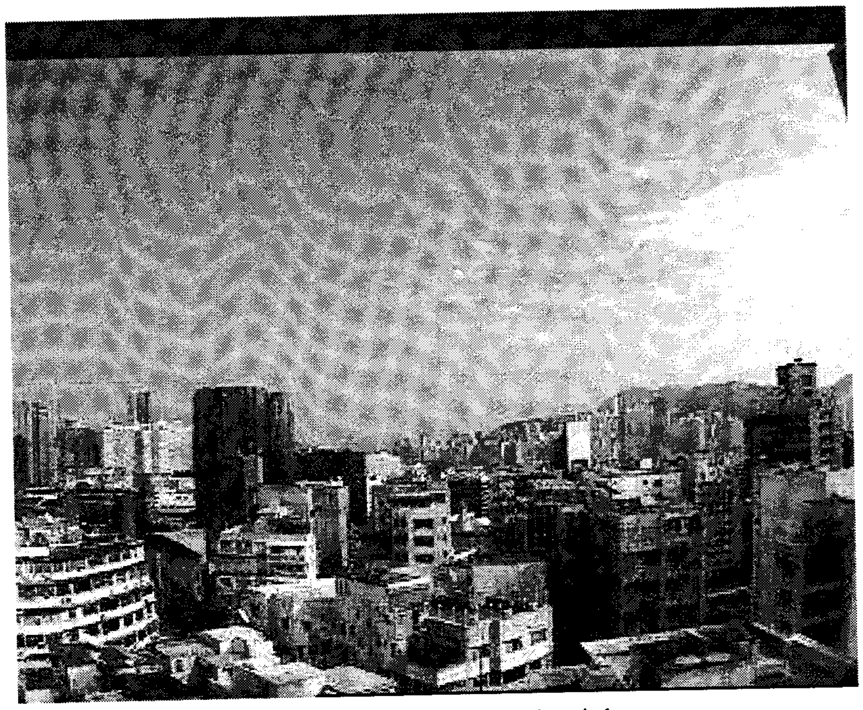
Figure 6c: building on the right

Here is the table of index values for different places different landmarks:

	Fig 2a	Fig 2b	Fig 4a	Fig 4b	Fig 4c	Fig 6a	Fig 6b	Fig 6c
sunny	685	1337	437	2413	3708	1897	746	4917
rainy	11	66	26	30	28	1794	34	2720

From the table we can see that the index values match the visibility very well.

5 Another method

Here we will give another method for determining the visibility. We will give the outline of the algorithm and the numerical results. This method is based on calculating the gradient of the color function at the boundary. So this is a method about first order derivative and the method in the previous section is about second order derivative.

The following is the outline of the algorithm:

- Mark the area around the boundary of landmarks manually with dark black color.
- Save the coordinate of the marked area into matrix M_i (element value =1 means marked and 0 means not marked).
- Compute gradient $|\nabla u|$ and get the average gradient

$$c_i = \frac{sum(M_i|\nabla u|)}{sum(M_i)}$$

- Mapping the value c_i to another value which is in the interval $(0, 1)$:

$$v_i = \frac{2}{\pi}\arctan(\beta_i c_i)$$

  Here β_i is a parameter to be chosen to make v_i not too small.

- Output the visibility by the threshold σ_i.

Next is the results for some examples. We choose one place but with 5 different weather conditions.

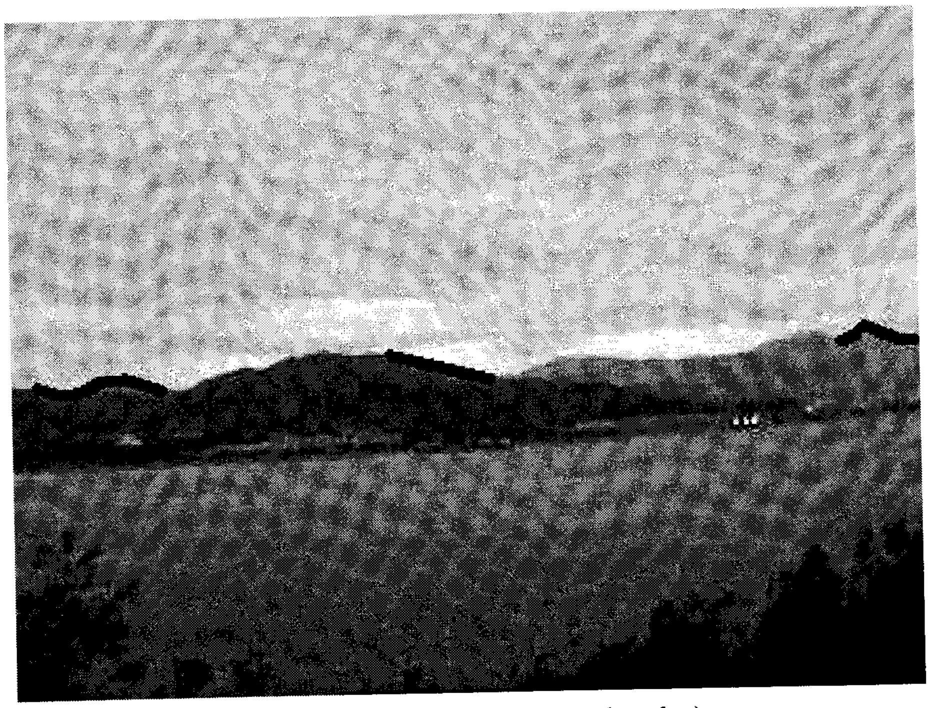
Figure 7a: Sunrise(with 3 landmarks)

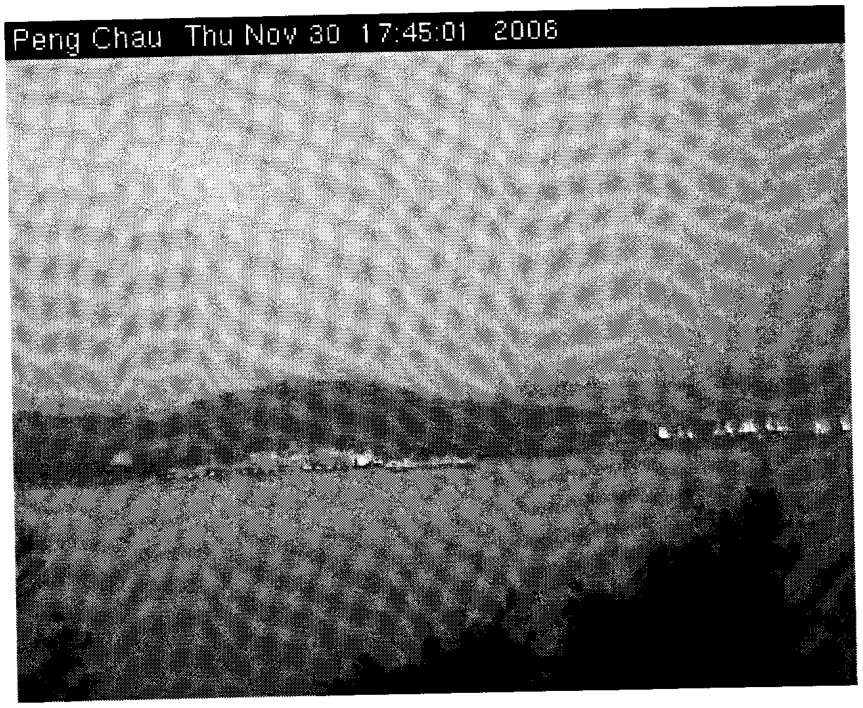

Figure 7b: Sunset

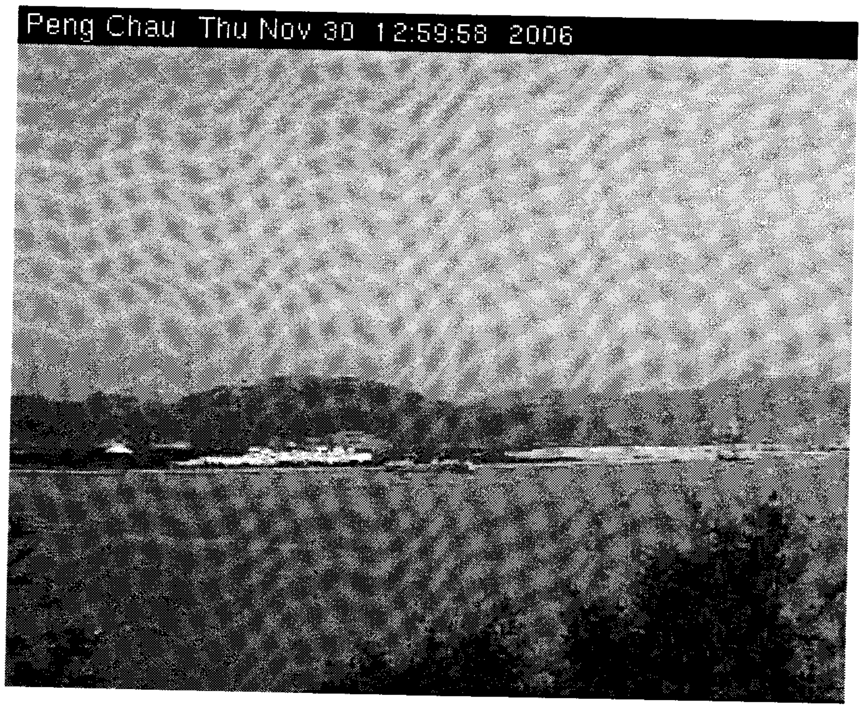

Figure 7c: Fine

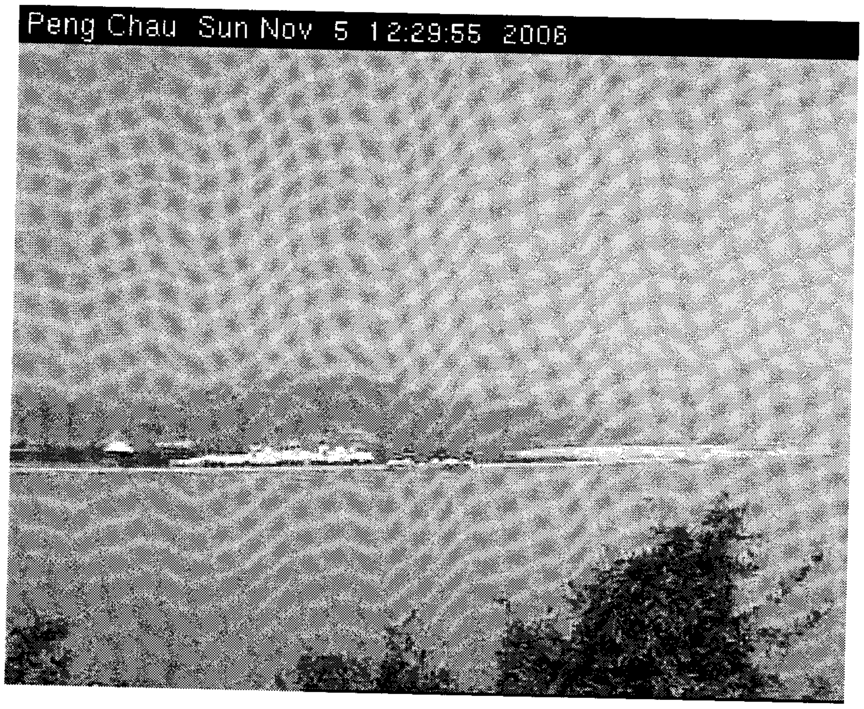

Figure 7d: Haze

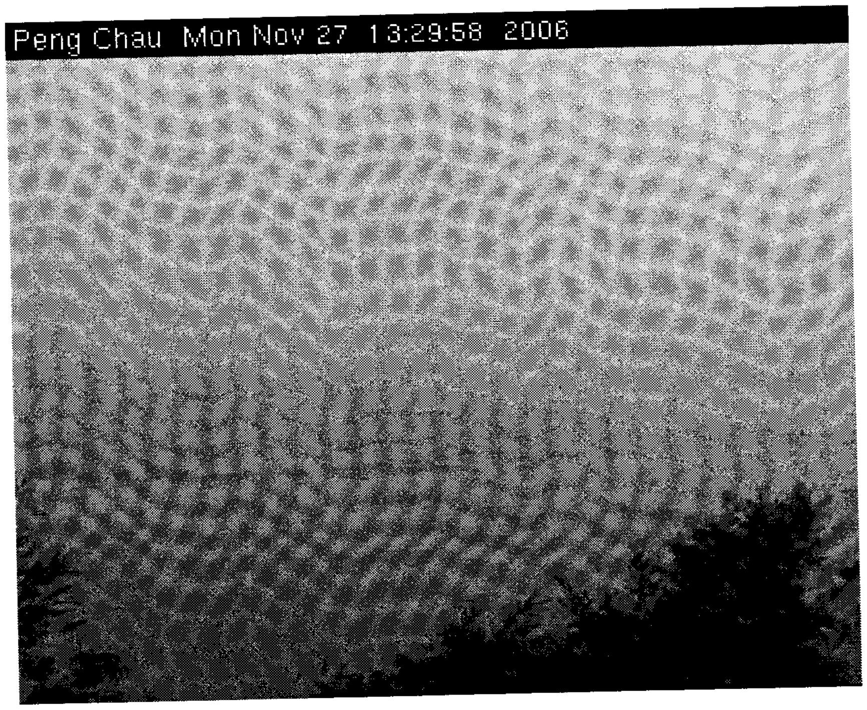

Figure 7e: Rainy

The following is the table of the index values

	Sunrise	Sunset	Fine	Haze	Rain
v_1	0.42735	0.21431	0.16485	0.22877	0.0086436
v_2	0.57249	0.27972	0.20568	0.23314	0.0095786
v_3	0.2451	0.13876	0.096531	0.083357	0.0086777

References

[1] X. Q. Wan, Y. B. Wang and M. Yamamoto, Detection of irregular points by regularization in numerical differentiation and application to edge detection, *Inverse Problems* **22** (2006), pp. 1089–1103.

[2] Y. B. Wang, X. Z. Jia and J. Cheng, A numerical differentiation method and its application to reconstruction of discontinuity, *Inverse Problems* **18** (2002), pp.1461–1476.