Eye Glint Detection and Location Algorithm in Eye Tracking

Fang Zhao, Haiying Wang, Siyang Yan

School of Information and Telecommunication Institute
Beijing University of Post and Telecommunication

[1] zf1992@bupt.edu.cn
[2] why@bupt.edu.cn
[3] yansiyang1992@126.com

***Abstract*—Eye gaze means either the gaze direction or the distance between the pupil and fixed points such as canthus or glint inside pupil. Eye-Tracking aims to record the relative offsets of this distance as time goes by. For eye-tracking, to guarantee the precision of eye glint detection and location is really important. An accurate glint detection and location algorithm based on Harris corner detection is proposed in this paper. We exploit an adaptive binarization threshold which referred to OTSU algorithm to get candidate glint connected domain via thresholding the gradient image of eye. Meanwhile, we take the advantage of improved Harris corner detection algorithm to get another candidate glint position, where Corner Reflection Function of Trajkovic corner detection is introduced to pre-filter large amount of pixels and decrease the number of pixels for Harris detection, which will reduce the computation time substantially. The result connected domain is then obtained by cross-correlating two different candidate glints.To get the accurate coordinates of glint and ignore the influence of color values, we deal the final glint with Centroid method inside which the values are all set to one. Result indicates that this algorithm achieves real-time glint detection and computes the centroid coordinates of glint, which is robust and adaptive for eye images in different environment.**

***Index Terms*—Corner Reflection Function; Harris detection; Adaptive Binarization; Centroid**

I. Introduction

The eye gaze techniques are important research domain of HCI[11, 12]. Eye gaze reflects the current situation of humanbeings and can be a powerful tools for estimating the behavioral trends. Eye-gaze-based algorithms request for high-precision detection and estimation to get the corresponding Region of Interest(ROI). The location of eye gaze requires the offset between pupil and other fixed points inside or outside the eye region such as canthus or possible glint. Procedure of eye tracking algorithm involves detection and location of eye, pupil, glint and canthus. Detection and location of eye region is a fundamental work among the whole procedure. Location of pupil is the main variation to do the estimate work,whereas canthus and glint exist as fixed points to build a relative coordinate. The direction of gaze is generally obtained by estimating the relative position of these features by mapping reference point of the glint vector and the center of the pupil. As a fixed point, glint must be detected and located with high accuracy. For now, eye tracking can be divided into two categories,namely,intrusive and non-intrusive. Classic intrusive eye tracking methods involve electromagnetic and electrode method, both of which embed contact lenses or electrode into eyes that may do potential damage to our bodies, especially to eyes.On the other hand, the non intrusive eye tracking methods mainly take the advantage of Video Oculographic(VOG)[1]. It uses monocular camera or some auxiliary equipments to get sequences of eye images and analyse it then characteristic parameters or other significant data can be obtained. Camera-based VOG may also be divided into two parts: IR camera or monocular camera. IR camera-based algorithms mainly involve Pupil Center Cornea Reflection(PCCR)[6] which gets the coordinates of pupil by differentiating light and dark image of two images of the same eye at nearly the same time when glint is the only difference between these and can be easily tracked out by connected domain extraction. non-IR camera based VOG algorithms include Hough Transform[4], gray projection operator and template matching and so on. These methods take some distinct features of eyes in gray-scale domain or gradient-scale domain. Meanwhile, for precision eye glint detection and location, available algorithms are relatively few and most of them are based on IR camera. Thus, no-IR camera based glint detection algorithm is of great value for researching afterwards recently smartphone is popular and most of these equipments have no IR equipments. This paper proposes a method for fast and robust eye glint detection and location which is running in real time. We employee a normal web-camera without IR assistant. This means we can adapt our glint detection algorithm with any tool equipped with a single camera like smartphone or common web-camera inside the laptop. We refine our main innovation points as follows:

1. we designed an adaptive binarization algorithm referred to OTSU[5, 7, 9] according to the characteristic of glint inside the pupil. This modified OTSU can deal with different eye

scenes and find out the targeted glint robustly.

2. A pre-filter procedure is introduced before Harris corner detection[2, 15]. This will reduce the candidate pixels suitable for the concept of corner in Harris substantially, which decreases the time for Harris computation in a large amount.
3. To ease the influence of pixel color value,we do a normalization process with the extracted glint connected domain. This will correct the glint coordinate offset caused by the change of light or other factors.

II. Glint Detection and Location Algorithm

The procedure of proposed glint detection and location algorithm are detailed in this section. For a input color image of eye, we first convert it into gray scale and apply Sobel operator to get the gradient image. Result image will be the basic image we take for subsequent detection. Before our algorithm, we segment gradient image into three parts based on the distribution of left and right eye in eye image. The first and third part of image will be taken for detecting while the second part dropped for no eye region. This step will reduce the pixels ready for Harris detection and can make use of multi-thread to do the job simultaneously. For following processes, take left image as an example, we use designed adaptive threshold algorithm to get a proper threshold for binarization. With this threshold, we can obtain a binary image in which glint area should be a connected domain[3] on account of the distribution of glint, that is why we take a step of connected domain extraction to exclude most that can not be the glint. Candidate connected domains are then stored for further handle. On the other hand, we utilize Harris corner detection method to do detection on the gradient image, before which a pre-filter procedure called corner reflection function will be introduced to preclude most of pixels no suitable for Harris corner concept. This step will get other candidate glint positions. Cross-correlating these positions with former candidate connected domains, we can get the real glint connected domain. Then we normalize this domain with 1 to eliminate the influence of color value and Centroid method will be applied to get the final coordinate of glint.

A. *Adaptive Image Binarization With Gradient Image*

Among an eye image, the average brightness of glint inside the pupil is higher compared to other part of eye, and its percentage of pixels is far less than others, thus we consider utilizing a binarization method to threshold the gradient image, after that we do a connected domain extraction work to get all possible glint candidates. To make it accurate and remove some futile candidates, we take a special judgement threshold according to the pixels of connected domain. Among all this flow,the accuracy of adaptive binarization threshold will make a great influence on glint as bigger threshold will compress the edge of glint or even disappear it while smaller introduce too many candidates which increase the computation for following handle. Classic image binarization algorithms in image process domain mainly involves OTSU, Bernsen and Cyclic threshold algorithm[10]. Take time-consuming and algorithm complexity into consideration, OTSU is a better algorithm which conduct a promising result and can be executed in real time. Whereas, when the difference between average value of foreground and background is huge, OTSU can not do a proper segmentation for these image, so we designed a simple threshold method which conquered this defect and is more adaptive for target object especially for glint or others whose average brightness are high.

For a M×N image, we count the distribution of all different pixels first of all. Assume that the corresponding number of pixels is $\vartheta\left(\varepsilon\right)$, ε[0,255] when pixel is set to ε. If the threshold we need is k, k[0,255], then we can divide all pixels into two part with this special threshold,namely,lo=[0,k] and hi=(k,255], where lo indicates pixels no bigger than k and hi the inverse. With above notations, we now begin to compute separate total pixels of two parts:

$$\varphi_{lo} = \mathrm{sum}\left(\mathrm{lo}\right) = \sum_{\varepsilon=1\ldots k} \vartheta\left(\varepsilon\right) \tag{1}$$

$$\varphi_{hi} = \mathrm{sum}\left(\mathrm{hi}\right) = \sum_{\varepsilon=k+1\ldots 255} \vartheta\left(\varepsilon\right) \tag{2}$$

φ means the sum of pixels and we get the average gray value of lo and hi as follows:

$$\mathrm{avg}_{lo} = \frac{\sum_{\varepsilon=1\ldots k} \vartheta\left(\varepsilon\right) * \varepsilon}{\varphi_{lo}} \tag{3}$$

$$\mathrm{avg}_{hi} = \frac{\sum_{\varepsilon=k+1\ldots 255} \vartheta\left(\varepsilon\right) * \varepsilon}{\varphi_{hi}} \tag{4}$$

if avg_{lo} and avg_{hi} have been calculated, we then utilize it to get the sum of variance of these two parts:

$$\delta^2{}_{lo} = \sum_{\varepsilon=0}^{\varepsilon=k} \left(\varepsilon - \mathrm{avg}_{lo}\right)^2 \tag{5}$$

$$\delta^2{}_{hi} = \sum_{\varepsilon=k+1}^{\varepsilon=255} \left(\varepsilon - \mathrm{avg}_{hi}\right)^2 \tag{6}$$

$\delta^2{}_{lo}$ implies the sum of variance of low and $\delta^2{}_{hi}$ the high,while this factor means fluctuation range in contrast to the average gray value in corresponding region. Moreover, in order to gain a comparative appropriate k, we define a special criterion like:

$$\arg\nabla\delta^2{}_{tmax} = \frac{\left(\delta^2{}_{hi} * \varphi_{hi} - \delta^2{}_{lo} * \varphi_{lo}\right)}{M \times N} \tag{7}$$

$\arg\nabla\delta^2{}_{tmax}$ is the relative difference between foreground and background relate to current k. Thus we can get the final result by another threshold:

$$\arg\nabla\delta^2{}_{max} = \begin{cases} \arg\nabla\delta^2{}_{tmax} & \arg\nabla\delta^2{}_{tmax} > \arg\nabla\delta^2{}_{max} \\ \arg\nabla\delta^2{}_{max} & \arg\nabla\delta^2{}_{tmax} \le \arg\nabla\delta^2{}_{max} \end{cases} \tag{8}$$

we take a step of iteration for k from 0 to 255 and obtain the biggest $\arg\nabla\delta^2{}_{max}$. The k that deduces the final biggest result will be selected as the threshold for our gray image binarization, then a binary image will be generated by this process. According to above representation about threshold method,we summarize it as four steps:

1. Count the distribution of all pixels and pick a value of k which will separate the whole gradient image into two part:foreground and background;
2. Sum up the pixels number of two parts and calculate their average value and get the sum of variance;
3. Apply our custom criterion $\arg\nabla\delta^2{}_{tmax}$. Calculate it and adjust it according to corresponding values;
4. If all k have been calculated, we take the k mapped to the final $\arg\nabla\delta^2{}_{max}$ as our threshold and finish algorithm, otherwise, increment k and continue the iteration until all k have been traversed.

B. Coarse Glint Location with Pre-filtered Harris Detection

As the glint has the characteristics of little pixels, high brightness and distribute like punctate or patchy, after doing comparison with classic corner detection algorithm and considering glints feature, we take Harris corner detection as our detection method as it may be sensitive to feature with high edge change in gradient image, which is the most obvious feature with glint.

Harris corner detection is one of the most common corner detection methods when detecting features of a image. It does a great improvement for the deficient of Moravec method in direction dependence, filter window and edge response[14], which drive Harris as a robust and accurate corner detection method. Whereas, a Gaussian window is introduced when we filter candidate corner, which results in a sharp increase in time and a low efficiency. To overcome it, we introduced a Corner Reflection Function(CRF) in Trajkovic corner detection[8, 13] before doing Gaussian filter to pre-filter a large number of pixels which is not in consistent with the concept of corner defined in Harris. This will decrease the whole computation as CRF is fast while Gaussian filter is slow and time-consuming.

The Definition of CRF in Trajkovic is:

$$\Re\left(\mathrm{x},\mathrm{y}\right)=\min_{x\in(0,1)}\left(\ell_1\left(x\right),\ell_2\left(x\right)\right) \tag{9}$$

where x is the parameter defines the position of pixel. Thus, we can deduce a CRF as:

$$\ell_1\left(x\right)=\left(\mathrm{I}_P-\mathrm{I}_N\right)^2+\left(\mathrm{I}_{P^,}-\mathrm{I}_N\right)^2 \tag{10}$$

$$\ell_2\left(x\right)=\left(\mathrm{I}_Q-\mathrm{I}_N\right)^2+\left(\mathrm{I}_{Q^,}-\mathrm{I}_N\right)^2 \tag{11}$$

where N is the center point and $I_P(I_Q)$ is the corresponding value for P(Q). $P^,$,$(P^,)$imply the symmetric point of P(Q) relative to N. We can get I_P,I'_P,I_Q,I'_Q by:

$$\mathrm{I}_P=\left(1-x\right)\mathrm{I}_A+x\mathrm{I}_B \tag{12}$$

$$\mathrm{I}_{P'}=\left(1-x\right)\mathrm{I}_{A'}+x\mathrm{I}_{B'} \tag{13}$$

$$\mathrm{I}_Q=\left(1-x\right)\mathrm{I}_{A'}+x\mathrm{I}_B \tag{14}$$

$$\mathrm{I}_{Q'}=\left(1-x\right)\mathrm{I}_A+x\mathrm{I}_{B'} \tag{15}$$

where A/B is the point in horizon/vertical axis, A/B is symmetric point of A/B. With above notations, we new can define a CRF before Gaussian filter is executed in Harris:

$$\mathrm{O}\left(x,y\right)=\begin{cases}0,\ \Re(x,y)<t_1\\1,\ \Re(x,y)\geq t_1\end{cases} \tag{16}$$

where t_1 is a threshold we define for including or excluding pixels for a map considering as candidate corner pixels for Gaussian filter. If the value through CRF is bigger than our threshold, corresponding pixels will be added into map or else dropped as useless pixels which will never be taken into filter work.

After the CRF is introduced, the whole corner detection flowchart is as follows:

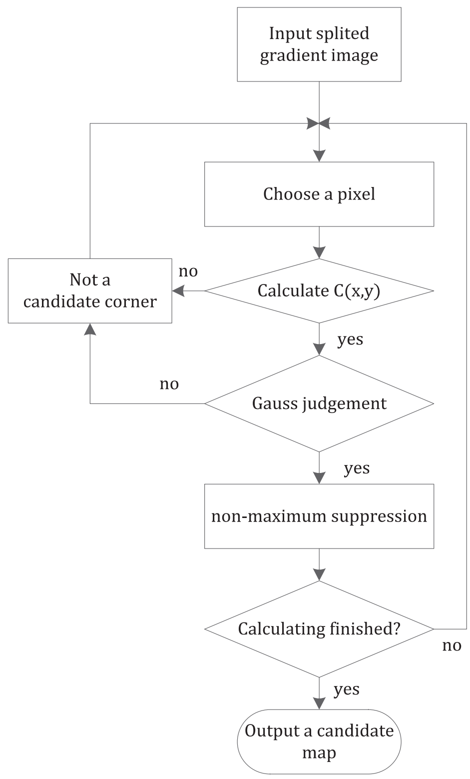

Fig. 1. Flow chart of pre-filtered Harris detection

We compute the gradient of origin gray image and get an image with glint where the feature and contrast of glint is more visible compared to origin image or gray image. Then we execute pre-filtered Harris detection algorithm to pick out all possible candidate corners. Because of the sensitivity of Harris detection for features like glint, all candidates relate to glint will be detected before all and thus we can simply select the first 4 to 10 candidate corners, in which glint region will be included surely.

The coordinate we get via above operation is not exactly the accurate center of glint, what we can be in sure is that this coordinate must be inside the glint. Thus we can take this as a cross-correlation criterion with former-obtained candidate glint regions and get the final connected domain of a glint.

C. Normalized Glint Center Location

There are some gray distribution limitations when using gray image or gradient image to obtain the center of an object, especially for non-rigid objects or an image with irregular gray distribution. To get an accurate coordinate of glint with obtained glint region, we introduce a normalization step to eliminate it's effect. For all pixels we assume its value to be 1 so that the color of glint will be white. Following this method, we describe the classic centroid method as:

$$\mathrm{x} = \frac{\sum_{i=0}^{rows-1} \sum_{j=0}^{cols-1} i * P(i,j)}{\sum_{i=0}^{rows-1} \sum_{j=0}^{cols-1} P(i,j)} = \frac{\mathrm{sum}(i)}{count(i)} \tag{17}$$

$$y = \frac{\sum_{i=0}^{rows-1} \sum_{j=0}^{cols-1} j * P(i,j)}{\sum_{i=0}^{rows-1} \sum_{j=0}^{cols-1} P(i,j)} = \frac{sum(j)}{count(j)} \tag{18}$$

To say as another form, the horizon coordinate of glint center equals to the average value of all horizon coordinate and vertical coordinate correspondingly. With this simple computation, we get an comparatively accurate coordinate of glint center.

III. Experiment Result

The source images we used in our experiment are all captured by a web-camera of 1.3 megapixel. The size of source image is 1280×1024, and extracted eye images are all resized into 360×100 to make all test examples run with nearly the same origin parameter. The develop environment we used in this experiment is Visual Studio 2013 with OpenCV configured. Among all the experiments, we analysed the differences between origin Harris corner detection and pre-filtered detection algorithm in time and accuracy, and the effect on the same eye image of our adaptive threshold and OTSU. At last we applied simple centroid and output the final results.

A. Result for Pre-filtered Harris Corner Detection

The results for pre-filtered Harris corner detection are show in Figure 2.

Result shows that the pre-filtered Harris corner detection algorithm has a better accuracy and less ambitious than original algorithm. To make a detail comparison between original and our method, we deal a bundle of eye images with two kinds of methods and output the time-consuming results as Table I:

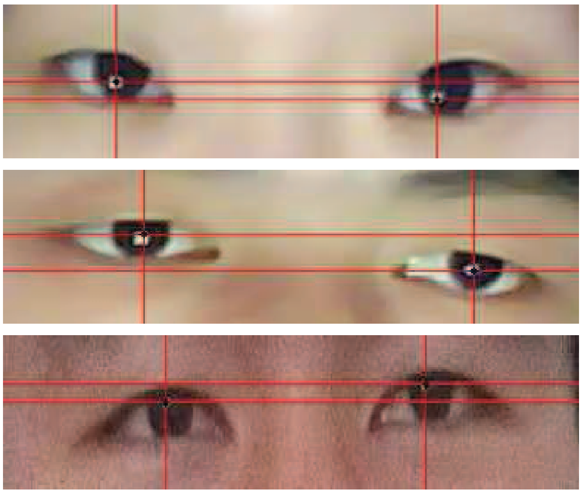

Fig. 2. Result of glint detection employed improved Harris corner detection

TABLE I
Time consumption of two different Harris corner detection

Pic Num	Origin Left Eye	Pre-filtered Left Eye	Origin Right Eye	Pre-filtered Right Eye
1	0.577	0.163	0.574	0.142
2	0.655	0.188	0.686	0.192
3	0.577	0.153	0.577	0.142
4	0.686	0.173	0.702	0.181
5	0.670	0.171	0.654	0.163
6	0.670	0.179	0.690	0.185
7	0.686	0.171	0.686	0.174

B. Result for Adaptive Binarization Threshold Algorithm

Picture (b) to (e) in Figure 2 show the comparison results after implementing our adaptive threshold algorithm and classic OTSU for image binarization. Two algorithms deal with the same image and we can observe from the result image that compared to OTSU, our adaptive threshold has a better effect on images with diverse environment. For the same glint in exactly the same image, our algorithm leave a more detailed contour for object like glint, which is essentially the key to get a more precision coordinate from a small connected domain. Picture (f) and (g) show corresponding result for connected domain extraction after binary image has been obtained, from which we can get that our method actually achieves a better persistence of contour for glint whereas there are some noisy candidate glint connected domain that can not be filtered our with our method. That is why we have to introduce former Harris detection to get another candidate position for glint.

After connected domain and candidate position for glint have all been extracted, we can cross-correlate two results and get final result of glint. Then our centroid method can be implemented thus coordinate will be calculated. This part will not be presented here.

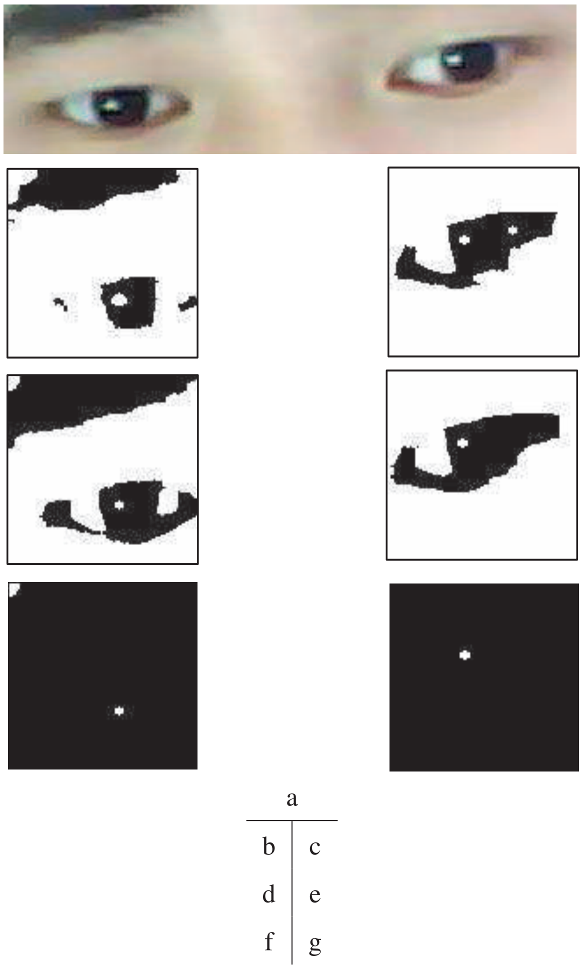

Fig. 3. Adaptive Threshold Result
(a)original eye;(b)our method with right eye;(c) our method with left eye;(d)OTSU with right eye;(e)OTSU with right eye;

IV. Conclusion

To extract possible glint inside pupil accurately and quickly, we designed a cross-correlation algorithm which combines adaptive threshold algorithm for image binarization and pre-filtered Harris corner detection according to the prior knowledge about the feature of glint. With experiments we can make a conclusion that our algorithm can extract the coordinate of glint inside pupil in nearly all kinds of background and light condition with noisy candidate glints eliminated by pre-filtering. While most of possible glint have been extracted from eye successfully, there are still conditions where light tends to be extreme or the contrast between glint and pupil is nearly ignorable, which results in failure extraction for glint. This is the direction of our future work.

V. Acknowledgement

This work was supported by Beijing Key Laboratory of Network System and Network Culture (Beijing University of Posts and Telecommunications).

References

[1] R Becker, TH Krzizok, and Heiko Wassill. Use of pre-operative assessment of positionally induced cyclotorsion: a video-oculographic study. *British journal of ophthalmology*, 88(3):417–421, 2004.

[2] Chris Harris and Mike Stephens. A combined corner and edge detector. In *Alvey vision conference*, volume 15, page 50. Citeseer, 1988.

[3] Parminder Kaur and Anupama Gupta. Contour detection of gradient images using morphological operator and transform domain filtering. In *Computational Intelligence & Communication Technology (CICT), 2015 IEEE International Conference on*, pages 107–111. IEEE, 2015.

[4] DJ Kerbyson and TJ Atherton. Circle detection using hough transform filters. In *Image Processing and its Applications, 1995., Fifth International Conference on*, pages 370–374. IET, 1995.

[5] Dong Ju Liu and Jian Yu. Otsu method and k-means. In *Hybrid Intelligent Systems, 2009. HIS'09. Ninth International Conference on*, volume 1, pages 344–349. IEEE, 2009.

[6] Carlos H Morimoto and Marcio RM Mimica. Eye gaze tracking techniques for interactive applications. *Computer Vision and Image Understanding*, 98(1):4–24, 2005.

[7] Nobuyuki Otsu. A threshold selection method from gray-level histograms. *Automatica*, 11(285-296):23–27, 1975.

[8] Yujing Qiao, Yanchao Tang, and Junshi Li. Improved harris sub-pixel corner detection algorithm for chessboard image. In *Measurement, Information and Control (ICMIC), 2013 International Conference on*, volume 2, pages 1408–1411. IEEE, 2013.

[9] Zhong Qu and Li Zhang. Research on image segmentation based on the improved otsu algorithm. In *Intelligent Human-Machine Systems and Cybernetics (IHMSC), 2010 2nd International Conference on*, volume 2, pages 228–231. IEEE, 2010.

[10] Prasanna K Sahoo, SAKC Soltani, and Andrew KC Wong. A survey of thresholding techniques. *Computer vision, graphics, and image processing*, 41(2):233–260, 1988.

[11] V. Scholl and A. Gerace. Removing glint with video processing to enhance underwater target detection. In *Image Processing Workshop (WNYIPW), 2013 IEEE Western New York*, pages 18–21, Nov 2013.

[12] A. Sharma and P. Abrol. Comparative analysis of edge detection operators for better glint detection. In *Computing for Sustainable Global Development (INDIACom), 2015 2nd International Conference on*, pages 973–977, March 2015.

[13] Miroslav Trajković and Mark Hedley. Fast corner detection. *Image and vision computing*, 16(2):75–87, 1998.

[14] Jun Zhang and Jinglu Hu. Image segmentation based on 2d otsu method with histogram analysis. In *Computer Science and Software Engineering, 2008 International Conference on*, volume 6, pages 105–108. IEEE, 2008.

[15] Wan-jin Zhao, Sheng-rong Gong, Chun-ping Liu, and Xiang-jun SHEN. Adaptive harris corner detection algorithm. *Computer Engineering*, 10(5):212–215, 2008.