International Journal of Signal Processing, Image Processing and Pattern Recognition
Vol.7, No.1 (2014), pp.393-404
http://dx.doi.org/10.14257/ijsip.2014.7.1.36

HUD Image Vibration Detection on Improved Edge Detection and Corner Extraction

Rong-gui Ma[1], Ting Cao[2,*] and Weixing Wang[3]
[1]*School of Information Engineering, Chang' an University, Xi'an 710061, China, rgma@chd.edu.cn*
[2]*School of Information Engineering, Chang' an University, Xi'an 710061, China, wellcaoting@163.com*
[3]*School of Information Engineering, Chang' an University, Xi'an 710061, China, wxwang@chd.edu.cn*

Abstract

HUD (Head Up Display) image vibration detection requires the real-time performance. In this paper, after analyzing different algorithms for the movement estimation of the HUD, a new vibration detection algorithm is studied based on feature extraction. In the algorithm, after image smoothing, an improved Canny operator is proposed for the edge detection, then a proposed corner detection method is applied to extract the feature points on edges. The testing results prove that the studied algorithm works in real-time and with high precision.

Keywords- *Head Up Display, vibration detection, feature extraction*

1. Introduction

Head Up display (HUD) is the transparent display that presents data without requiring a user to look away from his/her usual viewpoints. The origin of the name stems from a pilot being able to view the information with the head positioned up and looking forward, instead of angled down looking at lower instruments. As an example, a kind of HUDs is shown as in Fig1. Although they were initially developed for the military aviation, HUD is now widely used in the commercial aircraft, automobiles, video game and other applications. HUD can help pilot keep focus and avoid losing of situation awareness.

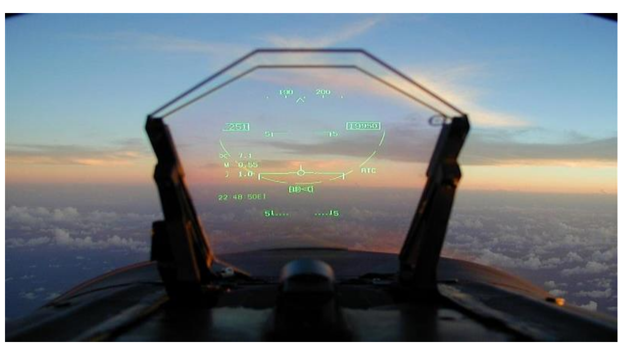
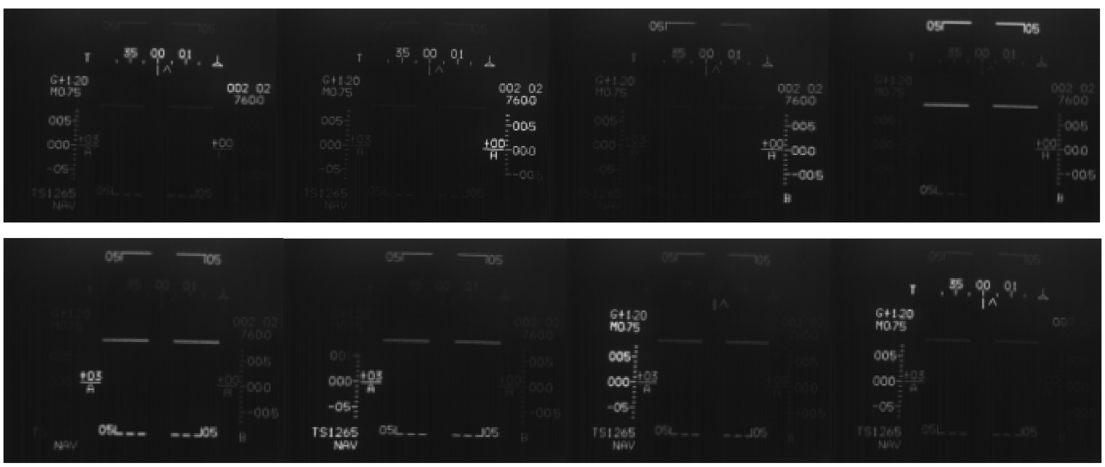


Figure 1. An example of HUD

* Corresponding Author: Ting Cao, School of Information Engineering, Chang' an University, Xi'an 710061, China.

However, in reality HUD's vibrations are inevitable and unpredictable. It will directly affect the pilot's visual results, which leads to the subjective misjudge easily. Therefore, the study of the HUD image vibration detection in real-time is important in applications.

In order to simulate the vibrations of reality, the HUD imaging systems are usually fixed on a platform which vibrates random within a given vibration frequency. Based on the images captured from a camera, one can detect the vibration. In this paper, HUD vibration frequency ranges from 10Hz to 2000Hz, and mainly concentrates between 300Hz and 1000Hz. The reaction time of human eye is 10Hz (0.1s). In order to capture every image's vibrations, the frequency of the camera is selected by 200Hz, which means the camera can take 500 images per second. However, the illumination which is made by eight lights is not continuous uniform at this speed. There is only one light lighting in every single image frame and every of lights will be lighted again according to the order from left to right and top to bottom. That means every sequence of eight images is a cycle. So, before the vibration detection, eight images should be averaged as an output image. As an example, the eight sequence images are shown as in Figure 2.



(a) Eight sequence images from left to right and top to bottom.

(b) Output image after image averaging

(c) Vibration image

Figure 2. Eight Sequence Images and their Averaging Result

The problem of digital image stabilization has been widely investigated in [1-3]. Previous work can be classified in two main categories: direct methods [4] and feature-based methods [5].The direct methods mainly consider image intensity values. The feature-based methods have gained larger consensus for their good performances, but the feature computation step can be very time consuming.

For the purpose in this study, the feature-based methods are applied in the proposed vibration detection algorithm. Due to the original image is given, the time consuming of feature point matching is less. What's more, the algorithm must be properly designed to work in real time.

So, based on the proposed vibration detection algorithm one can get all the feature points' vibration offsets and the simulation figures of the vibration. In addition, the whole image vibration can be estimated.

2. Motion Estimation Algorithm

In general, the motion estimation is the main task for the image vibration detection [6-10]. Its accuracy and speed could greatly influence the overall system's performance. Under many conditions, if it is not properly managed, it can degrade its performances, such as illumination changes, moving objects, image blur, and periodic patterns, *etc*.

The relationship between two adjacent images is the function of the combined action of the previous frame and the offset. The function is shown as follows:

$$\mathrm{Im}\,g(i+1) = f[\mathrm{Im}\,g(i), g(dx,dy)] \tag{1}$$

Where, $\mathrm{Im}\,g(i)$ denotes the previous frame, $g(dx,dy)$ denotes the offset, and $\mathrm{Im}\,g(i+1)$ denotes the next frame.

There are many algorithms about the motion estimation, such as block matching, gradation projection, and feature matching, *etc*., [11, 12]. As literature review, it can be found that the different algorithms have their own advantages and disadvantages, and some examples are shown in Table 1.

Table 1. Advantages and disadvantages of Motion Estimation Algorithms

Algorithms	advantages	disadvantages
Block Matching,	algorithm is simple little loss of information	more calculation consuming long time inefficiency poor interobserver agreement
Gradation Projection	high calculation speed less calculation the performance is stable,	Only for translation motion Request the image gradation change higher lower accuracy
Feature Matching	high calculation speed less calculation easy to get feature can use for rotary object	lower accuracy

From Table 1, it can be found that the algorithm of Feature Matching has more advantages than other algorithms. The only question is how to use other algorithms to improve the matching accuracy if the Feature Matching algorithm is selected to use.

3. Improved Feature Extracting Algorithm

3.1. Feature Point Detection

The point feature [13-14] in the Feature Matching method is the widely used feature. Compared with the line features and area features, the point feature is easily extracted, and it has the characteristics of less calculation, high accuracy, and high steadiness. The principle of the point feature is to separately select some feature points in the reference frame image and the current detecting image, then to match the selected feature points, and finally to calculate the movement vector.

The commonly used methods of the feature point detection include corner detection**,** interest point detection, and edge detection [8]. Both of corner detection and interest point detection are used for detecting the whole image, while edge extraction is applied for selecting the feature point from the edge points, so it performs in real-time. Since the HUD image vibration detection requires the real-time performance, the edge detection is selected to use.

3.1.1. Edge Detection Method

Edges are the pixels that have different intensity value comparing with its neighborhoods. The theory of edge detection is using the image’s first derivative extremum or the second derivative zero crossing information to judge the edge points. Hence, the selection of the edge points as feature points can guarantee the high accuracy and robustness. The common edge detection operators include Sobel, LOG and Canny operators [15-17].

3.1.2. Improved Edge Detection Method

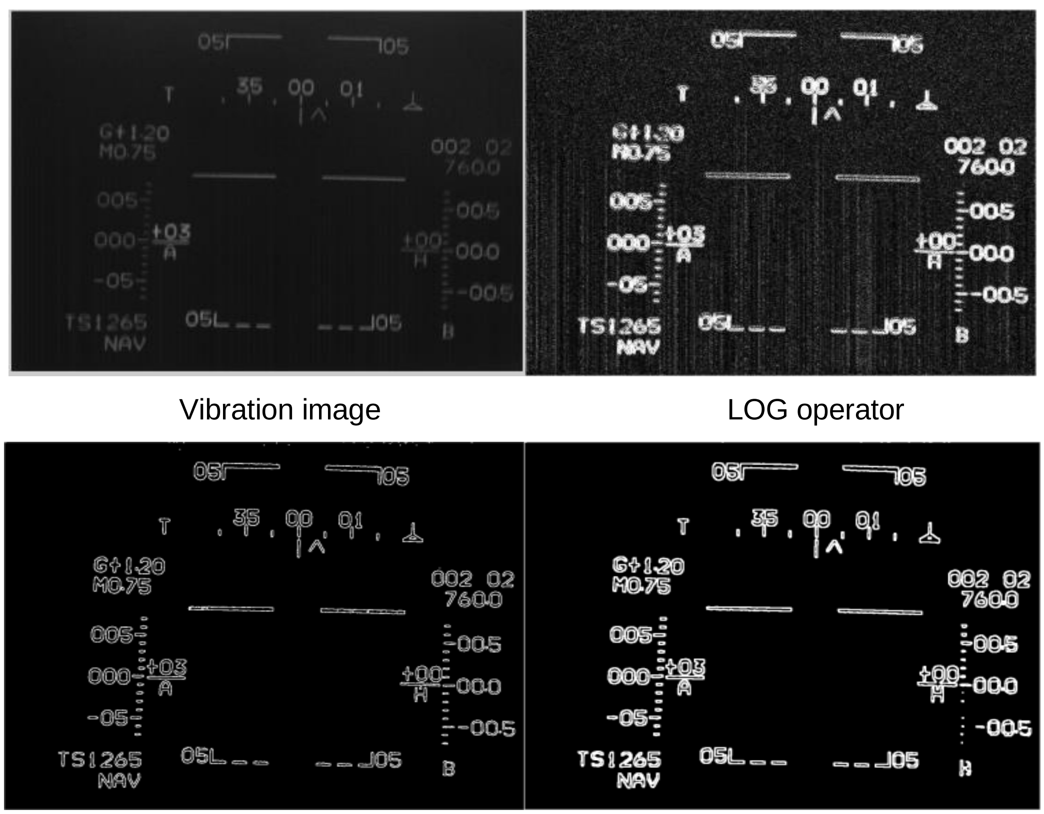
As one of the standard edge detection methods, the Canny operator presents three constraints for an edge matching filter, which includes good detection, good localization, and low multiplicity of false detection. Compared with the first derivative-based edge detectors, such as Sobel operator, the LOG operator is the second derivative operator that is used to detect the zero-crossings of image intensity. Canny also use gradient operator which makes use of Gaussian for finding gradient of an image in horizontal and vertical directions and apply a high threshold and a low threshold to discover an edge.

Consider the HUD images studied in this paper, a new edge detection algorithm is proposed based on Canny operator, and it is improved in the flowing two aspects to execute the edge detection.

Firstly, in the step of Canny operator, the gradient magnitude is used to set two thresholds of high and low to obtain edge points. It is important to choose the high threshold and low threshold correctly. However, the gradient magnitude calculated in formula (3) is easy to mix up low intensity edge and noise. The low intensity edge has the characteristics of gradient magnitude and gradient direction. The noise only has the characteristic of gradient magnitude. So a new formula (5) is proposed to calculate gradient magnitude. It integrates gradient direction with the calculation of gradient magnitude, which can be the ground for gradient magnitude in edge detection.

Secondly, the normal method for choosing two threshold values is by manual. However, the thresholds cannot be auto-fixed as image variation. So, a new method is proposed to calculate the optimal high and low thresholds through iteration arithmetic. It is shown in formulae (6) and (14).

The vibration image and the images after processed by different methods of edge detection are shown in Figure 3.



Vibration image LOG operator

Canny operator Proposed operator

Figure 3.Comparison Among Different Methods of Edge Detectors for HUD Image

The new edge detection algorithm based on Canny operator comprises the following steps:

a) Use a two-dimension Gaussian filter to get convolution, so as to restrain noise.

$$I(x,y)=G(x,y)\otimes f(x,y) \tag{2}$$

$$\text{And } G(x,y)=\frac{1}{2\pi\sigma^2}\exp(-\frac{x^2+y^2}{2\sigma^2})$$

Where, $f(x,y)$ denotes the original image, $I(x,y)$ denotes the image after smoothing, G denotes Gaussian template. $\otimes$ denotes convolution computation.

b) Using a derivative operator to find out two orientation derivatives, and to calculate the gradient magnitude and determine size and orientation of grads.

$$M(i,j)=\sqrt{G_x(i,j)^2+G_y(i,j)^2} \tag{3}$$

$$\theta(i,j)=acr\tan\left(G_y(i,j)/G_x(i,j)\right) \tag{4}$$

Where, $G_x(i,j)$ denotes the orientation derivative in horizontal direction, $G_y(i,j)$ denotes the orientation derivative in vertical direction, $M(i,j)$ denotes the gradient magnitude, and $\theta(i,j)$ denotes the gradient direction.

As mentioned above, in order to extract the low intensity edge, the new formula is proposed to calculate the gradient magnitude.

$$M(i,j,\theta)=\max\left(\cos\theta G_x(i,j),\sin\theta G_y(i,j)\right) \tag{5}$$

c) Non-maximum suppression. To traversal a image, if the grey value of a pixel is not the highest among the former and the latter on the gradient direction, the grey value of the pixel is set to 0, that means it is not an edge point.

d) To compute two threshold values by accumulate histogram of an image, the pixel's gray value higher than the high threshold value is the edge, in contrast, the lower grey value of the pixel is not the edge pixel. For the ones with the grey values between the two threshold values, if the gray values of abutting pixels are higher than the high threshold, they are edges.

The iteration arithmetic is as follows:

(1) Get the initial threshold.

$$T_0=\frac{Z_{\max}+Z_{\min}}{2} \tag{6}$$

$$T_0=\{T_k,k=0\} \tag{7}$$

Where, T_0 denotes the initial threshold, $Z_{\max}$ denotes the max gray scale value of an original image, $Z_{\min}$ denotes the min gray scale value of the original image, and K denotes the number of iteration.

(2) Split an image into two parts by the threshold.

$$H_1=\left\{f(x,y)\,\middle|\,f(x,y)\geq T_k\right\} \tag{8}$$

$$H_2 = \left\{ f\left(x,y\right) \middle| f\left(x,y\right) < T_k \right\} \tag{9}$$

Where, $f\left(x,y\right)$ denotes the original image, $H_1\left(x,y\right)$ denotes one part of the image, $H_2\left(x,y\right)$ denotes the other part of the image.

(3) Calculate the average grey values of H_1 and H_2 respectively.

$$M_1 = \frac{\sum_{f(i,j)\geq T_k} f(i,j)}{\sum_{f(i,j)\geq T_k} N_H(i,j)} \tag{10}$$

$$M_2 = \frac{\sum_{f(i,j)< T_k} f(i,j)}{\sum_{f(i,j)< T_k} N_L(i,j)} \tag{11}$$

$$N_H(i,j) = \begin{cases} 1 & f(i,j) \geq T_k \\ 0 & otherwise \end{cases} \tag{12}$$

$$N_L(i,j) = \begin{cases} 1 & f(i,j) < T_k \\ 0 & otherwise \end{cases} \tag{13}$$

Where, M_1 denotes the average gray scale of H_1, M_2 denotes the average gray scale of H_2 .

(4) Get the new threshold T_{K+1}.

$$T_{k+1} = \frac{M_1 + M_2}{2} \tag{14}$$

(5) If $T_k = T_{K+1}$, go to step (6), otherwise, $k = k+1$, and go to step (2) .

(6) Finish the iteration, the high threshold is M_1, and the low threshold is M_2 .

The image after the edge extraction is shown in Figure 4.

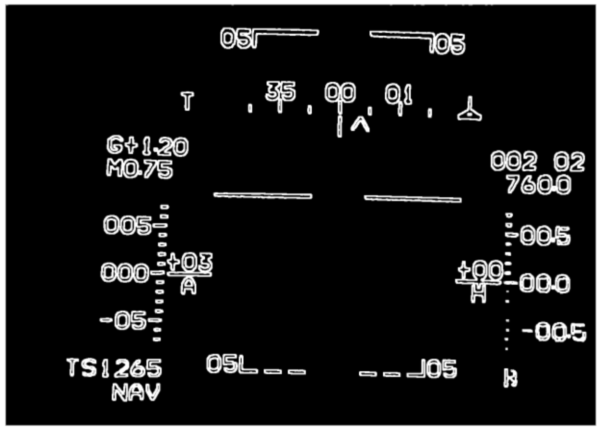


Figure 4. Result of Edge Detection

3.2. Improved Corner Extraction Method

According to the HUD image's characteristics, the image's grey level transformation is not obvious, moreover, the entire image only have two gray values (0 or 255) as a binary image. However, the feature point extraction based on edge detection selects the maximum grey scale region as the feature points. Apparently it is unreasonable to extract the feature points using the highest gray value. Therefore, it needs to improve the feature point extraction based on edge detection.

Because one might not select the highest gray value point as the feature point in the image, it means that one cannot extract the HUD feature point from gradation changes. Hence, one needs to select feature points based on the corner extraction.

The corner in an image generally refers to the point which has the high curvature on the image boundary. Harris operator [18-20] was proposed by Harris and Stephens in 1988 based on the signal strength spot feature extraction operator. This operator introduces the autocorrelation function of signal processing theory, and combines the image partial autocorrelation function with the corner detecting closely, then, determines whether the point is the corner point by analyzing the eigenvalue. This computation load of the operator is relatively small, because it only uses the first-order differential. Moreover, the Harris operator has the high stability. The Harris operator's formula is:

$$\hat{C} = G \otimes \begin{bmatrix} I_x^2 & I_x I_y \\ I_x I_y & I_y^2 \end{bmatrix} = \begin{bmatrix} \hat{I}_x^2 & \hat{I}_x \hat{I}_y \\ \hat{I}_x \hat{I}_y & \hat{I}_y^2 \end{bmatrix} \tag{15}$$

And $G = \frac{1}{2\pi\sigma^2} \exp(-\frac{x^2 + y^2}{2\sigma^2})$

Where, Ix denotes the horizontal differential, Iy denotes the vertical differential, G denotes the Gaussian template. $\otimes$ denotes convolution computation.

Considering the step of feature detection already including the Gaussian filter, the Gaussian filter in the corner extraction is unnecessary. The improved corner extraction is proposed. It does not only keep the good accuracy, but also increase the processing speed. Its formula is:

$$\hat{C} = \begin{bmatrix} I_x^2 & I_x I_y \\ I_x I_y & I_y^2 \end{bmatrix} \tag{16}$$

The improved corner extraction method comprises the following steps:

(1) Calculating the image gradients at the edge points, which are the differentials in the horizontal *Ix* and the vertical *Iy*;

(2) Constructing a matrix $\hat{C}$ which is related to a autocorrelation function. The eigenvalues of the matrix λ_1, λ_2 are the autocorrelation function's first-order curvatures, the formula is:

$$\hat{C} = \begin{bmatrix} I_x^2 & I_x I_y \\ I_x I_y & I_y^2 \end{bmatrix} \tag{17}$$

Where, *Ix* denotes the horizontal differential, and *Iy* denotes the vertical differential.

(3) Extracting feature points: if the eigenvalues λ_1, λ_2 are maximum, it means that the two first-order curvatures are high. So this point is considered to be the feature point. The formula is as follows:

$$H = \det(\hat{C}) - K \cdot trace^2(\hat{C}) \tag{18}$$

And $\det \hat{C} = \lambda_1 \cdot \lambda_2$, $trace\hat{C} = \lambda_1 + \lambda_2$

Where, λ_1, λ_2 are the eigenvalues of the autocorrelation matrix $\hat{C}$. det($\hat{C}$) denotes calculating the determinant. trace ($\hat{C}$) denotes calculating the matrix trace. K is a constant, and its experience range is from 0.04 to 0.06. $K \cdot trace^2(\hat{C})$ denotes the revision in order to suppress false detection of edge points.

After calculating, the image is divided into a number of r×s non-overlapping regions A_k, then, the maximum of *H* is selected as the feature points in A_k. The final result of extraction is shown in Figure 5.

Figure 5. Result of Improved Feature Points

3.3. Experiment Result

By comparing the results of different methods, it shows that the improved feature extraction on the HUD image vibration detection has robustness and stability. It can accurately detect the feature points in the situations of rotating image and noise jamming. It has a low false detection rate.

For the three above described methods of extracting feature points, this paper separately calculates the processing time of the three feature point extraction methods under VC6.0 platform. The results are shown in Table 2.

Table 2. Time of Three Feature Points Extraction Methods

Type of feature points extraction	computing time(single cores)
Improved Features Extracting Algorithm	1.263405s
Corner detection based on Harris	3.649676s
Interest Point detection based on Moravec	2.602403s

As it can be seen from Table II, the speed of the improved algorithm in the calculation is obviously faster. It verifies the superiority of the improved algorithm.

What's more, based on the feature point matching, one can get all the feature points' vibration offsets. The result of the feature point matching for two images is shown as in Figure 6.

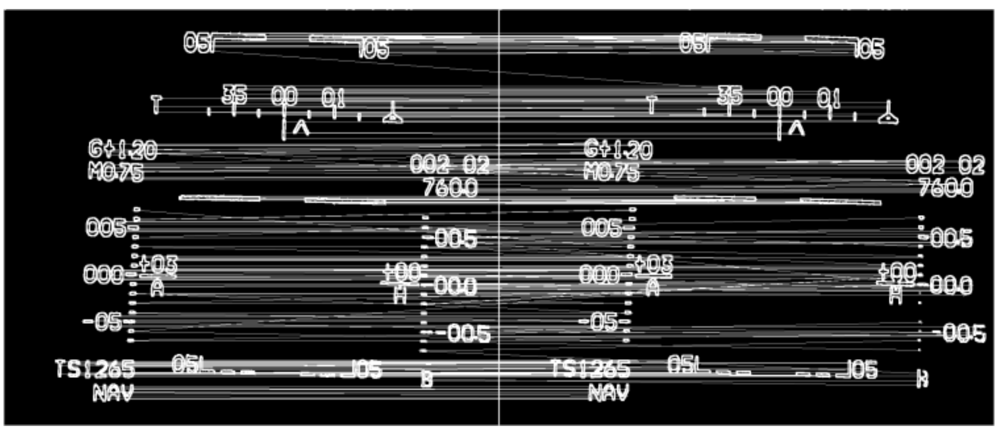


Figure 6. Result of Feature Point Matching for Two Images

According to the feature points' vibration offsets, the image's vibration offset can be calculated. The function is:

$$S = \frac{1}{N}\sum_{i=0}^{N-1} S_i \tag{19}$$

Where, S denotes the image's vibration offset, N denotes the numbers of feature points, S_i denotes the number i of feature point's vibration offset. The results are shown in Table 3.

Table 3. Every Feature Point's Offset and Image's Offset

The NO. of feature points	The offset of feature point
1	2.2
2	3.0
3	3.0
4	1.0
5	1.0
…	…
190	4.5
191	2.8
192	3.6
193	4.1
194	2.2
Offset of image	1.3

As shown in Table III, every feature point's vibration offset is not the same. The simulation of feature points' vibration offsets is shown in Figure 7.

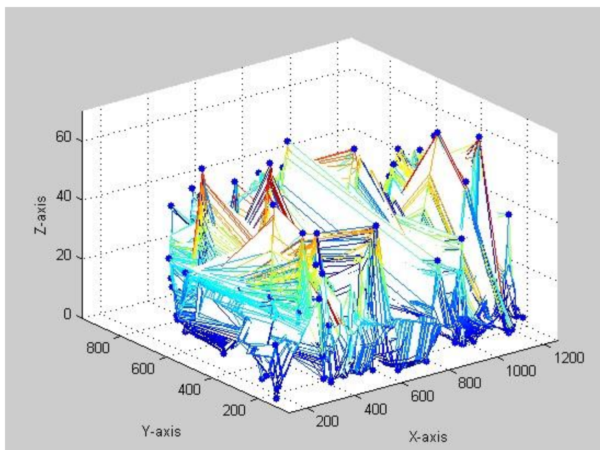


Figure 7. Simulation of the Whole Image's Vibration Offset

Where, the X-axis denotes the x-coordinate of feature point, the Y-axis denotes the y-coordinate of feature point, and the Z-axis denotes the image's vibration offset.

4. Conclusions

In this paper, a new HUD image vibration detection algorithm based on the improved feature extraction is introduced. If an original image is given, the feature-based methods are applied in the new algorithm, including the improved edge detection operator and the improved corner extraction algorithm. Comparing with the different algorithms, the testing results show that the proposed algorithm not only has higher precision, but also performs in real-time. What's more, the feature points' vibration offsets and the image's vibration can be calculated. A sequence of vibration images are used for testing the new algorithm. The result proves that it is satisfactory.

Acknowledgments

This research is financially supported by the National Natural Science Fund in China

(Grant No. 61170147), by Xi'an City Science & Technology Fund with no. CX1252(8) in China, and by Special Fund for Basic Scientific Research of Central Colleges (Innovation team), at Chang'an University of the Education ministry in China (Grant No. 2011xx26).

References

[1] S. Battiato and R. Lukac, "Video stabilization techniques", Encyclopedia of Multimedia. New York: Springer-Verlag, **(2008)**, pp. 941-945.
[2] R. Lukac, "Single-Sensor Imaging: Methods and Applications for Digital Cameras", Boca Raton, FL: CRC, **(2008)**.
[3] R. Szeliski, "Image alignment and stitching: A tutorial", Found. Trends Comput. Graph. Comput. Vis., vol. 2, no. 1, pp. 1-104, **(2006)**.
[4] M. Irani and P. Anandan, "About direct methods", Proc. Int. Workshop Vision Algorithms, Held During ICCV, Corfu, Greece, **(1999)**, pp. 267-277.
[5] P. H. S. Torr and A. Zisserman, "Feature based methods for structure and motion estimation", Proc. Int.Workshop Vision Algorithms, HeldDuring ICCV, Corfu, Greece, **(1999)**, pp. 278-294.
[6] K.-Y. Lee, Y.-Y. Chuang, B.-Y. Chen and M. Ouhyoung, "Video stabilization using robust feature trajectories", Proc. IEEE Int. Conf Computer Vision, **(2009)**, pp. 1397-1404.
[7] S. Battiato, Senior Member, IEEE, "Arcangelo Ranieri Bruna, and Giovanni Puglisi, A Robust Block-Based Image/Video Registration Approach for Mobile Imaging Devices", IEEE transtraction on multimedia, vol. 12, no. 7, **(2010)** November.
[8] Z. Hong-ying, J. Hong and X. Jing-wu, "Electronic Image Stabilizatiing Technology Overview", Optics and Precision Engineering, **(2001)** September, pp. 353-358.
[9] Y. Matsushita, E. Ofek, W. Ge, "Full-frame viedo stabilization with motion inpainting", IEEE Transactions on Pattern Analysis and Maching Intelligence, vol. 28, no. 7, **(2006)**, pp. 1150-1163.
[10] J. T. Qiu, Y. S. Li and C. L. Wan, "A dynamic detection bloks selection-based image stabilization algorithm", Acta Photonica Sinica, (in Chinese), vol. 39, no. s1, **(2010)**, pp. 202-205.
[11] H. R. Pourreza, M. Rahmatt and F. Behazin, "An electronic digital image stabilizer based on stationary wavelet transform(SWT)", ICIP, vol. 2, **(2003)**, pp. 383-386.
[12] S. Erturk and T. J. Dennis, "Image sequence stabilization based on DFT filtering", IEEE Proc.On Image Vision and Signal Processing, vol. 127, **(2000)**, pp. 95-102.
[13] A. Censi, A. Fusiello and V. Roberto, "Image stabilization by features tracking", 10th International Conf.on image analysis and processing, Venice Italy, **(1999)** September, pp. 27-29: 665-667.
[14] F. van der Heijden, "Edge and line feature extraction based on covariance models", IEEE transctions on pattern analylsis and machine intelligence, vol. 17, no. 1, **(1995)** January.
[15] H. Zhao, G. Qin and X. Wang, "Improvent of Canny Algrothim Based on Pavement Edge Detection", 2010 3rd International Congress on Image and Signal Processing(CISP 2010).
[16] J. Zhang, W. Chang and L. Wu, "Edge Detection Based on General Grey Correlation and LoG Operator", 2010 International Conference on Artificial Intelligence and Computational Intelligence.
[17] X. Wang, Member, IEEE, "Laplacian Operator-Based Edge Detectors", IEEE transctions on pattern analylsis and machine intelligence, vol. 29, no. 5, **(2007)** May.
[18] C. Harris, M. Stepheens, "A combined corner and edge detector", Alvy Vision Conference, **(1988)**, pp. 147-151.
[19] G. Qingji, X. Ping and W. Man, "Breakage Detection for Grid Images Based on Improved Harris Conern", 2011 International Conference on Transportation, Mechanical, and Electrical Engineering (TMEE), Changchun, China, December 16-18.
[20] J. Malik, R. Dahiya, G. Sainarayanan, "Harris Operator Corner Detection using Sliding Window Method", Internation Journal of Computer Applications(0975-8887), vol. 22, no. 1, **(2011)** May.

