

Vessel Tracking Vision System using a combination of Kalman Filter, Bayesian Classification, and Adaptive Tracking Algorithm

Yun Jip KIM*, Yun Koo Chung**, Byung Gil LEE**

*Department of Information Security Engineering, UST (University of Science and Technology), Korea

**ETRI (Electronics and Telecommunications Research Institute), Korea

dc3220@etri.re.kr, ykchung@etri.re.kr, bglee@etri.re.kr

Corresponding Author: bglee@etri.re.kr, Tel: +82 42 860 1689

Abstract— In these days, there are many vessel traffics to trade with foreign nations and travel abroad. Near coast or in harbor, the more traffics of transportation, the more possibility of accidents tends to occur. Thus, to reduce ships collision, vessel traffic services (VTS) centers have installed lots of equipment to keep a close eye on ships sailing in sea port, such as night observation device, telescope, and CCTV. To improve efficiently existing tracking system and overcome flaw of noises in the process of pursuit in maritime environment, considering bad weather and waves, this paper presents vessel tracking system using an image input device. The tracking system uses a fusion of Bayesian classifier to distinguish some images at initial stage, Kalman filter algorithm for keeping tracking the watercraft when it cannot be detected from the obtained image because some noises or inappropriate parameters used in the library functions may prevent detection from successive pictures, and the adaptive tracking algorithm for not only whether Kalman filtering is used as adaptive way to reduce a computational time but also disregarding the noise interference. The experimental results are included to prove the validity of the proposed method.

Keywords— Adaptive Tracking with Kalman Filter; Bayesian Classifier; Vessel Tracking; Object Detection

I. INTRODUCTION

The Hebei Spirit oil spill was a major oil spill in South Korea that began on the morning of 7 December 2007 local time, with ongoing environmental and economic effects. The incident occurred near the port of Daesan on the Yellow Sea Coast of Taean County. The collision punctured three of the five tanks aboard the Hebei Spirit and resulted in the leaking of some 10,800 tonnes of oil. On 9 December it was reported that the oil slick was already 33 km long and 10 m wide and 10 cm thick in some areas [10]. Consequently, to prevent that kind of terrible accident from happening again, vessel traffic controllers undertake a task of identifying and keeping watch on the watercrafts to avoid bumping into each other in a harbor. In spite of many existing programs to help the controllers, there are some technical problems which should be improved for getting constant tracking and detection despite requirements of real-time execution and noise robustness. Therefore, this paper presents object detection and

tracking algorithm effectively used in oceanic environment to find the movement of a ship.

First, to enhance those matters, we introduced a mixture of Bayesian classification ([2], [3]), Kalman filter [1], and Adaptive Tracking Algorithm.

Second, Bayesian classification was used to analyse which parameter value was suitable for threshold on input video stream. After classifying images, our program chose a proper threshold value related to the result of the classification. The value appropriate for threshold of each image had already been stored in system memory.

Third, it was Kalman filter which kept tracking a ship in case the characteristic spots of it were not found in image datum.

Lastly, this research presents Adaptive Tracking algorithm that reduced calculating time and was effective to keep off noises as the algorithm only implemented kalman's method if a function could not get the special mark on input frame from observation equipment.

The flow chart of our system is shown schematically in figure1.

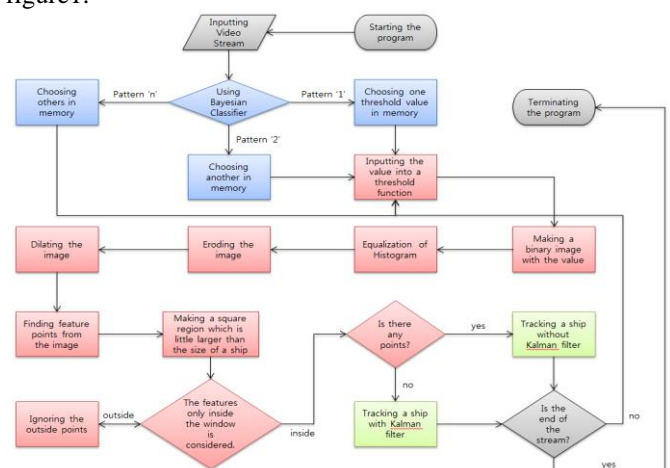


Figure 1. A flow chart of the vessel tracking vision system. The colour of the rectangular means Bayesian classification in blue, adaptive tracking algorithm in read, and Kalman filter in green.

This program was simulated under conditions that the system hardware configuration was Intel Core i5@2.5Ghz, 8Ghz DDR3 SDRAM, and Intel(R) HD Graphics 4000 and the software packages used were Microsoft Windows 7 Enterprise K, Microsoft Visual Studio 2010 (64bit), OpenCV 2.4.5, and armadillo-3.920.1 version.

II. APPLYING BAYESIAN CLASSIFICATION

We put the Bayesian's method to distinguish between a colour image from a normal CCTV during the daytime and a gray-scale image from an infrared camera at night.

The algorithm of Bayesian is a simple classification model, in case data belongs to a class follows a Gaussian or normal distribution. Given that some data has been offered to us, we can calculate a probability of the data attained from a class by using Bayesian's classification. It is defined by the following expression ([2], [3]).

$$P(\theta_i|F_j) = \frac{P(F_j|\theta_i)P(\theta_i)}{P(F_j)} \quad (1)$$

F_j means the extracted feature vector by a raw image from the video stream that is consist of all of red, green, blue (RGB) channels at an early stage. θ_i means the class we have to predict. It is said that $P(\theta_i|F_j)$ is likelihood, and $P(\theta_i)$ is prior probability.

The classification of Bayesian is simple to build a code and intuitive to understand its structure, so it is widely used in many programs. However, if data distribution is far from Gaussian density function with its complexity, it is difficult to expect a good result. To avoid this problem, we used the classification using Gaussian mixture model.

A. The procedure in Bayesian's way for categorization used in our system.

- 1) Prior to the beginning of this algorithm, determine not only the number of classes and sample datum but also an input image extracted from video stream.
- 2) The program calculates randomized normal deviation from an input image. Some images from various things or circumstances might not follow the model of Gaussian distribution. If so, the capture frame is required to follow this process. After it came done, we checked the values whether a distribution of the values resembles Gaussian Curve. We took the average and standard deviation (STD) of all values of the image. The following expression of STD is shown below [5].

$$\sigma = \sqrt{\frac{\sum_{k=1}^n (x_k - m)^2}{n}} = \sqrt{\frac{\sum_{k=1}^n x_k^2}{n} - m^2} \quad (2)$$

$\{x_1, x_2, \dots, x_n\}$ are observed values of the sample items, the denominator 'n' stands for the size of the sample, and 'm' means average of the values. After obtaining a histogram, Gauss's formula below results in drawing a graph.

$$f(x) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-\mu)^2}{2\sigma^2}} \quad (3)$$

The above expression [5], in other words, is called Normal Distribution. ' μ ' is a mean, ' σ ' stands for STD from the expression (2), and 'x' means a variable.

- 3) Now for getting Bayesian probability, the classifier categorized the randomized values within the Normal Distribution (3) into each group, by the formula expressed above (1).

B. The results of the experiment are shown the following the graphs and snapshots.



Figure 2. Two initial input images from a CCTV, the left is colour by a CCTV and the right is grayscale by an infrared camera.

Colour Image			Gray-scale Image		
Class	Frequency	Gaussian Dist	Class	Frequency	Gaussian Dist
0	0	0.000271055	-300	0	1.95827E-06
30	0	0.000599968	-260	0	6.18272E-06
60	2	0.001167723	-220	0	1.77084E-05
90	30	0.001998455	-180	1	4.60121E-05
120	48	0.003007399	-140	1	0.000108457
150	20	0.003979517	-100	2	0.000231917
180	15	0.004630331	-60	7	0.000498884
210	65	0.004737356	-20	8	0.000791701
240	19	0.004261892	20	16	0.001263902
270	6	0.003371408	60	12	0.001830449
300	44	0.002345106	100	19	0.002404884
330	45	0.001434353	140	24	0.002866306
360	6	0.000771422	180	43	0.003099152
390	0	0.000364813	220	40	0.003039872
etc	0		260	36	0.002704954
			300	34	0.002183516
Mean	200.33		340	21	0.001598987
STD	83.65121749		380	18	0.001062247
			420	10	0.000640174
			460	5	0.000349996
			500	1	0.000173588
			540	1	7.8103E-05
			580	0	3.18793E-05
			620	1	1.18043E-05
			660	0	3.96521E-06
			700	0	1.20832E-06
			etc	0	

Figure 3. Table of the values derived from the equations (2), (3) with Figure 2.

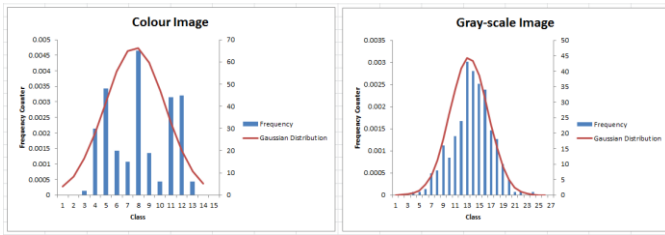


Figure 4. Normal distribution and histogram resulted from the formula (3) with Figure 3.

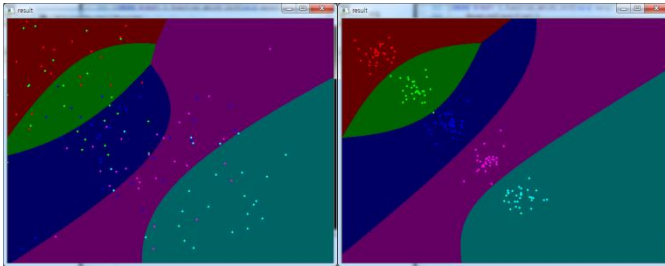


Figure 5. Different patterns after the prediction of Bayesian Classifier (1), colour is on the left and gray scale is on the right.

III. ADAPTIVE TRACKING ALGORITHM

This method allows Kalman filter to be applied selectively so as to make less dependent on system performance than that of an ordinary way that the filtering is fully invoked. There are several phases to develop the adaptive tracking algorithm.

A. A preparatory stage for detecting features on the captured frames.

- 1) All input visual datum from the video stream are comprised of red-green-blue (RGB) channels, which are not only colour by general CCTV but also black and white image by a specific camera including any kind of thermal, infrared, and night vision device. Nevertheless, most of the several preprocessings to eliminate or decrease noises demand to be grayscale before entering a data into the library function.



Figure 6. Initial image converted to monochrome information from Figure 2, having same depth but different channel from 3 to 1.

- 2) It is time to utilize the patterns obtained from the Bayesian Classification stage and select an optimal value for threshold from the memory. Before our program chooses a parameter corresponded to those

patterns, the optimal values for threshold should be in system memory or database. We could assort input images (Figure 2) to assign a proper value to the program totally influenced by threshold value. The expression of threshold process is as follows [4].

$$s = T(r) = \begin{cases} \max_value & \text{if } r > \text{threshold} \\ 0 & 0 \times w \end{cases} \quad (4)$$

If the value (r) of brightness of an input image is greater than that of a given threshold, it will take \max_value as the luminance. Otherwise, zero will be chosen. An optimal threshold value relies on features of image and greatly affects the next operations. Therefore, it is important that the value be adequate for each image. Our tracking method produced good results with acceptably detected features on the left video stream from Figure 2, which is the threshold value of 23.0 and it showed us a fine performance of finding some characteristic points on the other stream about that of 203.0.

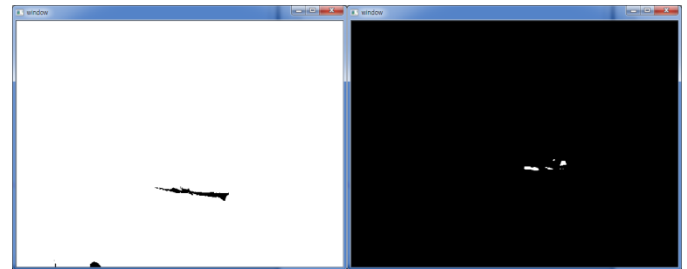


Figure 7. Result with threshold from each image at the same position of Figure 6, applied with 'max_value = 255'

- 3) For the sake of minimizing noises and filling some sparse holes on Figure 7, we needed to do morphological operation of which our program is an erosion and dilation technique with 3×3 window, solving the problem by convolution theorem.

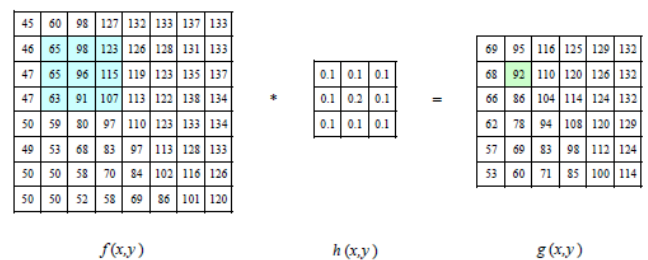


Figure 8. Neighbourhood filtering (convolution): The image on the left is convolved with the filter in the middle to yield the image on the right. The light blue pixels indicate the source neighbourhood for the light green destination pixel [6].

The theorem of convolution is shown as follows [7].

$$(f * g)(t) = \int_{-\infty}^{\infty} f(t-x)g(x)dx \quad (5)$$

Let $f(t)$ and $g(t)$ be two functions of t . The convolution of $f(t)$ and $g(t)$ is also a function of t , denoted by $(f * g)(t)$ and is defined by the relation. However if f and g are both causal functions then $f(t)$, $g(t)$ are written $f(t)u(t)$ and $g(t)u(t)$ respectively, so that

$$(f * g)(t) = \int_{-\infty}^{\infty} f(t-x)u(t-x)g(x)u(x)dx = \int_0^t f(t-x)g(x)dx \quad (6)$$

Because of the properties of the step functions ($u(t-x) = 0$ if $x > t$ and $u(x) = 0$ if $x < 0$). In summary above the formular (5), we get [7]

$$c = f * g \quad (7)$$

f substitutes for a binary image and then, g will be 3×3 window.

The binary morphology standard operations used in the system include two things below [6].

$$\begin{aligned} \text{dilate}(f, g) &= \theta(c, 1) \\ \text{erode}(f, g) &= \theta(c, S) \end{aligned} \quad \theta(f, g) = \begin{cases} 1 & \text{if } f \geq g \\ 0 & \end{cases} \quad (8)$$

S means the size of window (number of pixels).

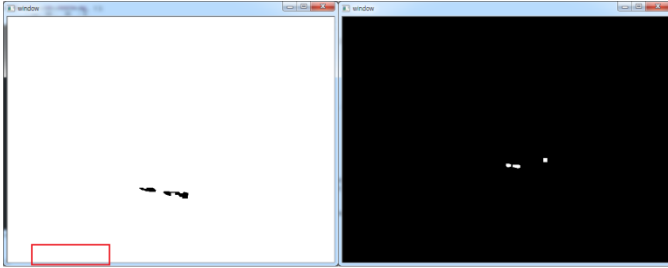


Figure 9. Noise was removed after running the erosion and dilation two times each. (See the rectangular in read, comparing with Figure 7.)

B. Detecting features based on characteristics of image.

The program extracts keypoints that are local scale-space extrema. The scale space is constructed by computing approximate values of laplacians with different sigma's values at each pixel. Instead of using pyramids, a popular approach to save computing time, all of the laplacians are computed at each pixel of the original high-resolution image. But each approximate laplacian value is computed in $O(1)$ time regardless of the sigma, thanks to the use of integral images. The algorithm is based on research paper that is Agrawal's CenSurE (Center Surround Extremas for Realtime Feature Detection and Matching) [9], but although the paper presents two level filter such as circle, square, and octagon, the method of our

system uses an 8-end star shape, consisting of overlapping upright and tilted squares [4].

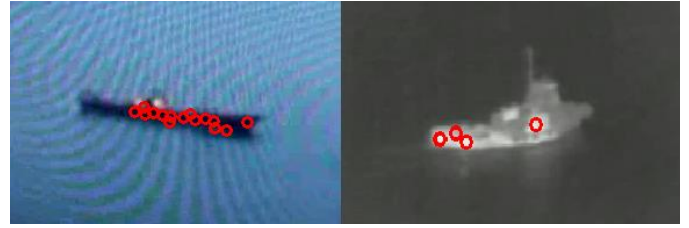


Figure 10. Detection of some features on the images from Figure 9

The pseud code of this step is shown below.

- 1) Set up the interval value.
- 2) Enter loop.
- 3) Find all features of a vessel on an input image.
- 4) Store coordinates of them into an array.
- 5) Sort it by bubble sorting algorithm.
- 6) Pick out both the highest and lowest value from the array.
- 7) Subtract each other to get width or height for dimension of which square makes an invisible boundary enclosed around an observed ship.
- 8) If either width or height of each current and previous square is larger than the interval value of step 1, go step 3 after using Kalman filter. Otherwise, move to the next step.
- 9) Calculate a centre point of each current and previous square.
- 10) If an interval of the two points is larger than that of step 1, go step 3 after using Kalman filter. Otherwise, move to the next step.
- 11) Update the previous height, width, and centre point to the current things.
- 12) If the video stream is not yet finished, go step 3. Otherwise, exit this loop.

IV. ADOPTING KALMAN FILTER

With sailing ships detected and tracked in marine environment, they cannot be caught on observation owing to movement of vessel moving away or rotation of the device obstructing its field of vision. In this situation, it is possible that adaptation of Kalman filter taking motion characteristics of an object into consideration makes the reliability and safety of its tracking improved efficiently. This filtering model was applied to our system and as follows ([1], [9]).

We defined the state vector $X_k = [x, y, v_x, v_y]$, measurement, where x and y were the centre coordinate of a detected watercraft; v_x and v_y are its velocity on each direction of x and y axis.

$$\begin{aligned} \text{Motion equation:} \quad X_k &= F \cdot X_{k-1} + W_k \\ \text{Observation equation:} \quad Z_k &= H \cdot X_k + V_k \end{aligned} \quad (9)$$

W_k and V_k are movement and measurement noise vectors which obey Gaussian distribution, F is state transition matrix, and H is measurement matrix.

Prediction equation and the update equation are as follows [9].

$$\begin{aligned}
 \text{Prediction equation 1:} \quad & X'_k = F \cdot X_{k-1} \\
 \text{Prediction equation 2:} \quad & P'_k = F \cdot P_{k-1} \cdot F^T + Q \\
 \text{Kalman-gain} \\
 \text{equation:} \quad & K_k = P'_k \cdot H^T \cdot (H \cdot P'_k \cdot H^T + R)^{-1} \\
 \text{Update equation 1:} \quad & X_k = X'_k + K_k \cdot (Z_k - H \cdot X'_k) \\
 \text{Update equation 2:} \quad & P_k = P'_k - K_k \cdot H \cdot P'_k \quad (10)
 \end{aligned}$$

The values of state transition matrix F , measurement matrix H , process noise covariance matrix Q , and measurement noise covariance matrix R list as follows.

$$\begin{aligned}
 F = \begin{pmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}, \quad Q = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}, \\
 H = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix}, \quad R = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \quad (11)
 \end{aligned}$$



Figure 11. Tracking the ship without Kalman filter from Figure 10



Figure 12. Tracking the ship with Kalman filter from Figure 10

Figure 11 and Figure 12 have shown us an adaptive way of using Kalman filter. The red circle means detecting kept its

operation well but the blue could not find any special features on the ship, depended on the filtering method. Through this way, our program continually was able to perform tracking the vessel in the screen.

V. CONCLUSIONS

In this paper, we conducted several steps to track a sailing ship with interferences of other vessels as well as detected it with noises.

First, the Bayesian Classification was used to classify some images of which video stream came from a common CCTV or thermal observation device. After classifying the images, we could get a proper threshold value related to a result of Bayesian's.

Second, the adaptive tracking algorithm made not only an image free from noises and unwanted object detection but also the system less dependent on its performance because it run Kalman filter when detection method did not work properly.

Lastly, the Kalman filter enabled our system to keep tracking a moving vessel continuously, even though it was not detected frequently.

Thanks to a combination of these steps and methods mentioned above, we could make the ship tracking program robust and specialized in oceanic environment.

ACKNOWLEDGMENT

This work was supported by ETRI through Maritime Safety & Maritime Traffic Management R&D Program of the MOF/KIMST (2009403, Development of next generation VTS for maritime safety) and the project was supported by vessel traffic system (VTS) with research funds from the ministry of Maritime Affairs and Fisheries in Korea.

REFERENCES

- [1] Jung Hong PARK, Jin Hwan KIM, Nam Sun SON, "A study on vision-based object detection and tracking in a marine environment," in *KAOSTS*, 2012, p. 510-513
- [2] Yuto Maruyama, Gamhewage C. deSilva, Toshihiko Yamasaki, Kiyoharu Aizawa, "Personalization of Food Image Analysis," *IEEE*, 2010, p. 77
- [3] Open Robotics. Available: <http://blog.daum.net/pg365/208/>.
- [4] Dong Geun KIM, *OpenCV programming*, revised edition, kame publishing company, Seoul, Korea, 2011
- [5] Naver encyclopedia of knowledge. [Online]. Available: <http://terms.naver.com/>.
- [6] Richard Szeliski, *Computer Vision: Algorithms and Applications*. Springer, 2010, p. 112, 128
- [7] The Convolution Theorem. [Online]. Available: http://www3.ul.ie/~mlc/support/Loughborough%20website/chap20/20_6.pdf.
- [8] Mosalam Ebrahimi, Walterio W. Mayol-Cuevas, "SUSurE: Speeded Up Surround Extrema Feature Detector and Descriptor for Realtime Applications," *IEEE*, 2009, p. 9-14
- [9] Shengluan Huang, Jingxin Hong, "Moving Object Tracking System Based On Camshift And Kalman Filter," *IEEE*, 2011, p.1424
- [10] Wikipedia, [Online]. Available: http://en.wikipedia.org/wiki/2007_South_Korea_oil_spill/.



The first author was born in Boryeong city, Korea in 1984. He graduated from Inha University where he got a bachelor's degree in computer information engineering in 2009. At present, the writer is also getting a master's degree in information security engineering at University of Science and Technology. His main interesting field includes real-time embedded system, computer vision, and field-programmable gate array.



Byung-gil Lee received a Ph.D degree in electrical engineering from the Kyungpook National University in 2003. In 2001, he joined the research member of ETRI in Korea and is currently a project leader of maritime-IT convergence and cooperative driving security research project. His current research interests include cooperative ITS and maritime IT convergence for e-navigation.



Dr. Yun Koo Chung was born in Seoul, Korea and received the BE degree in Electronics Engineering from Korea University, the MS degree in Computer Science from Cleveland State University and the PhD degree in Computer Science from Wayne State University in 1991. He is a professor in the University of Science and Technology since 2005. Also he has been working in computer vision research area in ETRI. He has interests in the field of user identification, 3D reconstruction, motion analysis in computer vision.