**Histogram of Oriented Gradients Controls Proportional Navigation and Proportional-Integral-Derivative (PID) on Quadcopter Platforms**

Alfarih Faza and Surya Darma\*

*Department of Physics, FMIPA Universitas Indonesia, Kampus UI Depok,   
Depok 16424, Indonesia*

a)Corresponding author: [suryadarma@sci.ui.ac.id](mailto:suryadarma@sci.ui.ac.id)

b)[alfarihfz@gmail.com](mailto:alfarihfz@gmail.com)

**Abstract.** Indonesia is located in Mediterranean Basin, Pacific Basin and flanked by 3 large plates i.e. Indo-Australian, Eurasian, and Pacific Plate. Therefore, Indonesia has many volcanic mountains and prone of natural disaster such as earthquake, tsunami, and volcanic earthquake. Usually after disaster happened, SAR (Search and Rescue) team would be deployed to search victims, and mapping disaster’s area quickly, however the assignment, especially after the disaster happened is very dangerous and high risk. Hence, SAR needs a high tech device to help that situation and minimize the risk. Here, we would like to propose Quadcopter as victim’s finder and area mapper as the solution to minimize the risk. A High speed and accurate response quadcopter is needed to conduct those things. Object detector mechanism must be installed to the Quadcopter in order to give signal for quadcopter to be able moving closer to the victims and to prevent from coliding objects during area mapping. The object detection is using image processing method, Histogram of Oriented Gradient (HoG). Proportional Navigation (PN) and Proportional-Integral-Derivative (PID) control system are used as quadcopter’s control system and both working simultaneously to control the movement of the quadcopter. Proportional Navigation (PN) will be used as default control system when the victim’s position is at long-distance. PN is also used to make quadcopter to move more aggressive with maximum acceleration. As the distance closer enough to the victims, the control system will be automatically switch to Proportional-Integral-Derivative (PID) Control System. PID is used at close distance because the flexibility and consistency of response in dynamic movement.

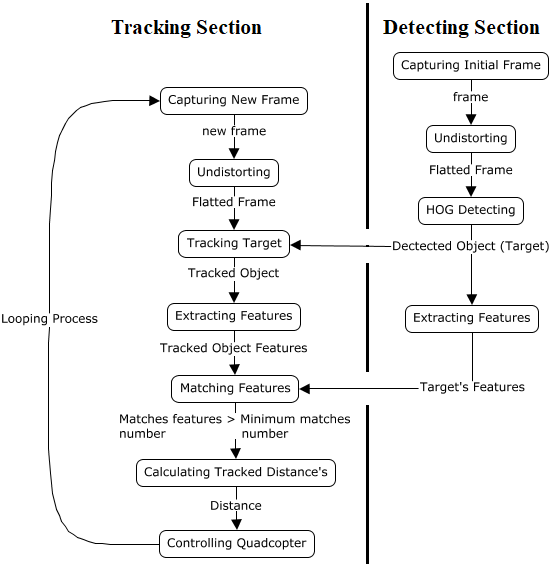
# INTRODUCTIONS

Human life is the most important thing that should be at priority when a disaster happened such as an earthquake, tsunami and others. Commonly, after an earthquake or tsunami happened, it shall jeopardize human being and their living place, like houses, buildings and other environment which lead to human life at risk. Then, SAR would mainly focus to save human life. Nevertheless, deploying human after or during disaster for area mapping and finding the victims is also very dangerous.

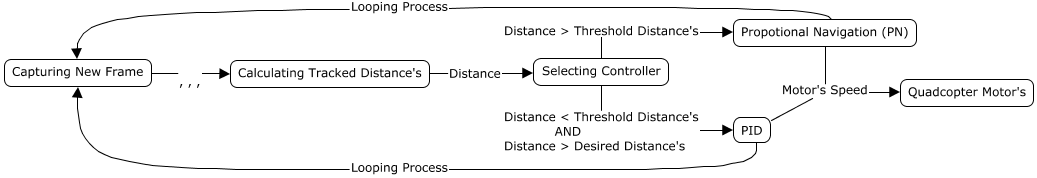
The first thing they need is a human detector tool that allows SAR to find and to recognize human victims. The tool which installed on the quadcopter is used as its controller parameter. The tools must have high response, and high accurateness to detect human. The tool might use a camera as the main sensor and use image processing to recognize human object. Hence, we propose in this paper is image processing method on the quadcopter. We use one of image features as the core of image processing which will be processed and will be extracted its information and for controlling the quadcopters. There are some work from our formers, they used image features as the main concept, such as develop existing image features [16], trying to find another kind of image features [12, 14, 15], or using image features for another purpose [13, 17]. The image features which we used is Histogram of Oriented Gradients (HoG) which is previously worked by Dalal and Triggs on his paper on CVPR-05[5]. Dalal and B. Triggs showed on their paper that their proposed HoG features combined with Support Vector Machine (SVM) had successfully detected pedestrian on occlusions. We used that previous work as basic human detection tools work. Since the object had been detected, we need to track the detected object and the tools must have an ability to track the same object continuously. For convenience, we used OpenCV’s library as base of our image processing. OpenCV’s library is a library for image processing purpose that provide an access to many kinds of image processing methods of previous works.

The tracked object features such as width of object on image form is in pixel representation is processed and calculated. The quadcopter’s controller will be using the processed features as PID and PN Controller parameters. The quadcopter’s controller is programmed to use PID and PN Control System as the main control system for the quadcopter to approach the detected object. Both will work alternately at certain programmed condition’s. We know PID is kind of control system which use its control error as control feedback to determine next control actions until it reaches and keep the aimed controls value which in other word the control error will be zero then. Generally, PID is applied in quadcopter [2] for controls all of 4 motors to reach the stability of quadcopter. PN is missile control system to catch missile targets with minimum spent time. We proposed to use PN Control Systems when the target at long range, until the distance is at certain condition then we switch to PID Control Systems to approach. We could give another action to the quadcopter for the victim after they are found and approached.

# DETAILED SYSTEM EXPLAINED



**Controlling Section**



**FIGURE 2.** Flow chart of method of this paper

Figures 2. shows the methodology’s flow charts of this paper. We take a part of whole system in three basic section. The block at Detecting Section must be done first before other section can be processed. The initial thing is capturing frame, then undistorts it to produce pure frame. Undistorting a frame need camera calibration to be done first. HoG Detecting works to detects human object on given frame. Extracting features: the image features of the object is one of the key for tracking target later, because its features must be registered as matching parameters in matching features mechanism then. Matching features: this step counts number of matches features of target’s features with tracked object’s features. If the number of matches features is greater than configured minimum matches number, then the distance of tracked object is measured. Measured distance is used as parameters to controlling quadcopter at Controlling Section. It will determine the quadcopter’s controller for approaches the target. Afterwards, to controlling quadcopter need an updated target position relative with quadcopter’s view after a command sent to quadcopter’s motor. So the Tracking Section must loop until mission is accomplished.

## 2.1. Camera Calibration

The quadcopter’s camera was programmed to take frame continuously, however we shouldn’t process the taken frame immediately to the next step because every camera has own characteristics which are extrinsic and intrinsic parameters. The extrinsic parameters represent the location of the camera in the 3-D scene, and the intrinsic parameters represent the optical center and focal length of the camera. They’re grouped in form of matrix called camera matrix which is always applied (multiplied/divided) to every captured frame so the frame will always had got distorted then. We need to undistorts it to pure condition by applying the same factor (camera matrix) respectively. Before all of those factor is re-applied to the frame, we must calculate the camera matrix by doing a calibration to the camera. The calibration algorithm calculating camera matrix using extrinsic and intrinsic parameters. OpenCV has a library to determine all of those things called CameraCalibration by using a chess board image then it is captured in various view by variating distance to the camera, rotating, and give a tilt angle. The CameraCalibration libraries will produce characteristics of camera such extrinsic parameters, camera distortion coefficients, and camera coefficients in form of matrix. All of those parameters doesn’t depend on the scene view so it can be used later as long as the focal length is fixed (lens zoom, and lens angle). We did CameraCalibration to the quadcopter’s camera to obtain those coefficients.

Image (a) is an original frame which were captured by the quadcopter’s camera. As shown above, the image got distorted by the camera until we haven’t applied distortion coefficient, and camera coefficient to undistorted it using ‘undistort’ function from OpenCV. We shows the result of undistorted image on image’s (b). The ‘Undistort’ function was made a flatted image (b) of the original image (a) which means those coefficients worked successfully to undistorting a captured frame of the quadcopter’s camera.

## 2.2. Blur Level Calculation

Firstly, before the frame is processed for human detecting by HoG, blur level of the frame is measured early. Petruz *et al* [7] reviewed nearly 36 different methods to estimate the focus measure on image. One of those method is by applying Laplacian’s matrix to frame’s matrix then it will produce second derivative of the frame. The second derivative of an image representing the intensity change. The assumption is if an image contains high variance then it must have a widespread of response. We have measured blur level by calculate the variance of the frame after had been convolved by Laplacian’s matrix.

Blur level of the captured frame is used as parameter to decide whether the frame will be processed or skipped by compares it with minimum value of accepted blur level. Below are examples of measured blur level of image iaken at same brightness condition.

|  |  |  |
| --- | --- | --- |
|  |  |  |
|  |  |  |

**FIGURE 3.** Example measured blur level at same brightness condition of frame taken.

## 2.3. Human HoG Detection

We use OpenCV’s function to detect and compute HoG Features. We used ‘HOGDescriptor’ from OpenCV and combined with SVM. The ‘GetDefaultPeopleDetector’ is a function of OpenCV’s library which is a trained SVM from Dalal and B. Triggs Paper’s [5] to detecting human object on frame. The function will return the human object position in shape of rectangle (rectangle of interest) and usually has an overlapping bounding box, so we applied Non-Maxima-Suppression (NMS) [8] to minimalize the bounding box. Sometimes the detector detects wrong object Using all of these function, we can detect human object on captured frame clearly as shown on images below.

|  |  |
| --- | --- |
| **Without NMS** | **With NMS** |
|  |  |
|  |  |
|  |  |

**FIGURE 4.** Compared object detection with NMS versus without NMS.

We also compared the effect of human size variated by distance with success rate of human detection by calculating ratio of detected human object with the total of human object on the given image as shown on figures below.

|  |  |  |
| --- | --- | --- |
| **Distance: 25 cm, Success Rate: 3/5** | **Distance: 28 cm, Success Rate: 2/5** | **Distance: 30 cm, Success Rate: 2/5** |
| **Distance: 32 cm, Success Rate: 2/5** | **Distance: 34 cm, Success Rate: 3/5** | **Distance: 36 cm, Success Rate: 3/5** |
| **Distance: 38 cm, Success Rate: 3/5** | **Distance: 40 cm, Success Rate: 3/5** | **Distance: 42 cm, Success Rate: 3/5** |
| **Distance: 44 cm, Success Rate: 3/5** | **Distance: 46 cm, Success Rate: 3/5** | **Distance: 48 cm, Success Rate: 2/5** |

**FIGURE 5.** Effect of human size variated by distance with success rate of human detecting

## 2.4. Object Tracking Method

The HoG Features Descriptor detects every human object at given image, so, there’s multiple rectangle of interest (ROI). We filtered the ROI based on only at certain criteria of human behaviour such as average ratio of human width [9] towards to human height in shape of pixel and for simplicity to approach, we choose a ROI which is located at nearest position to center of current view. Then, the filtered ROI is used to crop the frame to determine what target the quadcopter must approach and registered it as target image.

While the quadcopter in action to approach to the target, the target view would also move respectively towards the quadcopter current view, hence, longer, quadcopter would get lost line of sight (LOS). We used a tracker tools to keep tracking on the same object as guidance parameter and also to keep LOS of quadcopter because of its movement. OpenCV’s Library has many kind of method in the form of function for tracking object such as BOOSTING, MIL, KCF, TLD, MedianFlow, GOTURN, MOSSE, etc. We have had a survey to consider what kind of tracker that we will use by reading previous work [10] and did a test of several tracking methods on OpenCV’s Library. We considered to use MedianFlow tracker due it has a good fidelity compared with another tracking method on object tracking at our certain condition.

We also use a features extractor to improve accuracy and minimize failure of tracker on tracking target. OpenCV’s library gives several choice of features extractor such as SURF, SIFT, ORB, BRISK, AKAZE, etc. Earlier, we have tried using AKAZE, SURF, SIFT, and ORB extractor for extracts image features then conceded to use AKAZE features extractor. We also used ‘k-nearest-neighbor’ (knn) methods [11] for matching image features.

The MedianFlow tracker function’s return an update of ROI position of the target on every given frame. We check the coherency of image at updated ROI with the target image by extract image features of both image then the knn’s matcher matches all of extracted features. We called this process sequence as image matching phase. When the number of matches is less than the configured minimum matches number’s then it is count as false detect. The false detect won’t be processed and at certain number will trigger the tracker to reset to last known true image.

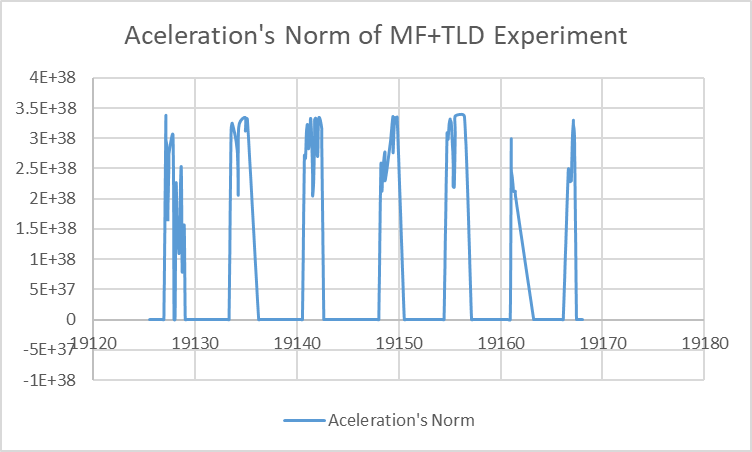
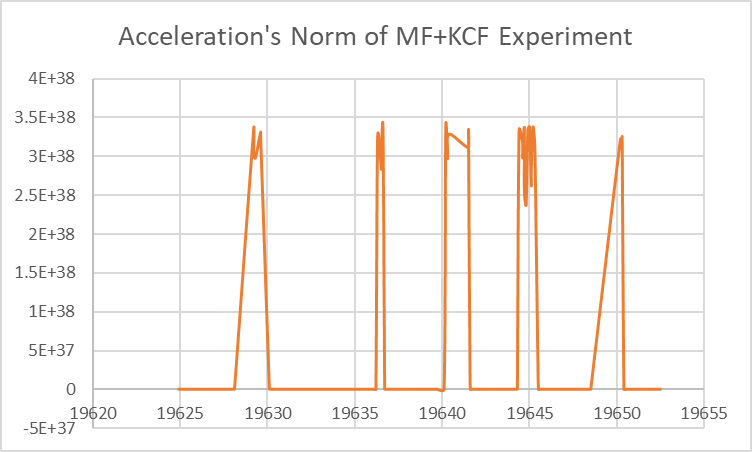
At high occlusion, the MedianFlow tracker tracks false object frequently, and has a consequence that tracker’s tracked target won’t be passed at image matching phase because it must do not has required matching features and also no distance for controller params to be measured. This condition, make controller such PID controller error’s grow up rapidly or ungovernable due to recent control error’s (feedback) isn’t get to PID controller right after PID controller has gave a control action at previous feedback. To accommodate this condition, we used a secondary tracker as backup tracker’s. We used TLD tracker to seek for when MedianFlow’s tracker fails on tracking target. But there is a cost in tracking time because of more process need to be done when there’s false track. We also compared the effect of used multi-tracker rather than only single-tracker as shown on table below.



**TABLE 1.** Comparison of usage multi-tracker (MF+TLD) than single-tracker (MF).

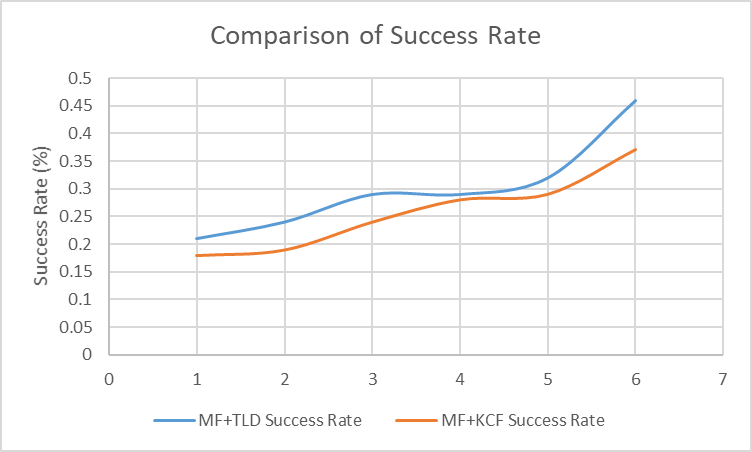
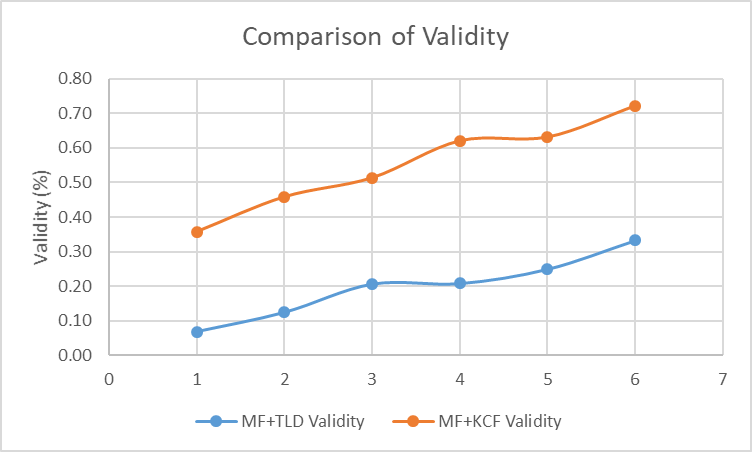


**TABLE 2.** Comparison of usage multi-tracker (MF+KCF) than single-tracker (MF).



(a) (b)

**FIGURE 6.** quadcopter acceleration on x’s and y’s vertices taken by Inertial Measuring Unit (IMU) on tracker test.

(a) (b)

**FIGURE 7.**  graph of success rate, and validity of multi-tracker test.

The figures 6 shows, the installed Inertial Measuring Unit (IMU) on quadcopter measuring quadcopter’s acceleration norm on randomized move. Thus, it will make the captured frame in high occlusion. Average tracking time is average of 100’s tracking time taken. Success rate is calculated by calculating the ratio of fails count towards total number of tracked target’s. At single tracker experiment, the validation is 1 because all of tracker process used only one tracker, so, no validation for other tracker. At multi-tracker experiment, validation number is ratio number of the secondary tracker was used toward total number of success of tracking process per 100 frame. As we can see at the table 1 and table 2, the usage of multi-tracker (2) increase average tracking time which means the tracking process has been taking longer time than the other ones as have mentioned above. However, the success rate also increase which means the controller’s limitation has been minimalized. Figures 7 is the graphs showing comparison of success rate and validity of tested multi-tracker method’s. From graphs shown on figures 7 (a), the MF+KCF has smaller success rate than MF+TLD but it has higher validity which means the usage’s number of KCF trackers is greater than the usage’s number of MF tracker. Even though the MF+KCF tracker has smaller success rate than MF+TLD or vice versa, the usage of multi-tracker has increased the success rate of tracking process. From those experiment, we conceded to use MF+KCF tracker due to lower of time elapsed than the others and the success rate is just slightly lower than MF+TLD trackers but also has increase the success rate than single-tracker usage.

## 2.5. Approching to Object

For approach to the object, the updated ROI from object tracking is calculated to determine control parameters. Quadcopter sight must be in line (centered) with target before it shall to approaches to target. Hence, we maintain its position by maintain its yaw and z position so the target in represent of ROI has positioned in the center of quadcopter sight. We used PID Controllers to adjusting the yaw speed and z speed of quadcopter by assign the center of image as PID controller’s set point parameter. Afterwards, we measured distance of the quadcopter towards the target to use as control’s parameter for quadcopter approach the target. Basically, we use 2 kinds of method to calculate distance from an object towards quadcopter’s camera. Those are by determine relation of distance versus target width in form of pixel and camera view range, by plotting those things into a graph then calculate linearity (regression analysis) or using equation below.

(1)

F is a constant called perceived focal length, D is known distance of object to camera, W is actual width of the object, and WP is perceived width of object in pixel forms.

(2)

We substitute F from first equation to above equation, then we obtain:

(3)

(4)

D prime is the measured distance which depends on constant WP multiplied by D. We named the distance which measured by those equation as DPD. By taking advantage of linear equations (linearity) which was obtained from regression analysis of distance versus target width in pixel, we also measured the distance using that linearity named as DDL. DPD and DDL have a limit in measure distance. Both are limited in only can measure distance at known and constant object’s width. To clarify this, we know if the object is different in such size it also must has different WP’s at same distance. Hence, each object size has its own constants (WP x D) for measure the DPD’s and DDL’s. Below are target images at steady behavior, then we moved the camera to test whether the target tracking mechanism was worked successfully.

|  |  |  |
| --- | --- | --- |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

**FIGURE 8.** Images on same column were taken continuously and each column shows tracking a different object.

Each images at same column on figures 6 were taken and measured continuously to examine the tracker ability to track same object when camera has moved as shown on figure. The measured distance of targeted object will be used as controller parameters for quadcopter approach them.

We programmed quadcopter’s control system to continuously switch between PID to PN and vice versa. The measured distance determined which control system will be used. Simply, the PN controller active as long as the distance to target still haven’t hit the threshold for the controller switched to PID controller and vice versa. The measured distance also used as PID control’s feedback. After the distance has been stabilized by PID, then it indicates the control system has been successfully reach the target and all of quadcopter’s control params such as PID error’s is reseted to zero value. Then, quadcopter is ready for captures further target.

# CONCLUSSION

From our proposed method above, we’ve made a several conclusions. First, HoG features gives a robustness for identify the object on image but HoG cannot stand-alone as identifier because its need a classifier for distinguishing known object from its background. Moreover, HoG cannot distinguish different object with same shape such as people with their own shadow. Hence, HoG features cannot be used as a stand-alone object tracker.

The MedianFlow tracker frequently fails to track a registered object (target) when its movement on fast-moving with rapid change on its direction, and subsequently lose the target by tracking false (negative) target or totally losing sight to tracking the target even it is still on frame. However, The MedianFlow tracker works well when it is combined with features matcher as detector when the tracker tracks negative target as had explained above.

# References

1. S. A. Murtaugh and H. E. Criel, Fundamental of Propotional Navigation, *IEEE Spectrum*, (1966), pp. 75-85.
2. L. M. Argentim, W.C. Rezende, P.E. Santos and Rs. A. Aguiar, PID, LQR, and PID-LQR on a Quadcopter Platform*,* International Conference on Infromatics, Electronics, & Vision, (Dhaka, Bangladesh, 2013).
3. A. Chovancova, T. Fico, L. Chovanec, and P. Hubinsky, Mathematical Modelling and Parameter Identification of Quadrotor in Modelling of Mechanical and Mechatronics Systems, (2014).
4. M. Kumar and T. Ruoyu, Tracking of Ground Mobile Targets by Quadrotor Unmanned Aerial Vechiles*,* Unmanned System No. 2 Vol. 2*,* (2014). 157-173. <http://dx.doi.org/10.1142/S2301385014500101>.
5. N. Dalal and B. Triggs, Histograms of Oriented Gradients for Human Detection, Computer Society Conference on Computer Vision and Pattern Recognition*,* (San Diego, CA, USA, 2005). <http://dx.doi.org/10.1109/CVPR.2005.177>.
6. C. Papageorgiou and T. Poggio. A trainable system for object detection. IJCV, 38(1):15-33, 2000.
7. Pertuz, S., Puig, D., and García, M.A. (2013). Analysis of focus measure operators for shape-from-focus. Pattern Recognition, 46, 1415-1432. <http://doi.org/10.1016/j.patcog.2012.11.011>.
8. Herout, A., Hradis, M., and Zemcík, P. (2011). EnMS: early non-maxima suppression. Pattern Analysis and Applications, 15, 121-132. <http://doi.org/10.1007/s10044-011-0213-2>.
9. Guan, P., Weiss, A., Balan, A.O., and Black, M.J. (2009). Estimating human shape and pose from a single image. 2009 IEEE 12th International Conference on Computer Vision, 1381-1388. <http://doi.org/10.1109/ICCV.2009.5459300>.
10. Janku. peter, Koplik. Karel, Dulik. Tomas, and Szabo. Istvan. (2016). Comparison of tracking algorithms implemented in OpenCV. 20th International Conference on Circuits, Systems, Communications and Computers, 76. <https://doi.org/10.1051/matecconf/20167604031>.
11. Friedman, J.H., Baskett, F., & Shustek, L.J. (1975). An Algorithm for Finding Nearest Neighbors. IEEE Transactions on Computers, C-24, 1000-1006. <http://doi.org/10.1109/T-C.1975.224110>.
12. Hassaballah, M., Abdelmgeid, A.A., & Alshazly, H.A. (2017). Image Features Detection, Description and Matching.
13. Manjunath, B.S., & Ma, W. (1996). Texture Features for Browsing and Retrieval of Image Data. IEEE Trans. Pattern Anal. Mach. Intell., 18, 837-842. <http://doi.org/10.1109/34.531803>.
14. Kovesi, P. (1995). Image Features from Phase Congruency Image Features from Phase Congruency.
15. Bay, H., Ess, A., Tuytelaars, T., & Gool, L.V. (2008). Speeded-Up Robust Features (SURF). Computer Vision and Image Understanding, 110, 346-359. <http://doi.org/10.1016/j.cviu.2007.09.014>.
16. Comaniciu, D., & Meer, P. (2002). Mean Shift: A Robust Approach Toward Feature Space Analysis. IEEE Trans. Pattern Anal. Mach. Intell., 24, 603-619. <http://doi.org/10.1109/34.1000236>.
17. Bradski, G.R. (1998). Real time face and object tracking as a component of a perceptual user interface. WACV. <http://doi.org/10.1109/ACV.1998.732882>.