# Moving Objects Detection Using a Thermal Camera and IMU on a Vehicle

Kruno Lenac, Ivan Maurović, Ivan Petrović
Dept. of Control and Computer Engineering
Faculty of Electrical Engineering and Computing
University of Zagreb
Unska 3, 10000 Zagreb, Croatia
kruno.lenac@fer.hr, ivan.maurovic@fer.hr, ivan.petrovic@fer.hr

***Abstract*—In this paper we present a novel algorithm for moving object detection in thermal images taken by a moving thermal camera. It allows a detection of moving objects in thermal images of low quality without imposing restrictions on the temperature and/or shape of the object. The main assumption required for good performance of the algorithm is that the transversal movement of the vehicle will not produce significant change in the optical flow of the static objects in the scene between two consecutive image frames. Our algorithm does not use any temperature thresholds and works well in urban environments detecting moving humans and other moving objects as well. To achieve this we use fusion of an inertial measurement unit (IMU) and a thermal camera. First we use IMU data to compensate for rotational movements of the thermal camera between two consecutive thermal images. Then we differentiate those images and filter the resulting image based on dense optical flow calculated using Farneback technique. After that moving objects are detected and further filtering is applied using random sampling consensus algorithm based on optical flow model.**

***Keywords*—autonomous vehicles; sensors; thermal camera; dynamic scenes; moving object detection**

## I. Introduction

Today autonomous vehicles need to move and work alongside humans and other moving objects whose locations and trajectories are, in most cases, impossible to predict upfront. To be able to work in such dynamic environments autonomous vehicles need to be aware of their surrounding and all dynamic objects in it.

Depending on application and working conditions moving object detection problem is solved using variety of sensors and algorithms. Laser scanners are used to detect moving objects in [1] [2]. In [3] fusion of laser scanner and stereo RGB pair is used for moving object detection. In [4] [5] moving objects are detected using background subtraction techniques. Comprehensive study on vehicle detection techniques can be found in [6]. In [7] experimental survey on vision based moving objects detection and tracking can be found.

As can be seen from the above paragraph, current trends in moving object detection primarily rely on two types of sensors: RGB cameras and laser scanners. Their advantage over thermal cameras are high precision and number of details they can extract from the scene. Thermal cameras however present a unique sensor capable of giving RGB camera like data during full darkness. In general any moving object detection algorithm that works on RGB images can also be applied on thermal image. However results when applied on thermal images taken while camera is moving will in most cases be drastically worse due to following characteristics of thermal images:

- **Textureless** - objects usually have the same temperature over their entire surface.
- **High amount of noise** - results in different images even when recording static scenes.
- **Low resolution** - thermal cameras with resolutions higher than QVGA are extremely expensive.

Due to these characteristics algorithms that use image feature extraction (such as Harris corners, SIFT and SURF features) and feature matching to build models (e.g affine or bilinear) for warping previous image into current image do not work well on thermal images since there are too many false matches that are impossible to filter. Also algorithms that use dense optical flow techniques to warp images fail when applied on thermal images because outliers cannot be filtered out since there are too few accurate estimations to build good model that would be used for filtering. Because of these problems thermal cameras are almost exclusively used for moving object detection while camera is static in which case background subtraction techniques are easier to apply [8] [9].

In order to overcome disadvantages of thermal cameras but also keep their ability to distinguish objects based on temperature differences some solutions combine thermal and RGB cameras [10], [11]. Although these approaches have significantly higher accuracy of detecting moving objects compared to detection using only thermal camera they are

978-1-4673-7376-0/15/$31.00 

dependant on the light source. Some approaches [12], [13] use predefined values of temperature and/or shape of entire human body, or just the head to detect moving humans in thermal images. Although these approaches can produce good results in some situations they are highly restricted (detects only humans) and sensitive to clothing and other objects of similar temperature in the environment.

Algorithm presented in this paper uses thermal camera and IMU to detect moving objects in thermal images taken by a moving thermal camera and does not restrict the type or specific temperature range of the object (as long as temperature of the object is within camera's sensing range). Algorithm targeted usage is as an aid for the driver while driving in poor lighting conditions. There are three main stages of the algorithm:

1) Thermal image conversion and warping.
2) Background subtraction and object detection.
3) Final object filtering and detection using RANSAC.

The main assumption required for good performance of the algorithm is that transversal movement of the vehicle does not produce significant change in the optical flow of static objects in the scene between two consecutive images processed by the algorithm. Reason for this is that we do not want to impose any restrictions on the shape or temperature of a moving object. Because of that and problems with thermal images listed in the previous paragraph, warping of a previous thermal image into the current thermal image and then subtraction of those images is the only way to get initial estimate of the moving objects. As stated before warping based on image features or optical flow does not work good on thermal images. That is why we use IMU to compensate for static scene movement between two consecutive images caused by rotational ego motion of the vehicle and disregard movement caused by transversal ego motion. Of course, after subtraction of a warped image and a current image some amount of differences will exist due to transversal movement and image noise. We filter those differences out using clustering techniques coupled with dense optical flow. Although estimation of a dense optical flow on the complete thermal image results in high amount of errors we only use optical flow of pixels that remained in the image after subtraction and clustering. Main idea behind this is that pixels which remained after subtraction and clustering belong to moving objects and that optical flow calculated in them will be accurate enough for a model generation and validation. After remained pixels are grouped and assigned to objects we use RANSAC [14] to determine if optical flow of pixels contained within the object can be fitted to an optical flow model. If the fit is successful we conclude that pixels indeed belong to a moving object. Although the assumption that static scenery does not change significantly due to transversal movement between consecutive images may seam as a major restriction, many modern safety aids for the driver also have maximum velocity restriction after which they are unusable. Also, since this algorithm is mainly intended for usage on electric trams that are part of public transportation system where most accidents occur while tram is leaving the station and on crossroads in which cases speed of a tram is reduced, this restriction is acceptable.

Main novelties presented in this paper are: 1) building dense optical flow models of only preselected parts of the image and using those models in RANSAC algorithm to detect moving objects without imposing any restrictions on the shape and temperature of the object; 2) thermal image to grayscale image conversion technique which preserves thermal image details and is robust to sudden high temperature variations. To the best of authors knowledge there is no other algorithm that would allow detection of moving objects in thermal images of this quality while thermal camera is moving without imposing restrictions on temperature and/or shape of the object.

The rest of the paper is organised as follows. In Section II we explain our image conversion, warping and initial background subtraction techniques. In Section III we explain a technique based on clustering and dense optical flow that is used for final detection of moving objects. Section IV is dedicated to experimental results and in Section V we give our conclusion and plan for future work.

## II. Image enhancement, warping and subtraction

First stage of the moving object detection algorithm consists of three main steps. First, thermal image is enhanced and converted to a grayscale image, after that previous image is warped into the current image. Lastly, the current and the warped image are subtracted and the resulted image is filtered using clustering and dense optical flow.

### A. Image Conversion

Before we can process thermal images and apply any algorithms on them we need to convert them into grayscale images. The problem is that it is not trivial to determine a conversion algorithm that will transform images taken by thermal cameras into classic grayscale images. One of the mostly used ways is contrast stretching. In contrast stretching we define an interval $L = [0,\ L_{max}]$ for a converted image (defined by the number of bits used to code an intensity of every pixel) and we transform a thermal image data into that interval using equation (1).

$$\boldsymbol{I}_C(x,y) = \frac{\boldsymbol{I}_{Thermal}(x,y) - T_{min}}{Span} L_{max} \tag{1}$$

where $I_C$ is a converted image, $Span$ equals $T_{max} - T_{min}$, $T_{max}$ is the maximum value and $T_{min}$ is the minimum value in a thermal image we want to convert. There are several ways we can choose $T_{min}$ and $T_{max}$ values. Simplest one is to select the region of interest that most of thermal data falls into and fix it for all images. Problem with this approach is that thermal


Fig. 1. Comparison of conversion algorithms: a) Region based conversion; b) Adaptive $T_{min}$ and $T_{max}$ values; c) fixed $T_{min}$ and $T_{max}$ values

images which have maximum values of for example $T_{max}/2$ will only use half of the $L$ range which will result in loss of details. Another way is to find real $T_{min}$ and $T_{max}$ values for every thermal image and convert it accordingly. The advantage of this is that we can custom fit conversion of every image and we are certain that maximum value of every thermal image will have value of $L_{max}$ in the converted image. Problem with this approach arises when one smaller hot or cold object enters the scene in which case we have the same situation as with fixed values of $T_{min}$ and $T_{max}$. Solution could be to limit $T_{min}$ and $T_{max}$ numbers and consider all the values higher than $T_{max}$ as $T_{max}$ and all the values lower than $T_{min}$ as $T_{min}$, but in that case if more hot objects enter the scene we lose distinctiveness between them. In order to solve those problems we have implemented region based conversion algorithm. We have selected the range of values ($Span$) in thermal images which covers almost entire relevant spectrum recorded by our camera and divided it into 10 regions. After thermal image is acquired we first determine how many pixels belong to each region. Based on the number of pixels, every region is assigned with a portion of interval $L$. When this is done all pixels belonging to one region are stretched over the interval assigned to that region using equation (2). This is done for all the regions and the result is a grayscale image. This approach ensures that $T_{max}$ value in every thermal image will get value of $L_{max}$ in grayscale image, and also ensures that if only small amount of pixels have significantly higher values than the rest they will be assigned only small part of $L$.

$$\boldsymbol{I}_C(x,y) = \frac{\boldsymbol{I}_T(x,y) - T_{min} - n * R_{size}}{R_{size}} Rn_{int} + Rn_{start} \tag{2}$$

where $R_{size} = Span/10$, $n$ is the number of current region $n$ element $[0, ..., 9]$, $Rn_{int}$ is interval assigned to the $n$-th region and $Rn_{start}$ is the lower limit of $Rn_{int}$. Fig. 1 shows the comparison between our region based conversion (a), the conversion based on adaptive $T_{min}$ and $T_{max}$ values (b) and the conversion based on fix values of $T_{min}$ and $T_{max}$ (c). It is obvious that the region based conversion is much more distinctive than two other techniques. It is also important to note that since we do not impose any restrictions on interval of object temperature we did not use any conversion formulas to get real temperature data (i.e. Celsius of Fahrenheit) but instead we used raw data recorded by a camera sensor. This reduces a noise and detail loss during a conversion.

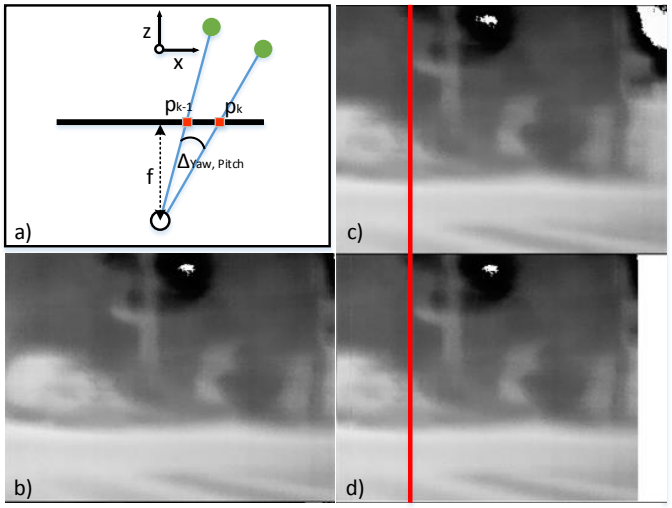


Fig. 2. a) Warping of individual pixels; b) Image from step k-1; c) Image from step k; d) Warped image

### B. Warping

Warping is used to account for relative motion of static objects between two consecutive images that resulted from vehicle rotation and vibration. Translational movement between the image frames is considered small enough to be neglected. Rotation angles (yaw and pitch) between two consecutive images are measured with IMU. Yaw angle is used to compensate for intentional vehicle rotation and pitch angle is used to compensate for camera vibrations. If we consider vehicle rotating while recording images and if we know exact angle of rotation between images $\boldsymbol{I}_{k-1}$ and $\boldsymbol{I}_k$ we can compute warped image $I_{warp}$ using the following equations (Fig. 2a):

$$\begin{aligned} p(x) &= \frac{f}{dp}\tan(\Delta_{Yaw} + atan2(((x - \boldsymbol{I}_w/2) * dp), f)) \\ p(y) &= \frac{f}{dp}\tan(\Delta_{Pitch} + atan2(((y - \boldsymbol{I}_h/2) * dp), f)) \\ \boldsymbol{I}_{warp}(p(x), p(y)) &= \boldsymbol{I}_{k-1}(x,y) \end{aligned} \tag{3}$$

where $\Delta Yaw$ and $\Delta Pitch$ are vehicle rotational movements between consecutive time steps measured by IMU, $(x, y)$ are pixel coordinates of the image $I_{k-1}$ ($x$ is column number, $y$ is row number and (0,0) is top left corner), $(p(x), p(y)$ are coordinates of the warped pixels, $f$ is camera's focal length, $dp$ is a pixel width, $\boldsymbol{I}_w$ is an image width and $\boldsymbol{I}_h$ is an image height. Fig. 2 (b-d) shows the result of warping. We can see that original image (Fig. 2c) and warped image (Fig. 2d) are almost completely aligned.

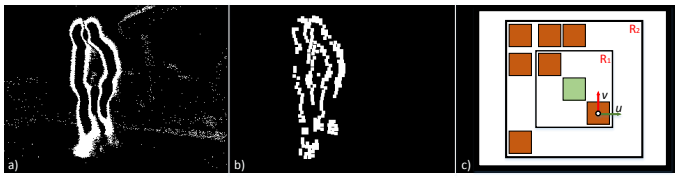


Fig. 3. a) Subtracted image; b) Subtracted image after filtering; c) Pixels filtering algorithm

### C. Image Subtraction and Filtering

Once warped image has been acquired we can compute subtracted image $\boldsymbol{I}_S$ from $\boldsymbol{I}_k$ as

$$\boldsymbol{I}_S(x,y) = \begin{cases} L_{max}, & \text{if } \boldsymbol{I}_k(x,y) - \boldsymbol{I}_{warp}(x,y) \geq D_{thr} \\ 0, & \text{otherwise} \end{cases} \quad (4)$$

where $D_{thr}$ is a predefined threshold used to filter small differences that are mostly produced by noise. Example of $\boldsymbol{I}_S$ is shown in Fig. 3a. In order to filter out pixels that remained in $\boldsymbol{I}_S$ due to noise and transversal movement, a filtering based on clustering and optical flow is performed. Dense optical flow is calculated between $\boldsymbol{I}_{warp}$ and $\boldsymbol{I}_k$ using Farneback algorithm [15]. Filtering is based on grouping pixels in regions of 5x5 by analyzing pixels neighbourhood. Neighbourhood is divided into two parts $R_1$ (8 pixels) and $R_2$ (16 pixels) as shown in Fig. 3c. For every pixel in $\boldsymbol{I}_S$ whose value is larger than 0 steps listed in algorithm 1 are performed. Pixels that already

**Algorithm 1** Grouping of pixels into 5x5 regions

1: Count the number of pixels ($N_{R1}$, $N_{R2}$) with positive values in $R_1$ and $R_2$
2: Calculate the sum of optical flows $\vec{U}_s$ and $\vec{V}_s$ in $\vec{u}$ and $\vec{v}$ directions of all pixels with positive values in $R_1$ and $R_2$
3: Check if $N_{R1} \geq N_{thr1}$ and $N_{R2} \geq N_{thr2}$ ($N_{thr1}$ and $N_{thr2}$ are predefined thresholds)
4: If previous condition is satisfied check if $|\vec{U}_s| \geq U_{thr}$ and $|\vec{V}_s| \geq V_{thr}$ ($U_{thr}$ and $V_{thr}$ are predefined thresholds)
5: If both conditions are satisfied set all pixels in this 5x5 region to $L_{max}$, otherwise set all to 0

belong to one 5x5 region that was filled cannot be a part of another region. One example of this filtering technique applied on image shown in Fig. 3a is presented in Fig. 3b. After the filtering of $\boldsymbol{I}_S$, the resulted image $\boldsymbol{I}_{SF}$ serves as a base for final moving object detection.

## III. Moving Object Detection

Final object detection is done in two stages. First 5x5 regions are grouped and assigned to objects and then RANSAC is used to determine which of those objects are true moving objects and which are faulty detections.

### A. Region Grouping

Image $\boldsymbol{I}_{SF}$ still contains only pixels grouped in regions of 5x5 without any information on which region belongs to which object. Before assigning regions to particular objects in order to speed up the process and make it more robust we do another grouping of 5x5 regions into 10x10 regions. This is done using a sliding window approach on image $\boldsymbol{I}_{SF}$. Created 10x10 regions are saved in matrix $\boldsymbol{R}$. Sliding window size is 10x10 and sliding process starts at the top left corner of $\boldsymbol{I}_{SF}$. Algorithm than sums optical flows in $\vec{u}$ and $\vec{v}$ directions of all 5x5 regions whose centres are located within the sliding window. If the sums in any direction are different than 0, the algorithm marks current location of the sliding window as filled in matrix $\boldsymbol{R}$ and records the information on sums of optical flows in both directions. Since dimensions of sliding window are 10x10 dimensions of matrix $\boldsymbol{R}$ are $I_h/10$ x $I_w/10$. All positive elements in matrix $\boldsymbol{R}$ potentially belong to a moving object. Grouping of individual elements into one object is done using the algorithm 2. The result of the algorithm

**Algorithm 2** Creating objects from elements

1: Find positive element $e$ in $\boldsymbol{R}$
2: If $e$ is not part of another object create new object $O$ containing element $e$
3: Search neighbourhood of $e$ and add all positive elements to $O$ and to stack $S$ if they do not belong to another object
4: For every element in $S$ repeat step 3
5: When $S$ is empty go to step 1 and save object $O$

*Neighbourhood of an element $e_{i,j}$ in matrix $R$ are all elements $e_{n,m}$ that satisfy conditions: 1) $|n-i| \leq 2$ and 2) $|m-j| \leq 2$*

is a list of potential moving objects. Before a final object filtering using RANSAC, objects in the list are filtered based on their width, height and density. Minimum values for width and height are used to filter out objects that are either false detections or are too small or too far away to be of importance. Maximum values are used to filter objects that span over more than 70% of the image width or more than 85% of the image height and are probably results of vibrations that were unable to be corrected by the IMU. Width $O_w$ and height $O_h$ of an object containing $N$ elements $e_{i,j}$ are $O_w = j_{max} - j_{min}$ and $O_h = i_{max} - i_{min}$. Density of an object is calculated as $Density = N/(O_w O_h)$. All objects that have height and width within predefined intervals $[H_{min},\ H_{max}]$ and $[W_{min},\ W_{max}]$ and whose density is larger than predefined value $DENS_{min}$ are added to a list $L_O$ and passed onto the final filtering step using RANSAC algorithm.

### B. RANSAC Filtering

RANSAC algorithm is used for final filtering of detected objects in the list $L_O$ based on an optical flow model derived

in [15]. Model is given in equation (5).

$$\begin{aligned} d_u(x,y) &= a_1 + a_2x + a_3y + a_7x^2 + a_8xy \\ d_v(x,y) &= a_4 + a_5x + a_6y + a_7xy + a_8y^2 \end{aligned} \tag{5}$$

where $d_u$ and $d_v$ are the amplitudes of optical flows in $\vec{u}$ and $\vec{v}$ directions of a pixel located in image coordinates $(x, y)$ and $a_1$ to $a_8$ are model parameters that need to be evaluated. Model (5) can be rewritten as:

$$\boldsymbol{d} = \boldsymbol{S}\boldsymbol{p} \tag{6}$$

$$\boldsymbol{S} = \begin{pmatrix} 1 & x & y & 0 & 0 & 0 & x^2 & xy \\ 0 & 0 & 0 & 1 & x & y & xy & y^2 \end{pmatrix} \tag{7}$$

$$\boldsymbol{p} = (a_1\ a_2\ a_3\ a_4\ a_5\ a_6\ a_7\ a_8)^T \tag{8}$$

Now that we have the model we can use RANSAC algorithm. There are two main steps of RANSAC algorithm:

1) Create model based on random sampling.
2) Evaluate the model.

Our goal is to create a model for every object in $L_O$ from optical flows of randomly selected pixels belonging to those objects and then, in evaluation step, to determine if optical flow fits the model and really belongs to a moving object. If the flow is too random to be fitted to a model we conclude it is not a real moving object. In order to get model parameters ($a_1$ to $a_8$) we define the least squares problem

$$\begin{aligned} &\sum_{i=0}^{N_e} ||\boldsymbol{S}_i\boldsymbol{p} - \boldsymbol{d}_i||^2 \\ &\boldsymbol{d}_i = \begin{bmatrix} |\vec{u_i}| \\ |\vec{v_i}| \end{bmatrix} \end{aligned} \tag{9}$$

where $N_e$ is the number of pixels randomly selected for model generation and ($\vec{u}_i$, $\vec{v}_i$) are optical flow vectors of the $i$-th pixel. Solution to the problem (9) is:

$$\boldsymbol{p} = (\sum_{i=0}^{N_e} \boldsymbol{S}_i^T\boldsymbol{S}_i)^{-1} \sum_{i=0}^{N_e} \boldsymbol{S}_i^T d_i \tag{10}$$

Once the model is generated, we can evaluate how many pixels that were not used in model generation fit into the model using $||\boldsymbol{S}_n\boldsymbol{p} - d_n||^2$. If the ratio of the number of pixels that fit into the model and the number of all the pixels contained within the object is larger than predefined threshold we say that model is good enough and that the object it belongs to is indeed a moving object. If model does not satisfy this, another iteration of RANSAC algorithm starts and a new model is created based on newly and randomly selected pixels. If after $N_{it}$ iterations, none of the generated models satisfies required ratio, the object is discarded and not reported as a moving object. Pixels used in RANSAC model in our case are represented by 5x5 regions contained within all 10x10 regions that particular object consists of. Number of regions $Nm$ selected for model generation is computed as:

$$Nm = \begin{cases} \frac{N_{reg}}{3}, & \text{if } \frac{N_{reg}}{3} \leq Nm_{max} \\ Nm_{max}, & \text{otherwise} \end{cases} \tag{11}$$

where $N_{reg}$ is the total number of 5x5 regions contained within the object and $Nm_{max}$ is a predefined threshold. Finally all the objects that have passed RANSAC filtering are reported as moving objects in the scene. Since we do not track detected objects we did not use any technique to ensure that entire area of the object is detected since for this application it is not required. Fig. 4 shows the entire process from grouping 5x5 regions into 10x10 regions to a final display of detected moving object. Steps that sum up the entire process of moving object detection are listed in algorithm 3.

**Algorithm 3** Moving object detection algorithm

1: Acquire the thermal image $IT_k$
2: Convert $IT_k$ to grayscale image $I_k$ using region based contrast stretching algorithm
3: Warp the image $I_{k-1}$ according to differences in Yaw and Pitch angle between steps $k$ and $k-1$ that are measured by IMU
4: Calculate dense optical flow between warped image and $I_k$
5: Get $I_S$ by subtracting warped image from $I_k$
6: Filter pixels of $I_S$ based on their neighbourhood and optical flow and group remaining pixels into 5x5 regions
7: Group 5x5 regions into 10x10 regions using sliding window approach and connect 10x10 regions into objects
8: Filter objects based on width, height and density
9: Filter remaining objects using RANSAC and optical flow model
10: Display detected moving objects in grayscale image $I_k$

## IV. Experimental Results

Experiments were conducted using Husky A200 mobile platform equipped with FLIR A320 thermal camera and Xsens MTI-G 700 IMU. Images acquired with Flir A320 have resolution of 320x240, maximum frame rate of 30Hz, horizontal field of view $25°$ and vertical field of view $18.8°$. Wavelengths captured by the camera sensor are between $7.5\mu m$ and $13\mu m$. Maximum speed of Husky A200 is 1.5m/s. Values of all algorithm parameters are listed in the Table I. Detection algorithm was running on Fujitsu H series notebook with Intel i7-3630QM mobile processor.

Results are displayed in Figs. 5 to 10. First image of every sequence displays the objects that are detected. Fig. 5 displays the detection of a moving car that is far away from the camera while Husky is moving forward. Fig. 6 shows continuous detection of a human who is walking while robot is

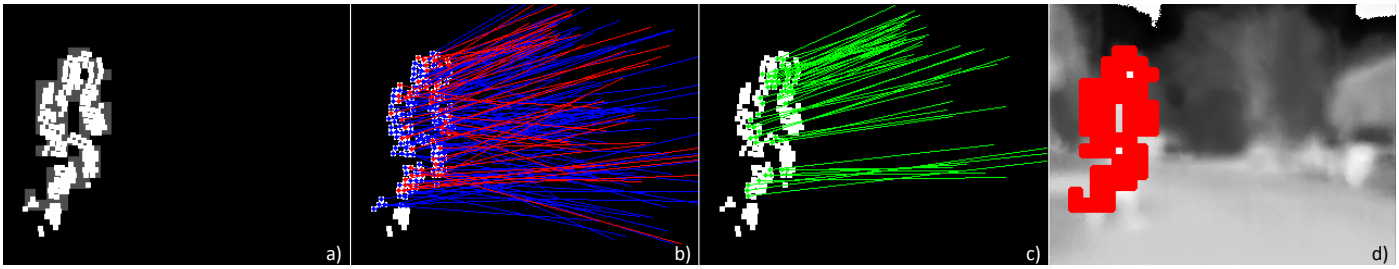


Fig. 4. Final moving object detection: a) Grouping of 5x5 regions from filtered subtracted image into 10x10 regions and connecting them into objects b) Optical flow of all 5x5 regions contained within the object (blue) and optical flow selected for model generation (purple); c) Optical flow that was successfully fitted to the generated model; d) Object displayed in the grayscale image

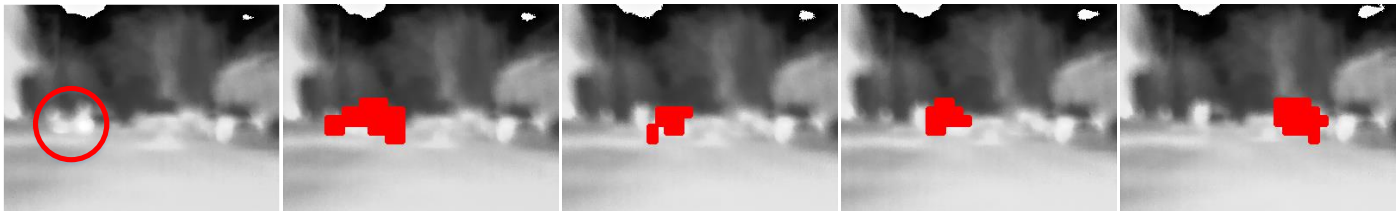

Fig. 5. Detection of a car while moving forward

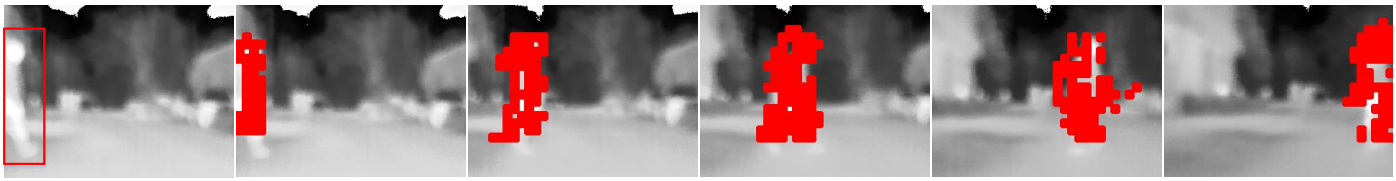

Fig. 6. Human walking detection while rotating and moving forward

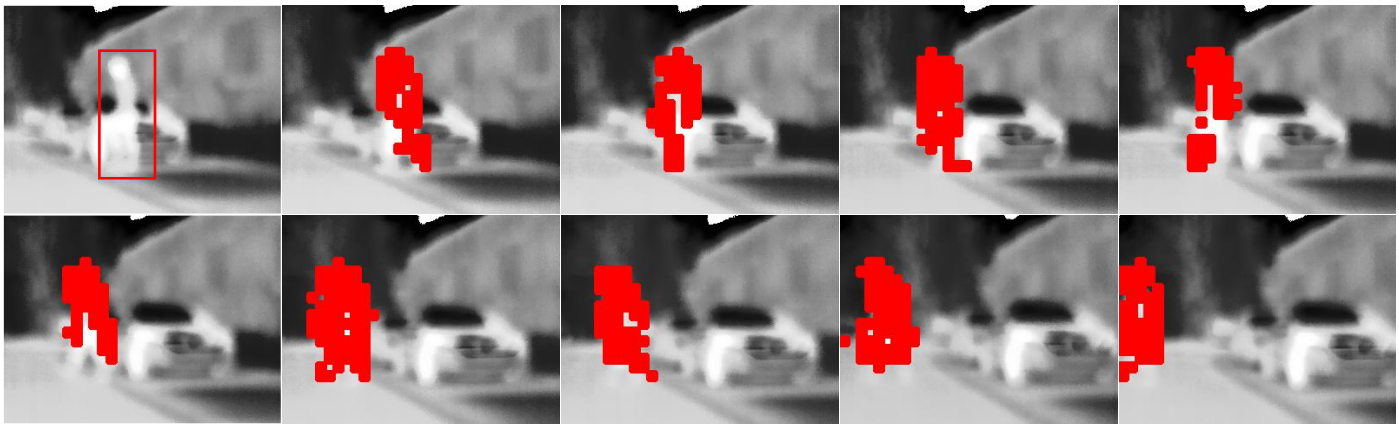

Fig. 7. Continuous detection of a walking human while moving forward

we

rotating to the left and moving forward at the same time. Fig. 7 shows continuous detection of a human who is walking while robot is moving forward. In Fig. 8 a human is detected while walking. At image 3 of the sequence a car enters the scene

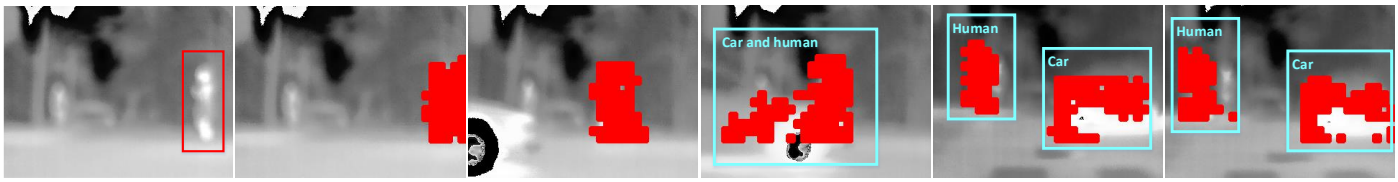


Fig. 8. Two objects detection while standing still

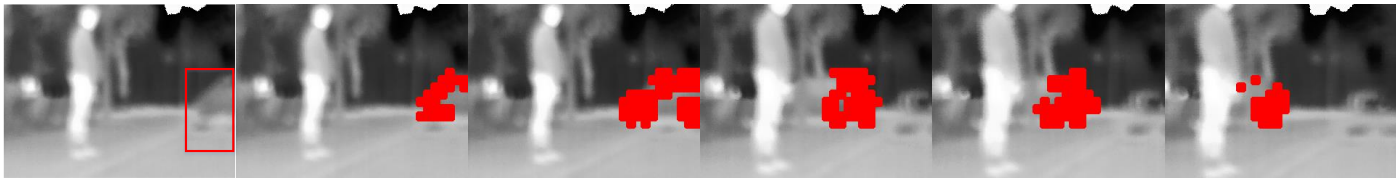

Fig. 9. Detection of a moving car and no detection of human who is standing still while Husky is moving forward

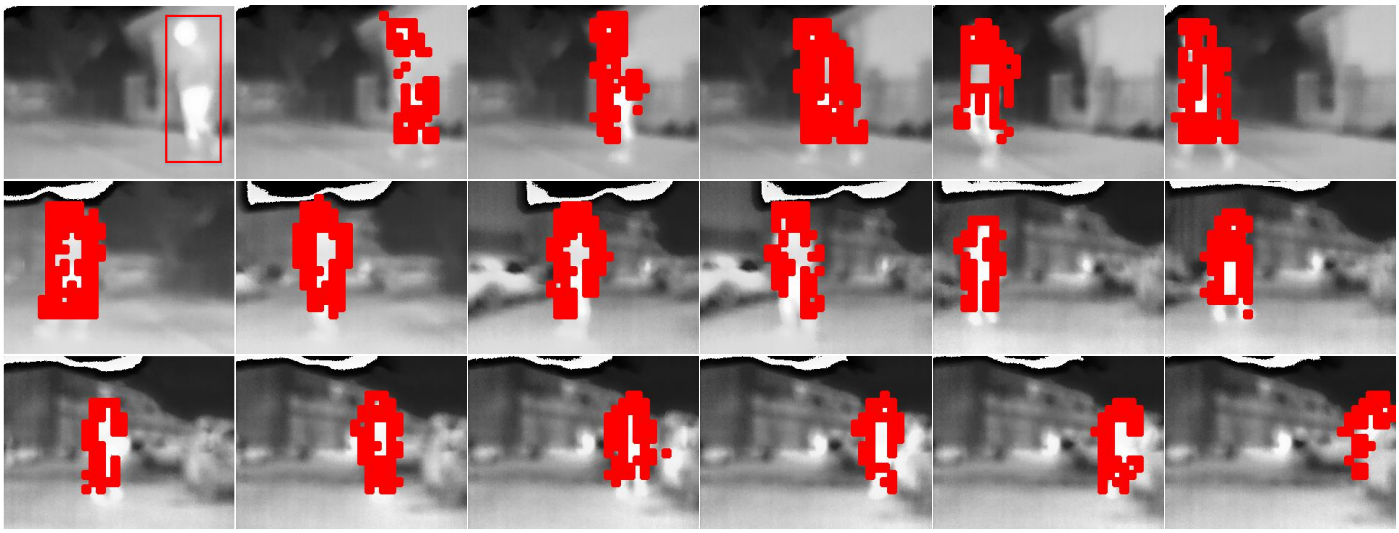

Fig. 10. Continuous detection of a human while Husky is moving forward and rotating

and covers the human. After the human is visible again both the car and the human are detected as two moving objects. Fig. 9 shows detection of a moving car and it can also be seen that human who is standing in place is not detected although Husky is mowing forward. Fig. 10 shows successful continuous detection of a human while Husky is rotating and moving forward. As can be seen from the results, the algorithm performs quite well in different scenarios. However there are some fault detections. Most of them occur because noise in the image resulted in the subtracted image containing too many pixels that were inaccurately set to positive values and at the same time optical flow was incorrectly estimated resulting with error in detection. Also in some cases objects are not detected instantly upon entering the scene due to inaccurate optical flow estimation.

TABLE I. List of used parameters and their values

| Parameter | Value | Parameter | Value | Parameter | Value |
|---|---|---|---|---|---|
| $T_{min}$ | 16500 | $T_{max}$ | 21500 | $Span$ | 5000 |
| $f$ | 18mm | $dP$ | 24.8$\mu m$ | $D_{thr}$ | 3000 |
| $N_{thr1}$ | 2 | $N_{thr2}$ | 4 | $U_{thr}$ | 100 |
| $V_{thr}$ | 100 | $H_{min}$ | 3 | $H_{max}$ | 21 |
| $W_{min}$ | 3 | $W_{max}$ | 21 | $DENS_{min}$ | 0.3 |
| $N_{it}$ | 10 | $Nm_{max}$ | 90 | $L_{max}$ (16 bits) | 65535 |

## V. Conclusion

In this paper we have presented a novel approach for detecting moving objects in thermal images. When using thermal

images main problems occur from the fact that most objects have similar temperatures all over their surfaces. Because of that, the images captured by thermal cameras are almost textureless and provide very little details that would allow efficient feature matching. We have solved this problem by using fusion of IMU and thermal camera in order to track ego motion of the robot while rotating. This has allowed us to implement image subtraction between consecutive images and by using filtering techniques based on clustering and dense optical flow estimation differentiate moving objects from static scenery in the subtracted images. We have shown by experiments done in real world scenarios that our algorithm works well, but also still has room for improvement. In the future we plan to implement tracking algorithm that would allow further filtering of faulty detections. Also we plan to reduce the number of parameters required for algorithm to work and to improve optical flow estimation algorithm in order to account for specific properties of thermal images.

## ACKNOWLEDGMENT

This work has been partly supported by the European Regional Development Fund under the project „Advanced technologies in power plants and rail vehicles“.

## REFERENCES

[1] T. Mori, T. Sato, H. Noguchi, M. Shimosaka, R. Fukui, and T. Sato, “Moving objects detection and classification based on trajectories of LRF scan data on a grid map,” *IROS*, pp. 2606–2611, 2010.

[2] H. Zhao, J. Sha, Y. Zhao, J. Xi, J. Cui, H. Zha, and R. Shibasaki, “Detection and Tracking of Moving Objects at Intersections Using a Network of Laser Scanners,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 13, no. 2, pp. 655–670, 2012.

[3] Q. Baig, O. Aycard, T. D. Vu, and T. Fraichard, “Fusion between laser and stereo vision data for moving objects tracking in intersection like scenario,” *IEEE Intelligent Vehicles Symposium, Proceedings*, no. Iv, pp. 362–367, 2011.

[4] F. Cheng, S. Huang, and S. Ruan, “Illumination-sensitive background modeling approach for accurate moving object detection,” *IEEE Transactions on Broadcasting*, vol. 57, no. 4, pp. 794–801, 2011.

[5] H. Yang, J. Tian, Y. Chu, Q. Tang, and J. Liu, “Spatiotemporal smooth models for moving object detection,” *IEEE Signal Processing Letters*, vol. 15, pp. 497–500, 2008.

[6] S. Sivaraman and M. M. Trivedi, “A review of recent developments in vision-based vehicle detection,” *IEEE Intelligent Vehicles Symposium*, no. Iv, pp. 310–315, 2013.

[7] A. W. M. Smeulders, D. M. Chu, R. Cucchiara, S. Calderara, A. Dehghan, and M. Shah, “Visual tracking: An experimental survey,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 36, no. 7, pp. 1442–1468, 2014.

[8] Y. Zhang, T. Zhao, J. Gu, and S. Yu, “Accurate moving object detection in thermal imagery,” *CSAE*, vol. 3, pp. 282–286, 2011.

[9] J. W. Davis and V. Sharma, “Background-subtraction in thermal imagery using contour saliency,” *International Journal of Computer Vision*, vol. 71, no. 2, pp. 161–181, 2007.

[10] A. Königs and D. Schulz, “Fast visual people tracking using a feature-based people detector,” *IROS*, pp. 3614–3619, 2011.

[11] S. Kumar, T. Marks, and M. Jones, “Improving Person Tracking Using an Inexpensive Thermal Infrared Sensor,” *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pp. 217–224, 2014.

[12] W. K. Wong, Z. Y. Chew, C. K. Loo, and W. S. Lim, “An effective trespasser detection system using thermal camera,” *ICCRD*, pp. 702–706, 2010.

[13] S. Zubaidah, B. Rajemi, R. Hamid, F. Naim, and N. W. Arshad, “Pedestrian Detection for Automotive Night Vision Using Thermal Camera,” *ICCCT*, pp. 694–698, 2012.

[14] M. A. Fischler and R. C. Bolles, “Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography,” *Commun. ACM*, vol. 24, no. 6, pp. 381–395, Jun. 1981.

[15] G. Farnebäck, *Two-Frame Motion Estimation Based on Polynomial Expansion*. Springer Berlin Heidelberg, 2003, vol. 2749.

**K. Lenac** was born on May 19th 1989. He received his B.Sc. degree in 2011. and M.Sc. degree in 2013., both in Electrical Engineering from Faculty of Electrical Engineering and Computing (FER Zagreb), University of Zagreb, Croatia. He has been employed as a researcher on the ACROSS project in October 2013 at FER. At the same time he became a Ph.D. student at FER under mentorship of prof. dr. sc. Ivan Petrović. Currently he is employed as a researcher on FER-KIET project. His main areas of interest is mobile robotics with focus on autonomous localization, navigation and map building.

**I. Maurović** was born on November 14, 1986 in Pula. He finished undergraduate study and awarded academic degree Bachelor of Science in 2008 and received his M.Sc. degree in 2010 in Electrical Engineering from Faculty of Electrical Engineering and Computing, University of Zagreb, Croatia. In 2010 he joined the Department of Control and Computer Engineering as a research assistant on SEE-ERA.NET PLUS project ThermalMapper and enrolled postgraduate study. From 2012 he was employed as a researcher on ACROSS project and currently he is working as a researcher on FER-KIET project. His main area of interest are mobile robotics, exploration of unknown environmets and path planning.

**I. Petrović** received B.Sc. degree in 1983, M.Sc. degree in 1989 and Ph.D. degree in 1998, all in Electrical Engineering from the Faculty of Electrical Engineering and Computing (FER Zagreb), University of Zagreb, Croatia. He had been employed as an R&D engineer at the Institute of Electrical Engineering of the Končar Corporation in Zagreb from 1985 to 1994. Since 1994 he has been with FER Zagreb, where he is currently a full professor. He teaches a number of undergraduate and graduate courses in the field of control systems and mobile robotics. His research interests include various advanced control strategies and their applications to control of complex systems and mobile robots navigation. He has published more than 40 journal and 160 conference papers, and results of his research have been implemented in several industrial products. He is a member of IEEE, IFAC - TC on Robotics and FIRA - Executive committee. He is a member of the Croatian Academy of Engineering.