Rule-Based Multiple Target Monitoring in Robot Vision Using Partitioning

Davood Pour Yousefian Barfeh*,1,2,3, Terrence A Lim1, Editha C Jose2, Felino P Lansigan2, Eduardo R Mendoza1 and Vladimir Y Mariano5

1Lyceum of the Philippines University-Laguna, Calamba City, Philippines  
2University of the Philippines-Los Bosños, Laguna, 4031, Philippines  
3Adamson University, 1000 Manila, Philippines  
4Ludwig Maximilian University, 80539 Munich, Germany  
5RMIT University, 702 Nguyen Van Linh, HCMC Vietnam

* davoodpb@gmail.com

Abstract. In this study, the researchers propose an algorithm and implement it as a robot-camera which not only detects many moving objects and captures them, but also follows the moving objects to continue capturing and monitoring. The robot uses Gaussian filtering to detect all motions and then applies partitioning on the scene to obtain location of all moving objects in three major areas and different sub-areas. Based on the partitioning and position of the moving objects, the robot might be in any of unsafe state, safe state, and over-safe state. Regarding each state, the robot performs proper movement following two cardinal rules to pursue monitoring the moving objects.

1. Introduction
There are three levels of algorithms in computer vision and digital image processing: low level vision algorithms, intermediate level vision algorithms, and high level vision algorithms [1–6]. As the names of levels show, the algorithms in high level are more difficult and more important, since these algorithms deal with perception and the context. Indeed, in these algorithms, the input and output of systems are symbolic representation of images’ features [1–6]. The best instances of these algorithms can be found in the fields of target (moving object) tracking and motion detection in which the aim is to detect motion and moving object as target or to find the track of the target [1][6]. Beside importance and significance, there are several complications in target tracking or motion detection such as; occlusion, changes in illumination, change in background, clutter, occurrence of shadow and noisy video [7]. Regardless of these complications and challenges, motion detection and target tracking are interesting topics in computer vision and digital image processing, since they are bases in numerous fields such as supplying advanced interfaces between humans or between humans and devices, medicine, sports, military, and especially PTZ (Pan-Tilt-Zoom) or smart cameras [7–12].

2. Review of related studies and products
Numerous studies and researches are performed in the field of motion detection, target tracking and capturing. Moreover, there are several products in industry regarding these fields. All of these studies and products can be classified into four categories.
Category-A consists of studies and researches which are not real time, but systems and algorithms try to detect moving objects and then track the targets in recorded videos [13–19].

Category-B consists of studies and researches that work real time. Systems in these studies detect and track many targets using stationary cameras [20–22].

Category-C consists of studies performed based on a moving camera. Here, the systems focus on detecting and tracking of targets while the camera has a random movement [23–25]. The movement of camera is not dependent on motion of targets. In studies in Category-C same as the studies in Category-A and Category-B, the term “track” means that path of moving objects.

Category-D consists of industrial works and products in which the algorithms and systems use non-stationary cameras to detect and track moving object [26–28]. Systems in these products not only detect moving object, but also track, monitor and follow the moving object either by focal zooming or by movement. Products and systems of this category are able to detect, track, monitor and follow only one moving object. In Category-D, the term “track” means that following and monitoring of the moving object.

3. Statement of the problem
Most of the time, when it is talked about motion, there are many targets in the scene. In other words, there are many objects which are moving – not only one object. Therefore, we need a multiple moving object monitoring system that can capture and monitor many targets.

After considering all categories of researches, studies and products, the researchers realized that there is no algorithm or system to monitor and follow many targets. In fact, regarding monitoring and following target, the best studies are placed in Category-D in which systems are able to monitor and follow only one target, not more.

4. Proposed system
To solve the problem mentioned, the researchers propose a rule-based multiple target monitoring (RBMTM) system which is a robot system having a camera. RBMTM not only detects multiple moving object, but also moves forward, backward, left and right to continue capturing and monitoring of the moving objects.

In this study, researchers particularly design and implement an algorithm as a simple robot that;

a) Performs multiple moving object detection (motions detection), and
b) Tries to maintain monitoring of the moving objects using proper movements to right, left, forward and backward.

4.1. Motions detection
There are several methods and algorithms for motion detection. All of these methods work based on the comparison of the current video frame with one (or more) frame from the previous frame (or frames) [29].

The system RBMTM uses Gaussian filtering for motions detection to benefit simplicity and rapidity [29–32]. In digital image processing, Gaussian filtering (Gaussian smoothing) is the outcome of blurring a digital image using a Gaussian function that is extensively used in graphics tools to alleviate noise of image. Moreover, Gaussian function is used to reduce some details in image. In any case, the effect of the Gaussian function is generating a smooth blur resembling image [29–33].

4.2. Movement
After capturing the scene as a stream of images and detecting motions, the system RBMTM may need to move. The movement is the effort of the system to continue capturing and monitoring of moving objects. In fact, the system tries to chase all of the moving objects using movement. In other words, the system may move forward, backward, right or left to maintain capturing all targets in the scene. In this regard, the scene is divided into three major areas; unsafe area, safe area and over-safe area as shown in figure 1.
There are two parts inside the safe area; right side and left side, each of which has two parts. Totally, safe area has four different sub-areas; sLL (safe, left side of left side), sLR (safe, right side of left side), sRR (safe, right side of right side) and sRL (safe, left side of right side) as shown in figure 2.

There are also four different sub-areas inside unsafe area; uRR (unsafe, right side of right side), uLL (unsafe, left side of left side), uLR (unsafe, right side of left side), and uRL (unsafe, left side of right side) as shown in figure 2.

![Figure 1. Location of three major partitions in scene](image1)

![Figure 2. Different sub-areas inside unsafe area and safe area](image2)

Based on the partitioning, all movements of the system are based on two rules regarding their priorities;

R1) Keep all moving objects in the safe area, and
R2) Keep all moving objects in the center of the scene (not left side and not right side).

R1 that is always prior to R2, consists of two sub-rules regarding their priorities;

R1-a) Not to keep any moving object in unsafe area – for not losing the moving object, and
R1-b) Not to keep any moving object in over-safe area – for capturing and monitoring the target better, having more details.

R1 says when some moving object is inside unsafe area, there is a danger that the moving object gets out of the field of view, and in result the target gets lost, since the system cannot capture it anymore. On the other hand, when all moving objects are inside over-safe area, system moves towards the moving objects and gets closer to them such that the system captures and monitors the moving objects with more details.

When R1 is satisfied – no target is inside unsafe area and not all targets are inside over-safe area, R2 says if all moving objects are inside only one of the four sub-areas of safe area, then the system moves such that all targets are located inside the center of the scene. In fact R2 immigrates fairness to the system giving equal chance of monitoring to all moving objects that will come inside the scene from different directions.

Generally based on partitioning and location of the moving objects, the system can be at one and only one of three modes; unsafe mode, safe mode and over-safe mode.

4.3. Unsafe mode
Unsafe mode is the situation in which there exists at least one moving object in unsafe area. In this case, system has to move backward, move to right or move to left to follow the R1-a. In fact, the system using movement decreases the danger of losing the moving objects via not including them in unsafe area.

4.4. Safe mode
Safe mode is the situation in which there is no motion in the scene or there is at least one moving object in safe area while there is no moving object in unsafe area. System is said to be in SSNM (safe mode and no move) when;

1) The system is in safe mode and there exists at least one moving object in the left side of safe area and at least one target in the right side of safe area, or
2) The system is in safe mode and there is no moving object in the left side of safe area and there is no moving object in the right side of safe area.

4.5. Over-safe mode
The system is said to be in over-safe mode when the system is neither in unsafe mode nor in safe mode. In this situation, R1-a is already followed, and the system moves forward to follow R1-b. In fact, system increases the details of monitoring via moving forward and getting closer to the moving objects.

4.6. Unlimited backward movement
As mentioned earlier in video processing, all methods in motion detection including Gaussian filtering work based on comparing of the current video frame with one or more frames from the previous frame that is called as background. It means that, if there is some change in some pixels in two consequent frames, then the system considers this change as a motion. On the other hand, when system moves, almost entire scene changes and in result pixels of almost all three major areas (unsafe area, safe area and over-safe area) of consequent frames change. In this situation, RBMTM presumes to have motions in all areas including sub-areas of unsafe area. Hence, RBMTM proceeds to unsafe mode and has to perform backward movement to change the situation from unsafe mode to safe mode, but the result of movement is unsafe mode again. In other words, with first movement, RBMTM is fooled and gets inside an unlimited loop of unsafe mode and then backward movement.

To omit the problem of unlimited backward movement, RBMTM does not perform motions detection during its movement.

4.7. Algorithm
When RBMTM starts capturing and monitoring, value of the variable “target” is false, since the system did not detect any moving object yet. The function detect_motions is responsible to detect all moving objects among all areas in the scene. This function changes the value of the variable target to true if there is a motion. The function detect motions also returns the area and sub-area of all moving objects. Then the system discovers its mode based on the areas and location of the moving objects. If the system is in unsafe mode, based on R1-a, system has to change the mode to safe mode using move_right, move_left, or move_backward.

If the system is in SSM, R1 is already followed, but based on R2, system has to change the mode to SSNM using move_right or move_left.

If value of variable target is true, and RBMTM is neither in unsafe mode nor in safe mode, then the system is in over-safe mode. In this situation, based on R1-b, system moves and gets closer to the moving objects using move_forward to change the mode to the safe mode.

The overall algorithm for system RBMTM is as following:
```
detect_motions()
while (target){
if (unsafe_state){
if (uLL & NOT sRR & NOT sRL)
{move_left twice;}
elseif (uRR & NOT sLL & NOT sLR)
{move_right twice;}
elseif (uLR & NOT sRR)
{move_left;}
elseif (uRL AND NOT sLL)
{move_right;}
else {move_backward; exit;} // end of unsafe
}
else if (safe_state){
if ((sRR OR sRL) AND NOT sLL AND NOT sLR)
{move_right;}
elseif ((sLL OR sLR) AND NOT sRR AND NOT sRL)
{move_left; exit;} // end of safe
move_forward; // over-safe state
}
}
```

5. Implementation
From hardware view, the system RBMTM is a simple robot consisting a Raspberry Pi 3 Model B with a camera module v1, two servo motors (TowerPro SG5010) each of which connected to a wheel, power supplies (for Raspberry Pi and servo motors) and a free wheel in the front, while from software
view, RBMTM uses Raspbian Jessie as operating system, OpenCV as library, and Python as programming language.

5.1. Partitioning
RBMTM uses ratios of width and ratios of height of the image (frame) to divide the scene to different areas and sub-areas. Each image of the scene is divided to many parts vertically based on height of the frame and horizontally based on width of the frame.

5.2. Motions detection
The system benefits Gaussian filtering to detect all moving objects. The system uses function “GaussianBlur”, for Gaussian filtering, and then function “accumulateWeighted” and function “absdiff” to accumulate the weighted average between the current frame and previous frames due to calculate the difference between the current frame and running average frame. Then the system uses functions “threshold” and “dilate” in order to threshold the delta image and then dilate the thresholded image to fill the holes in the image based on the size of structuring element. Finally the system uses functions “findContours” and “contourArea” to find area and location of each moving object.

5.3. Movement unit
After detecting the areas and locations of moving objects, system may need to move to right, to left, forward or backward to satisfy R1 and R2. Since the system does not have any movement in Y-axis (vertical axis), the unit of movement is based on horizontal angle of view of the camera. As figure 1 and figure 2 show, the scene is basically parted into a 10×10-unit page, while horizontal angle of view of the camera is 53.50 [34]. Hence, the system uses the formula HMU=HAV/HBP, where HMU represents horizontal movement unit, HAV represents the horizontal angle of view, and HBP represents the horizontally basic partitioning of the scene. In result, the movement unit is equal to 5.35 degrees (53.50/10).

6. Result and discussion
Researchers tested the system in indoor. When there is no change in the light of indoor place (all lights are on) and the maximum number of moving objects is 6 – containing human, dog and electrical fan, while the maximum distance between moving object and the system is in the range of 0.5 to 5 meters, the best results are obtained when the resolution is at least set to 640×480, FPS is at least set to 16, minimum number of motion frames is set to 8, and minimum area of motion is set to 1500 pixels.

The system RBMTM will be tested in outdoor places where actual moving objects – such as moving cars and walking humans – exist to find the exact value of resolution, FPS, minimum area of motion and minimum number of motion frames.

References
[1] Pour Yousefian Barfeh D, Balba N, Bustamante R, Jose E, Albacea E and Mariano V 2017 Multi-informational auto tracking algorithm I2BM, Indonesia
[2] Tu Z and Bai X 2010 Auto-context and its application to high-level vision tasks and 3D brain image segmentation IEEE Trans Pattern Anal Mach Intell
[3] Belongie S, Malik J and Puzicha J 2002 Shape matching and object recognition using shape context IEEE Trans. on PAMI 24(4) pp 509–522
[4] Oliva A and Torralba A 2007 The role of context in object recognition Trends in Cognitive Sciences 11(12) pp 520–527
[5] Tu Z, Chen X, Yuille A and Zhu S 2005 Image parsing: unifying segmentation, detection, and object recognition Int’l J. of Com. Vis. 63(2) pp 113–140
[6] Pitas I 2000 Digital Image processing Algorithms and Applications
[7] Wang L, Hu W and Tan T 2003 Recent Developments in Human Motion Analysis Pattern Recognition 36(3)
[8] Lina H, Shafranb I, Yuhc D and Hagenr G 2010 Towards automatic skill evaluation: Detection and segmentation of robot-assisted surgical motions *Taylor&francisOnline*. pp 220-230
[9] Caballeroa A, Cisticloa J, Cantos J and Tomasd R 2010 Optical flow or image subtraction in human detection from infrared camera on mobile robot *Robotics and Autonomous Systems* 58(12) pp 1273–1281
[10] http://www.axis.com/ph/en/products/ptz-cameras (Accessed August 22, 2017)
[11] https://www.bhphotovideo.com/c/product/752624-REG/Samsung_SPU_3750T_1_4_Vertical_Double_Density.html (Accessed August 22, 2017)
[12] http://www.bkgrupe.lt/uploads/docs/Hikvision/DS-C2DF8836IV-AELW.pdf (Accessed August 22, 2017)
[13] Choi W and Savareses E 2012 A unified framework for multi-target tracking and collective activity recognition *12th European Conference on Computer Vision* Florence, Italy
[14] Butt A and Collins R 2013 Multi-target tracking by Lagrangian relaxation to min-cost network flow *CVPR* (2013)
[15] Andriyenko A, Schindler K and Roth S 2012 Discrete-continuous optimization for multi-target tracking *CVPR* (2012)
[16] Qin Z and Shelton C 2012 Improving multi-target tracking via social grouping *CVPR* (2012)
[17] Yang B, Huang C and Nevatia R 2011 Learning affinities and dependencies for multi-target tracking using a CRF mode *CVPR* (2011).
[18] Andriyenko A and Schindler K 2011 Multi-target tracking by continuous energy minimization”. *CVPR* (2011)
[19] Benfold B and Reidstable I 2011 Multi-target tracking in real-time surveillance video *CVPR* (2011)
[20] Sridhar S, Mueller F, Oulasvirta A and Theobalt C 2015 Fast and robust hand tracking using detection-guided optimization *CVPR* (2015)
[21] Melax S, Keselman L and Orsten S 2013 Dynamics based 3d skeletal hand tracking *Canadian Information Processing Society*
[22] Xing J, Ai H and Lao S 2009 Multi-object tracking through occlusions by local tracklets filtering and global tracklets association with detection responses *CVPR* (2009).
[23] Jung B and Sukhatme G 2010 Real-time motion tracking from a mobile robot *International Journal of Social Robotics* 2(1) pp 63–78
[24] Choi W, Pantofaru C and Savareses E 2012 Detecting and tracking people using an RGB-D camera via multiple detector fusion *International Workshops on Computer Vision* Daejeon, Korea
[25] Yang B and Nevatia R 2012 Multi-target tracking by online learning of non-linear motion patterns and robust appearance models *CVPR* (2012)
[26] http://www.axis.com/files/datasheet/ds_215ptz_34462_en_0902_lo.pdf (Accessed Aug 22, 2017).
[27] https://www.bhphotovideo.com/c/product/752624-REG/Samsung_SPU_3750T_1_4_Vertical_Double_Density.html (Accessed Aug 22, 2017).
[28] http://www.bkgrupe.lt/uploads/docs/Hikvision/DS-2DF8836IV-AELW.pdf (Accessed Aug 22, 2017).
[29] Shapiro L and Stockman G 2001 *Computer Vision*
[30] Nixon M and Aguado A 2008 *Feature Extraction and Image Processing*
[31] Ahi K 2016 Modeling of terahertz images based on x-ray images: a novel approach for verification of terahertz images and identification of objects with fine details beyond terahertz resolution *Proc. SPIE* 9456, *Terahertz Physics, Devices, and Systems X: Advanced Applications in Industry and Defense*
[32] Reinhard E 2006 *High dynamic range imaging: Acquisition, Display, and Image-Based Lighting* Morgan Kaufmann
[33] Fisher R, Perkins S, Walker A and Wolfart E 2003 *Spatial Filters - Laplacian of Gaussian*
[34] https://www.raspberrypi.org/documentation/hardware/camera/ (Accessed Aug 22, 2017).