http://dx.doi.org/10.21611/qirt.2008.09_01_06

Real-time mean-shift based tracker for thermal vision systems

G. Bieszczad* T. Sosnowski**

* Military University of Technology, Faculty of Electronics, 2 Kaliskiego Str., 00-908 Warsaw, Poland
** Military University of Technology, Institute of Optoelectronics, 2 Kaliskiego Str., 00-908 Warsaw, Poland

Abstract

In this article, the problems of object tracking with use of digital image processing techniques are discussed. Two methods of object tracking are described and compared. Evaluation of the algorithms will comply with portable thermal camera application.

1. Introduction

Real-time objects tracking is the critical task in many computer vision applications such as surveillance, object-based video compression, or driver assistance. Object tracking is a process of finding a chosen object on following frame using knowledge about its position in previous frames. The most challenging issues encountered in visual object tracking are cluttered background, noise, occlusions and change in appearance of tracked objects. The whole process of object tracking is made with a component called tracker. In typical visual tracker we can distinguish two major components. One is responsible for target representation and localization, while the second is responsible for filtering and data association with respect to object dynamics. This article focuses on the first component responsible for data representation and localisation. Visual tracker has to cooperate with some other components. To start the tracking procedure, the interesting object should be detected. Detection algorithm has to point interesting object out for the tracking algorithm. This component is not always necessary. There are systems where an interesting object is pointed by the system operator. Results of tracking are also used in some reasoning and decision making components. Localization of tracking algorithm in video processing chain is shown in Fig. 1.

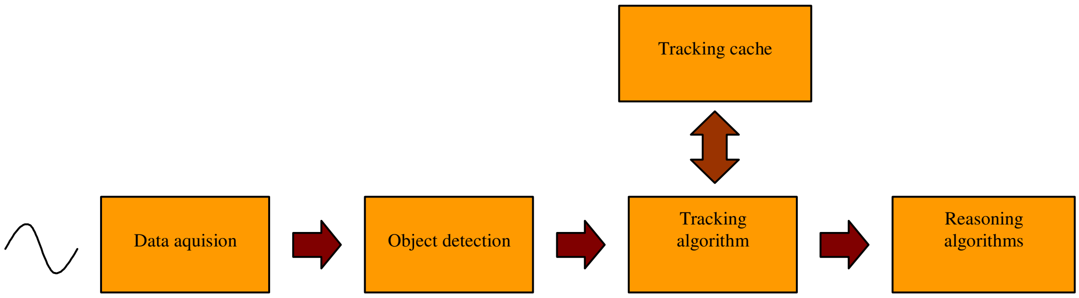


***Fig. 1.** Video processing chain.*

The object tracking uses techniques of digital image processing. Digital Image is built of elementary objects of certain values that represent luminosity. These elementary objects are called pixels. The pixel set is oriented in perpendicular Cartesian coordinate system. An object from our area of interest is represented as a subset of pixels.

In typical visual tracking system, the following images are recorded in fixed periods of time and they are called frames:

$$f = \begin{bmatrix} f(0,0) & f(0,1) & \dots & f(0,N-1) \\ f(1,0) & f(1,1) & \dots & f(1,N-1) \\ . & . & & . \\ f(M-1,0) & f(M-1,1) & \dots & f(M-1,N-1) \end{bmatrix}, \quad (1)$$

where:

f - single frame – the picture,
$f(m,n)$ - value of the pixel in *m*-th row and *n*-th column,
M - picture height,
N - picture width.

2. Methods revision

Tracking algorithms can be divided in four wide categories:

1. Gradient-based methods locate target objects in the subsequent frame by minimizing a cost function [1].

http://dx.doi.org/10.21611/qirt.2008.09_01_06

2. Feature-based approaches use features extracted from image attributes such as intensity, color, edges and contours for tracking target objects [2].
3. Knowledge-based tracking algorithms use “a priori” knowledge of target objects such as shape, object skeleton, skin color models, and silhouette [3].
4. Learning-based approaches use pattern recognition algorithms to learn the target objects in order to search them in an image sequence [4].

Gradient-based methods like SSD (Sum of Squared Differences) evaluate target transition by finding changes between two consequent frames. Changes are estimated with gradients in space and time. This method has relatively low computational complexity but in practice is reliable when small differences between two frames are assured. There is a need to add some special routines to make this algorithm immune to occlusions and lumination changes.

Knowledge-based methods demand excessive knowledge database about objects appearance and dynamics. They have high computational complexity and are used rather in off-line systems. In return, these methods provide high accuracy.

Learning-based approaches unlike previous method class do not use previously collected knowledge base because they learn objects properties on the spot. This method is also combined with a detection algorithm because there is no provided prior information about which object should be tracked. These methods are robust but the usage in real time systems is limited due to high computational power demands.

One of the most robust algorithms used in object tracking systems is color-based kernel density estimation, introduced in [5], which belongs to feature-based class. What is important, feature space is not limited to color data. It can employ any other feature space extracted from image with success. Localization is obtained by comparing and matching areas of image using estimated features. This procedure is usually made using Mean-Shift mode estimation. This approach gained more attention among research community due to its low computational complexity and robustness to appearance change, and eventually evolved too many variations. Adaptation in this method is simpler than in learning-based methods, because it employs only model update.

Choosing a proper method depends on needed application. In offline systems, the most suitable would be ones employing knowledge-based and learning-based methods. In real-time systems, fast and robust methods with low computational demands should be used.

There was a need to find optimal tracking algorithm for portable thermal camera. The architecture of thermal camera is shown in Fig. 2.

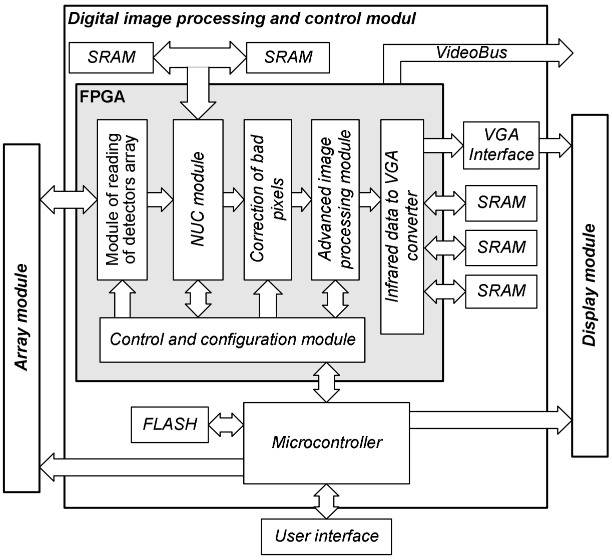

Digital image processing and control modul
SRAM
SRAM
VideoBus
FPGA
Module of reading of detectors array
NUC module
Correction of bad pixels
Advanced image processing module
Infrared data to VGA converter
VGA Interface
SRAM
SRAM
SRAM
Control and configuration module
Array module
Display module
FLASH
Microcontroller
User interface

Fig. 2. *Thermal camera architecture.*

Search for tracking method for this application was executed with respect to presumptions that are consequences of thermal camera architecture. In this application, an algorithm should meet real-time constrains, should be easy to implement in general purpose processor or FPGA structure. That’s why in this application learning-based and knowledge-based methods were not put in consideration. Two methods which present the most promising properties for such application were tested and compared: Mean-Shift and Sum-of-Squared-Differences methods. These methods are well known and successfully used in typical visual systems, but are poorly investigated in systems employing thermal images. In further part of this article, Mean-Shift and Sum-of-Squared-Differences methods will be described in details and comparison between these algorithms will be performed.

http://dx.doi.org/10.21611/qirt.2008.09_01_06

3. Mean shift algorithm

To characterize the object, first a feature space has to be chosen. The reference target model is represented by its Probability Density Function (PDF) in the feature space. For example, the reference model can be chosen to be the luminosity PDF of the object. The reference model is associated to the point **0** = (0,0) in the spatial domain - centre of the reference object. The target candidate localized at the point **y** is represented by its own temperature PDF. Both probability density functions are estimated from m-bin histograms. Histogram is not the best PDF estimation but it is satisfying in this application, and it can be efficiently calculated in hardware like that proposed in [6]. The target and the model have certain size. What is more, in the practice, pixels placed farer form the centre are affected by occlusions, background interference and other noises. To avoid this undesirable effect the kernel *k(r)* is added. In result, probability of feature *u* of the object is now calculated with a formula below:

$$p_u(y) = c_p \sum_{i=1}^{n_p} \left[k\left(\left\| \frac{x_i - y}{h} \right\|^2 \right) \delta\big(b(x_i) - u\big) \right], \tag{2}$$

where:

$p_u(\mathbf{y})$	-	probability of the feature *u* in object centered at the point *y*
$b(\mathbf{x}_i)$	-	function associating the pixel x_i to appropriate feature (here to histogram bin)
k(r)	-	kernel profile
h	-	bandwidth – constant that determinate object size (algorithms area of interest)
δ(x)	-	kronecker delta function
c_p	-	normalization constant

To find a new location of an object in following frame, there is a need to find the most similar target object to the model. Similarity is obtained by the special coefficient, called Bhattacharyya coefficient, which is calculated from Eq. (2).

$$\rho(y) \equiv \rho[p(y), q] = \sum_{u=1}^{m} \sqrt{p_u(y) q_u}, \tag{3}$$

where:

P(y)	-	probability density function of features in object centred at the point *y*
q_u	-	probability density function of features in model object.

To accommodate comparisons among various targets, this distance should have a metric structure. To achieve this, a distance between two probability density functions is calculated using Bhattacharyya coefficient from Eq. (3).

$$d(y) = \sqrt{1 - \rho[p(y), q]}. \tag{4}$$

This statistical measure has some desirable properties like:

it has metric structure,

has geometric interpretation – it is a cosine of angles between two m-dimensional vectors p and q. It uses luminosity as a feature space, therefore its invariant to scale and rotation.

Minimizing Eq. (4) is made by maximizing Eq. (3). Searching of new target starts from a position of the target in the previous frame – $\mathbf{B_n}$, and its neighbourhood. To reduce algorithms computing complexity, the linearization of Bhattacharyya coefficient with Taylor expression around the point y_0 was performed:

$$\rho[p(y), q] \approx \frac{1}{2} \sum_{u=1}^{m} \sqrt{p_u(y_0) q_u} + \frac{1}{2} \sum_{u=1}^{m} p_u(y) \sqrt{\frac{q_u}{p_u(y_0)}} \tag{5}$$

It turned out that this linearization provides great reduction in computational complexity with unnoticeable penalty to algorithms accuracy. Localization is obtained by searching for the coordinates of similarity function maximum. The search of maximum is done with mean-shift procedure [7]. This iterative procedure moves from the location $\mathbf{B_n}$ to the most similar location in area of interest placed at $\mathbf{B_{n+1}}$. Next, the PDF is calculated for the new location $\mathbf{B_{n+1}}$.

$$B_{n+1} = \frac{\sum_{i=1}^{n_p} \left[Y_i w_i g\left(\left\| \frac{Y_i - B_n}{h} \right\|^2 \right) \right]}{\sum_{i=1}^{n_p} \left[w_i g\left(\left\| \frac{Y_i - B_n}{h} \right\|^2 \right) \right]} \tag{6}$$

where:

http://dx.doi.org/10.21611/qirt.2008.09_01_06

$$w_i = \sum_{u=1}^{m} \left[\sqrt{\frac{q_u}{p_u(A)}} \delta\left(b(Y_i) - u\right) \right].$$

$$g(r) = k'(r)$$

The localization is repeated until correlation between the target and the model is low enough. This means that target was found. The more practical ending condition can be adopted. Algorithm can finish tracking when difference between estimation of two following points is smaller then the given threshold. For real-time application, constrain in number of searching iteration is needed too. During tests it turned out that number of iterations is rarely grater than 4, and limit of 10 iterations does not put any noticeable impact on algorithms accuracy.

4. Sum-of-Squared-Differences

Gradient based methods like Sum-of-Squared-Differences localize targets by analyzing differences between consequent frames. Finding target movement is performed by searching minimum of cost function in space and time. Cost function in this approach is a sum of squared differences. Sum of squared differences coefficient is a measure of difference between two fragments of images. Both fragments of images should have equal size. For two square fragments of size 2h+1 by 2h+1 pixels which are centred at the points (x,y) and (u,v) SSD coefficient is equal:

$$SSD = \sum_{i,j} \left[\left(Y_{n-1}(x+i, y+j) - Y_n(u+i, v+j) \right)^2 \right], \quad (7)$$

where:

$$i, j \in [-h, h].$$

If the searched object was detected at the point (x,y) in previous frame, finding its location in following frame would mean finding (u,v) for which SSD coefficient is the smallest. The point (u,v) will be a centre of an object in a new frame found by the algorithm.

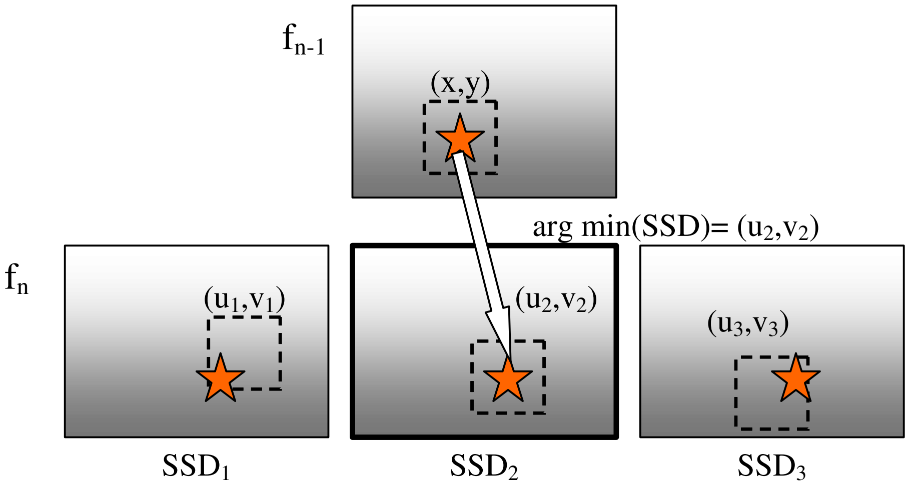


***Fig. 3.** SSD coefficient calculated for various frame fragments. Tracking result is obtained by selecting fragment with the lowest SSD.*

Tracking for all next frames is made in the same way as described above. The fragment containing the found object became a model for further comparison. This approach can lead to some undesired effects. In case where the object tracking will fail for one frame, the algorithm will forget an object. After that, tracking of this object will not be possible. That is why some special routines should be added to make this algorithm immune to partial or full occlusions, noise and changes of appearance like proposed in [7].

5. Test Results

The method was simulated and evaluated with specially recorded sequences using MATLAB software. Sequences were chosen to test algorithms in difficult conditions. They contain the objects that are partially occluded and they appearance change in time. Sequences were containing the frames with resolution of 384x288 pixels. Sequences were registered with a thermal camera.

http://dx.doi.org/10.21611/qirt.2008.09_01_06

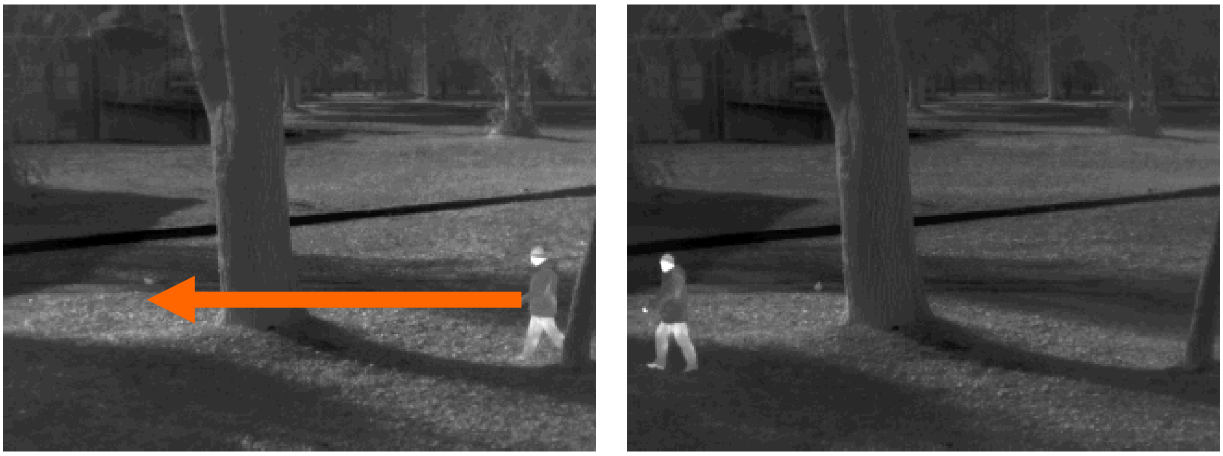
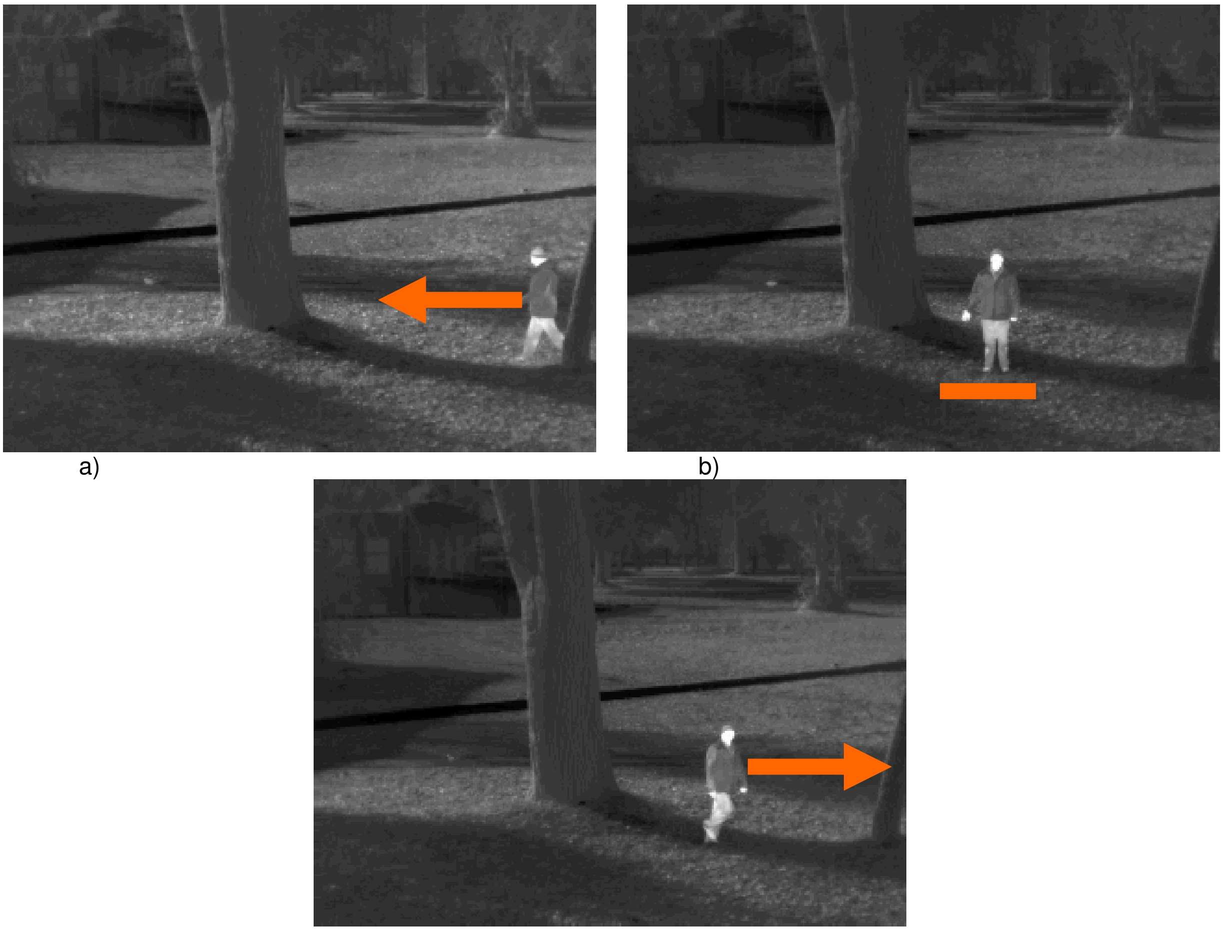


Fig. 4. *First test sequence.*

First test sequence was prepared to check algorithms basic properties. It contains 25 frames and shows a human walking towards left side. This sequence is not affected by serious noise or distractions. A picture of human is large and clear.



a) b) c)

Fig. 5. *Second test sequence.*

Second test sequence contains 72 frames. It shows a human that walks from right side of scene to the centre of scene, stands for several seconds and walks back to the right. In this sequence, appearance of object is changing a lot because it is observed from different angles during its movement.

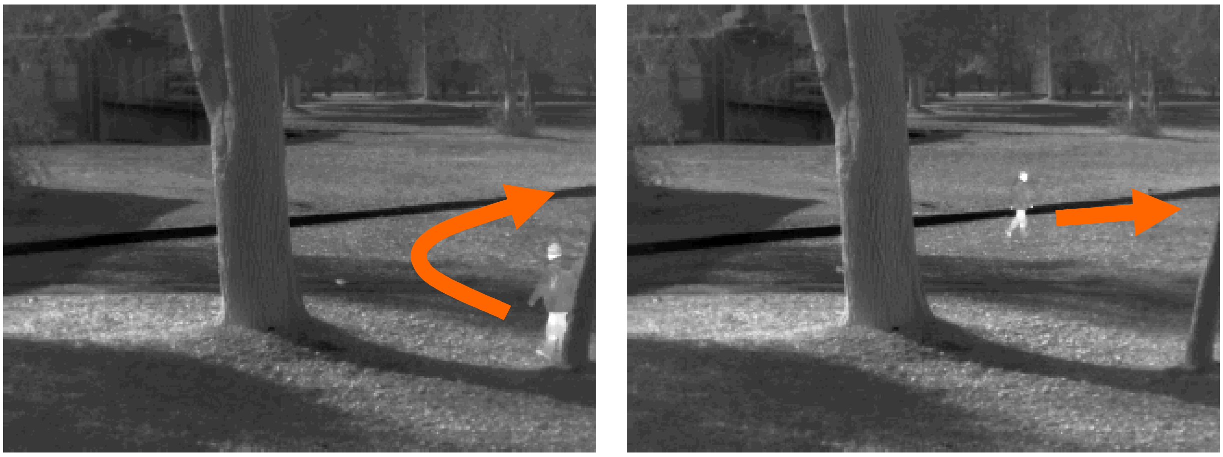


Fig. 6. *Third test sequence.*

Third test sequence contains 80 frames. In this sequence, a tracked human is entering a scene from right and walks by the arc. Finally, he is leaving a scene on the right side. In this sequence, a tracked object is changing appearance drastically. A picture of a human is scaled over time because of a perspective effect. What is more, in this sequence, appearance of the object is getting similar to the background, what can test algorithms in a context of efficient detail extracting and adaptation.

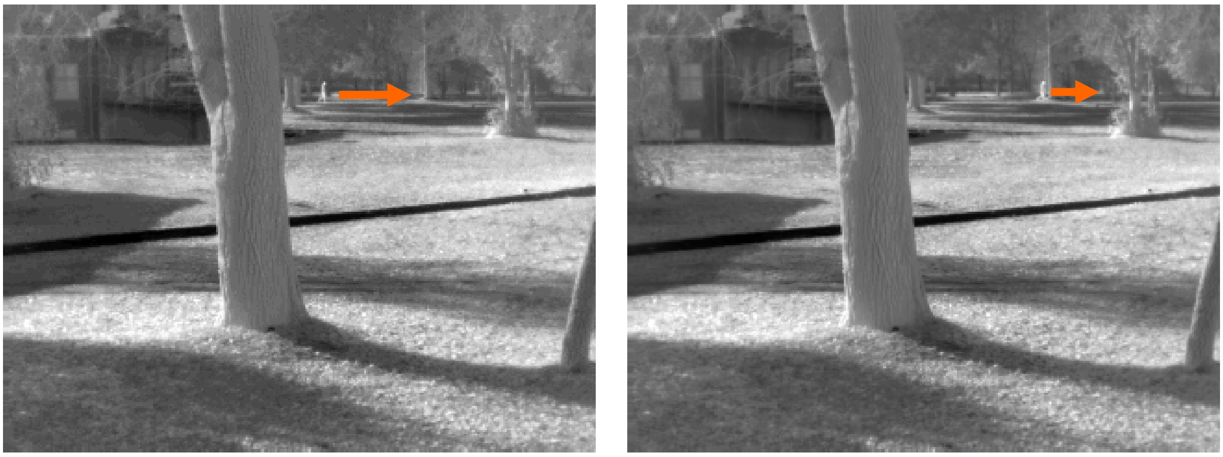
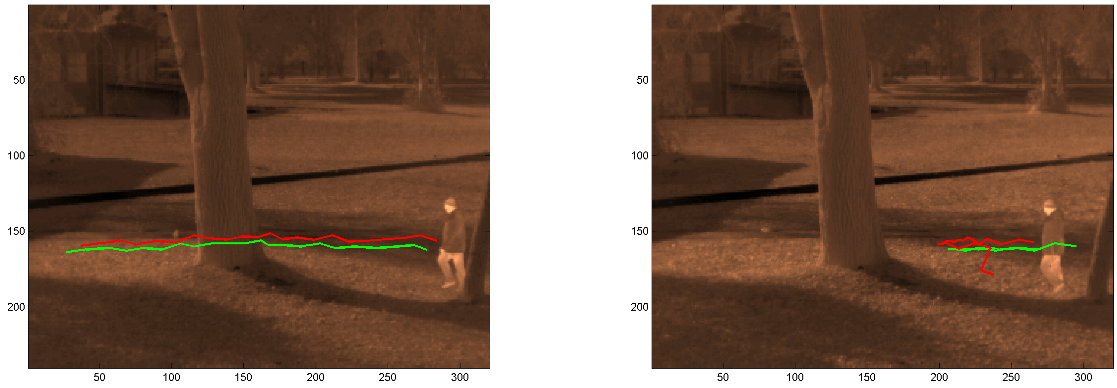


Fig. 7. *Forth test sequence.*

Forth test sequence contains 40 frames. This sequence was recorded to test tracking algorithms efficiency in operation with the objects that contain only several pixels in the picture. This is highly required for a tracking algorithm to handle the objects containing limited amount of pixels, especially for thermal pictures which are of rather low resolution. For small objects, all noises and distractions have much grater impact on objects appearance. Image discretisation effects are also considerable.



http://dx.doi.org/10.21611/qirt.2008.09_01_06

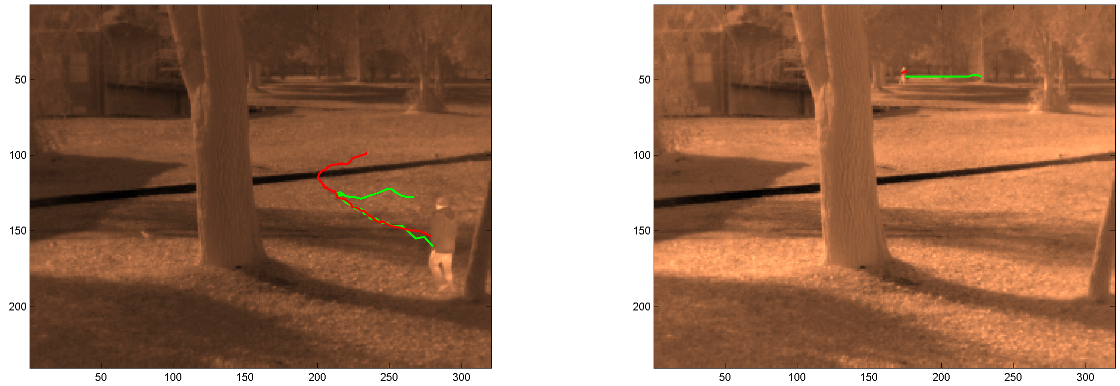


Fig. 8. *Tracking algorithms simulation. Mean-shift tracking results marked with red color. Sum-of-Squared-Differences tracking results marked with green color.*

Simulation showed that both algorithms can track objects, but different properties of these algorithms are noticeable. MS algorithm made better job for the third sequence only. A trajectory was smoother and better fitted to actual one. In the first and second sequence, both algorithms had similar performance. The real demanding forth test sequence showed that SSD algorithm is much better in tracking small objects. MS tracker failed to track the object in this sequence just at the beginning. Problems with tracking small objects with MS method are probably caused by errors in PDF estimation for low amounts of picture data.

General properties of these algorithms can be evaluated on a simulation basis. Properties were set together in a table bellow:

Table 1. *SSD and MS algorithms comparison.*

Feature	Tracking algorithm	
	SSD	MS
Speed	Smaller	Grater
Tracking range	Computational power constrained	Smaller than tracked object
Small objects tracking efficiency	Grater	Smaller
Calculation time	Constant	Variable
Change of appearance immunity	Smaller	Grater
Occlusion immunity	Additional procedures needed	Partial
Rotation invariance	Smaller	Grater
Scale invariance	Smaller	Grater

This simulation showed that Mean-Shift based tracker is not suitable for thermal image systems, especially when it has to track small objects. This evaluation showed that porting algorithms which are successfully used in traditional vision systems to the thermal vision systems is not always successful. The better would be using another algorithm like Sum-of-Squared-Differences in this particular application.

REFERENCES

[1] G. Hager, P. Belhumeur, Efficient region tracking with parametric models of geometry and illumination, IEEE Trans. Pattern Anal. Mach. Intell. 20 (10) (1998) 1025-1039.

[2] M. Isard, A. Blake, Contour tracking by stochastic propagation of conditional density, in: European Conference on Computer Vision, 1996, pp. 343-356.

[3] G. Cheung, S. Baker, T. Kanade, Shape-from silhouette of articulated objects and its use for human body kinematics estimation and motion capture, in: IEEE Conference on Computer Vision and Pattern Recognition, vol.1, 2003, pp. 7784.

[4] O. Williams, A. Blake, R. Cipolla, Sparse Bayesian learning for efficient visual tracking, IEEE Trans. Pattern Anal. Mach. Intell. 27 (8) (2005) 1292-1304.

[5] D. Comaniciu, V. Ramesh, P. Meer, Real-time tracking of non-rigid objects using mean-shift, Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, Hilton Head, SC, 2000.

[6] G. Bieszczad, Hardware implementation of picture histogram estimation in FPGA, SECON 2007 conference, Warsaw, Poland.

[7] R. Venkatesh Babu, Patrick Perez, Patrick Bouthemy, Robust tracking with motion estimation and local Kernel-based color modeling, Image and Vision Computing 25 (2007) 1205/1216

[8] Rafael C. Gonzalez, Richard E. Woods, Digital Image Processing Prentice Hall, ISBN 0201180758

http://dx.doi.org/10.21611/qirt.2008.09_01_06