Indonesian Journal of Electrical Engineering and Computer Science
Vol. 11, No. 2, August 2018, pp. 446~452
ISSN: 2502-4752, DOI: 10.11591/ijeecs.v11.i2.pp446-452

Implementation of Robust Tracking Algorithm on Nano-Computer

Khaled HAMMEMI[1], Mohamed ATRI[2]
[1]Departement of Electrical, School of Engineers, University of Monastir, Tunisia
[2]Laboratory of Electronics and Microelectronics, Faculty of Sciences, University of Monastir, Tunisia

Article Info

Article history:

Received Jan 5, 2018
Revised Apr 2, 2018
Accepted Apr 21, 2018

Keywords:

Robust tracking
Partial occlusion
Failure recovery
Raspberry pi

ABSTRACT

In this work, we developed the NSSD-DT method, which allows us to track a target in a robust way. This method effectively overcomes the problems of geometrical deformation of the target, partial occlusion and allows recovery after the target leaves the field of view. The originality of our algorithm is based on a new model, which does not depend on a probabilistic process and does not require data-based beforehand. Experimental results on several difficult video sequences have proven performance benefits. The algorithm is implemented on a BCS 2835 system based on a quad core ARM processor, it is also compared to the software solution. NSSD-DT can be used in several applications such as video surveillance, active vision or industrial visual servoing.

Corresponding Author:

Khaled HAMMEMI,
Departement of Electrical, School of Engineers,
University of Monastir, Tunisia.
Email: kh.hammami@gmail.com

1. INTRODUCTION

Object detection is based on image processing, many methods have been modeled and developed and implemented on many systems. Nano-computer development opens up great opportunities for embedded systems that apply in several research areas [1]. The ARM-based embedded system is a generation of nano-computers with limited memory capacity, but effective for an implementation of non-resource intensive processing algorithm.

Raspberry Pi is implemented to support object detection algorithms and monitoring in many applications such as surveillance, car navigation and autonomous robot navigation [2], to maintain continuous monitoring of the underlying transmission lines marine [3]. Tracking objects with partial or complete occlusion using features such as colors and contours based on the HSV color model by [4]. RGA and SKDA are two different methods that can be used to detect the object in the moving surveillance camera system, both methods have been tested to be examined, which is more reliable to implement in a computer to a only table. A discussion of the implementation and testing of two different methods of object detection using subtraction of backgrounds. They implement two methods combined with Extended Kalman Filter in a Raspberry Pi, by [5]. An application of precision farming in the detection process to control weeds by computer vision. The system can be used for the fractal size treatment of weeds, the average speed ratio between the PC and Raspberry Pi is 0.04 times faster. The authors deduce that the use of Raspberry Pi is cheaper and the energy consumption is efficient compared to a personal computer, by [6]. SIFT algorithm for palmar vein recognition is proposed is built on Raspberry Pi. The image is cropped into Region of Interest (ROI) to reduce computational time in real-time systems and then preprocessed to improve the visibility of the system, image to extract by [7].

2. RESEARCH METHOD

The matching model is adopted to detect a small part that corresponds to a template image. This technique is widely used in object detection fields such as vehicle tracking, robotics, medical imaging and in the industry as part of quality control.

The crucial point is to adopt an appropriate "measure" to quantify similarity or matching. However, this method also requires a high computational cost since the matching process involves moving the model image to all possible positions in a larger source image and calculating a numerical index indicating how much the pattern corresponds to the image in that position. This problem is therefore considered as an optimization problem.

The measurement of the correspondence between two images is considered as a metric that indicates the degree of resemblance or dissimilarity between them. This metric may be increasing or decreasing with a degree of similarity. When the metric is specifically indicated as a measure of inadequacy, it is an amount that increases with the degree of dissimilarity.

By sliding, we move the patch one pixel at a time (left to right, up to down). At each location, a metric is calculated, it represents “good” or “bad” match at that location. For each location of Template over source image, we store the metric in the result matrix (R). Each location (xy) in R contains the match metric. The image in Figure 1 is the result R of sliding the patch with a metric NSSD (normalized sum squared difference) [A1] or Normalized cross correlation method [A2]. The brightest locations indicate the highest matches. The location marked by the red circle is the one with the highest value. Thus, that location (the rectangle formed by that point as a corner and width and height equal to the patch image) is considered the match.

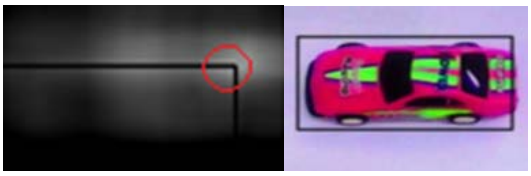


Figure 1. Result matrix (R)

$$s(x,y) = \frac{\sum_{y=0}^{k-1}\sum_{x'_{-0}}^{w-1}[T(x',y) - I(x+x',y+y')]x^2]^2}{\sqrt{\sum_{y'=0}^{k-1}\sum_{x'=0}^{w-1}T(x',y')^2 \ \sum_{y'=0}^{k-1}\sum_{x'=0}^{w-1}I(x+x',y+y')^2}} \qquad [A1]$$

$$R(x,y) = \frac{\sum_{x',y'}(T(x',y').I(x+x',y+y'))}{\sqrt{\sum_{x',y'} 2T(x',y')^2 \cdot \sum_{x',y'} I(x+x',y+y')^2}} \qquad [A2]$$

2.1. Comparaison de deux methods NCC and NSSD

In practice, the algorithm is applied to a series of images, which are parts of the reference image. The calculation is performed between the reference image and a distorted image. The distorted image that gives the greatest correspondence with the reference image is chosen as the best and thus makes it possible to estimate the displacement at this stage (Figure 2).

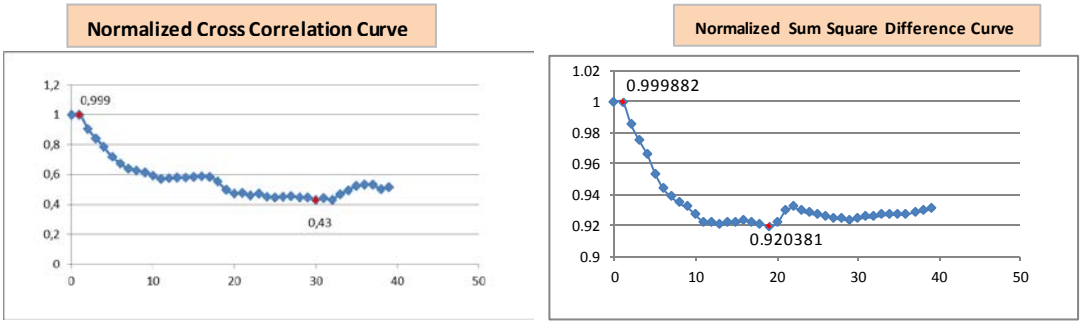


Figure 2. Correspondence values for normalized cross correlation and Normalized Sum Square Difference method

For values between 0.999 and 0.930, a 44% match is obtained by the NCC method. For values between 0.999 and 0.430, a better match of 98% is obtained by the NSSD method.

2.2. The NSSD_DT Algorithm

The proposed Matching model NSSD_DT is based on the updating of the template according to a sentinel of recognition. The update is triggered with each change of the source in its geometric form, its scale, Rotation, or occlusion. The tracking begins with an original template (Figure 3). At the first change of the source that exceeds the sentinel index, the updating is done by substitution of the old template by the new one.

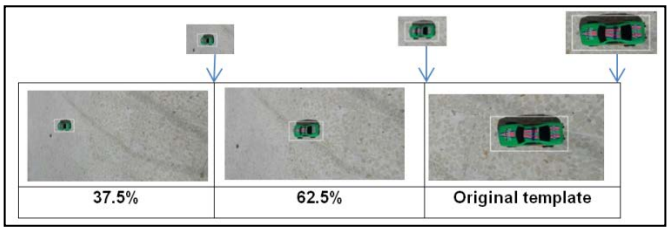


Figure 3. Sentinel principle

The Normalized sum squared difference algorithm with dynamic template

1: Load an input image (source)
2: Load an image Template
3: Matching
4: If the matching factor <threshold
{
Updated Sizes Template
Restore a new Template
}
5 : Matching
6 : Locate the location with a higher likelihood of adaptation
7: Locate by a rectangle the area of highest correspondence

Matching
Perform a Matching Procedure Model
Template Reconstruction Procedure
Home
{Calls parameter Template (j-1)
Extract Template (j)
Swap Template (j-1) / Template (j)
} End

2.3. Implementation on SOC

With a relatively simpler architecture than other processor families, and with low power consumption, ARM processors have become dominant in the field of embedded computing, especially mobile telephony and tablets. ARM is best known for its SOC, which integrates on a single chip: microprocessor, graphics processor (GPU), DSP, FPU, SIMD, and device controller. Nano-computers with ARM11 processors and a minimum size, can be qualified as a computers, as it allows to run a high-level operating system, It integrates display and input devices. It includes a dedicated storage means (hard disk, SD card, USB key), and operates autonomously.

2.4. Implementation language of the NSSD-DT algorithm

The implementation is done by Python language, it is a powerful, interpreted-compiled language, improved progressively for exponential, multiplatform execution speeds, and the old code works on new versions of python. The Python language is advantageous by using high-level data types and using bytecode as an intermediate representation before the transformation into machine code into the ARM architecture. Automatic compilation into byte code, reduces the dependence on the hardware and facilitates its execution on several architectures.

2.5. Hardware System

The BCM2835 SOC contains an ARM ARM1176JZF-S core processor (ARM11) clocked at 700 MHz, a Video- Core IV GPU for video processing and capable of processing the 1080p30 h.264 / MPEG-4, as shown in Figure 4 (encoder and decoder), 256 MB RAM or 512MB of RAM, a cache, and ports (GPIO, timers, I2C - SPI, interrupt controller - PIC, memory controller - MMU, UART ...).

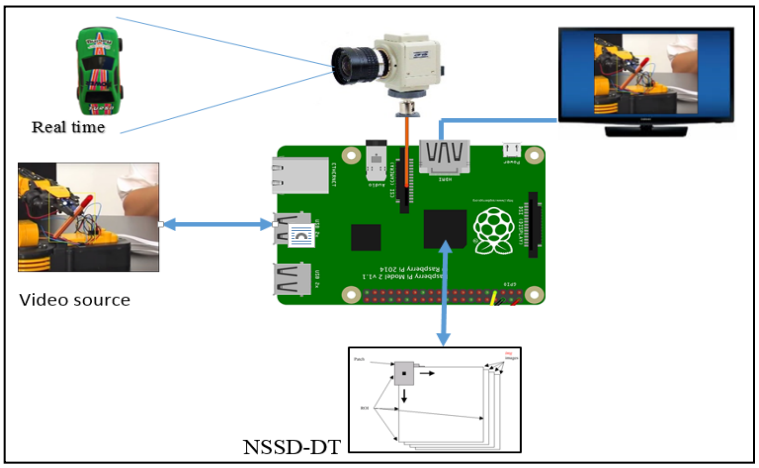


Figure 4. Hardware System

2.5.1. Evaluation of the NSSD_DT algorithm

In this section, to evaluate our approach, we apply our NSSD_DT algorithm implemented in C ++ (on soft) and python(on chip) on 8 different videos at 20 Fps: different resolution (v1), source rotation and scaling (v2, v3, v4) and we resume tracking the race car after leaving the field of view (v5).

2.5.2. Tracking with fixed template and different resolutions of the source.

With a fixed resolution template 1080x1920, we track the targets at different resolution, as shown in Figura 5. La Table 1 summarizes the results that represents the average of 10 values raised for each resolution. The recognition results are evaluated at an average of 92%.

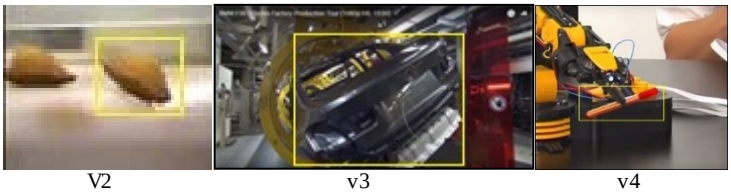
Figure 5. with different resolution

Table 1. Fixed template and different resolutions of the source

resolution	Average recognition rate	Average execution time Soft solution	Average execution time Embedded solution
1080 x 1920 HD	0.8754	0.321	0.420
720 x 1080 HD	0.8354	0.295	0.361
480 x 720	0.8452	0.261	0.343
320 x 480	0.8356	0.257	0.332
240 x 320	0.8147	0.243	0.355
240 x 144	0.7984	0.222	0.310
Average	**92.03%**	**0.266**	**0.353**

2.5.3. Test with rotation of the source

The test is applied to three industrial videos; v2 to follow the madeleine, v3 to follow the hull of the car on the chain and v4 to follow the red pen held by the arm of the robot. Fidure 6 and Table 2 summarizes the measures and gives a means of recognition of 87%.



V2 v3 v4

Figure 6. With rotation of the source

Table 2. Pen Tracking Table with Rotation and Scale Change

video	Fps	Average recognition rate	Average execution time (s) Soft solution	Average execution time (s) Embedded solution
V2	20	0.7584	0.262	0.365
V3	20	0.8455	0.236	0.384
V4	20	0.7562	0.256	0.395
	Average	**87%**	**0.251**	**0.381**

2.5.4. Partial Occlusion and Field of View Output Test

The evaluation of our algorithm is replicated 5 times on this sequence. The Figure 7 and Table 3 summarizes the average of recognition and the average of the execution time.

Figure 7. Partial Occlusion and Field of View Output Test

Table 3. The Average Result for occlusions

Fram	Average recognition rate	Oclusions	Average execution time Soft solution	Average execution time Embedded solution
10	0.8523		0.256	0.366
75	0.7912		0.262	0.354
93	0.7455		0.236	0.395
120	0.7562		0.256	0.354
144	0.6214	Partiel oclusion	0.265	0.368
176	0.2356	Total oclusion	0.254	0.374
198	0.5784	Partiel oclusion	0.256	0.385
215	0.7587		0.248	0.376
Average	**74%**		**0.254**	**0.371**

3. RESULTS AND ANALYSIS

Based on the various robustness tests of our tracker, we obtain very relevant results with an average overall recognition rate of 84%, as shown in Table 4.

Table 4. Recap of Results

Recognition tests	Recognition rate
Tracking with different resolutions	92%
Tracking with source rotation	87%
Tracking after quitting the field of view	74%
General Recognition	**84 %**

In this section we analyses the calibration of the NSSD-DT method, a comparison between the software solution by the C ++ IDE and the embedded solution as well as the performance of the adopted SOC system as well.

The tracking principle is based on a vigilant control of the similarity index. If it reaches the predefined nominal value (ie the sentinel value) then an immediate update of the model is triggered to continue the tracking with a similarity closer to the ideal. This assumes that the value of the sentinel index must be very well chosen, since a value close to 1 affects the strength of the tracking, and a value close to 0.2 may affect the accuracy of the marking of the target. A series of tests for a given subject allows choosing this famous index. Generally, a value close to 0.5 gives satisfactory results.

Table 5 and the curve below summarizes a comparison of the software solution and the embedded solution on sequences varying in level of occultation as shown in Figure 8, the difference in response time is evaluated on average at 175 ms.

Table 5. Tracking with occlusion

occlusion	100%	20%	28%	48%	60%	64%	75%
correspondence	0.90125	0.7226	0.6482	0.4702	0.3625	0.3245	0.2253
Soft time	0,233	0,256	0,245	0,265	0,236	0,265	0,256
Embeded time	0,39	0,412	0,455	0,43	0,413	0,422	0,462
différence	0,157	0,156	0,210	0,165	0,177	0,157	0,206

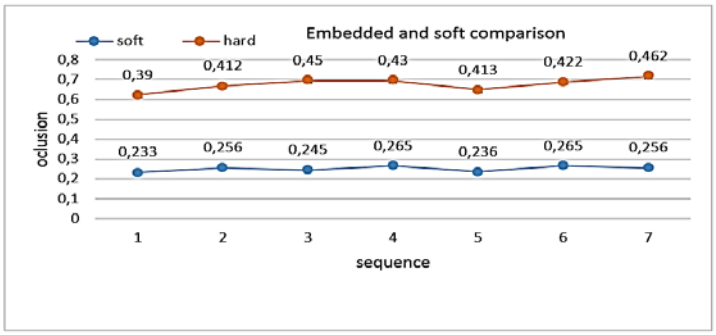


Figure 8. Tracking with occlusion - comparison soft and embedded solution

In this work we have embedded our algorithm on a system-on-chip with important performance, but the responses, while satisfactory, can be better with a faster system on chip and higher memory. For the design of industrial control systems, embedded solutions will now have to outperform software solutions and go beyond the characteristics of the usual nano-computers [8], by the easy processing of videos in full HD.

4. CONCLUSION

In this work, we have dealt with the problem of tracking moving objects. We develop a robust, long-term tracking algorithm with the performance of detecting failures to follow up and recover after a failure, by solving deformation and occlusion problems. The NSSD-DT algorithm is implemented on a BCM 2835 SOC system based on a quad-core processor, ARM 1176 JZF-S CORE. The execution time by the embedded processor-based solution Arm is relatively higher compared to the soft implementation (1.68%) under software environment, with the processor quad-cœur 2 GHz. But the benefits of Arm are multiple in terms of power consumption (3.5W (5V and 0.75A)), size and cost.

REFERENCE

[1] Nugroho, Dita et Lonsdale, Michele. Evaluation of OLPC programs global: a literature review. 2010

[2] JANA, Sampa et Borkar, Shubhangi. Autonomous Object Detection and Tracking using Raspberry Pi. *International Journal of Engineering Science*, 2017, vol. 14145

[3] AMIR, Samreen, Siddiqui, Ali Akbar, Ahmed, Nimrah, et al. *Implementation of line tracking algorithm using Raspberry pi in marine environment.* In: Industrial Engineering and Engineering Management (IEEM), 2014 IEEE International Conference on. IEEE, 2014. p. 1337-1341.

[4] Gajbhiye, Samar D. et Gundewar, Pooja P. *A real-time color-based object tracking and occlusion handling using ARM cortex-A7*. In : India Conference (INDICON), 2015 Annual IEEE. IEEE, 2015. p. 1-6

[5] Jati, Agung Nugroho, Novamizanti, Ledya, Prasetyo, Mirsa Bayu, et al. Evaluation of Moving Object Detection Methods based on General Purpose Single Board Computer. *Indonesian Journal of Electrical Engineering and Computer Science*, 2015, vol. 14, no 1, p. 123-129.

[6] Suriansyah, Mohamad Iqbal, Sukoco, Heru, et Solahudin, Mohamad. Weed Detection Using Fractal-Based Low Cost Commodity Hardware Raspberry Pi. *Indonesian Journal of Electrical Engineering and Computer Science*, 2016, vol. 2, no 2, p. 426-43.

[7] Kumar, Ranjith, Deepika, G. G., Krishnan, Meenakshi, et al. An Open Source Contact-Free Palm Vein Recognition System. *International Journal of Advances in Applied Sciences*, 2017, vol. 6, no 3, p. 328-332 IJAAS

[8] ALI, Murat, VLASKAMP, Jozef Hubertus Alfonsus, EDDIN, Nof Nasser, et al. *Technical development and socioeconomic implications of the Raspberry Pi as a learning tool in developing countries*. In: Computer Science and Electronic Engineering Conference (CEEC), 2013 5th. IEEE, 2013. p. 103-108.