2014 International Work Conference on Bio-inspired Intelligence (IWOBI)

Automatic Tracking and Counting of Moving Objects

Jonathan Johannes Philipps, Ingrid Bönninger, Martin Weigert
Brandenburg University of Technology
Cottbus-Senftenberg, Germany

Javier Vásquez
Universidad de Costa Rica
San José, Costa Rica

***Abstract*—This work presents the conception and development of a software system to detect, track and count automatically moving objects in videos. The aim of the system is the automatically counting of moving turtles on a beach. To achieve a high recognition rate of the moving objects we consider three segmentation strategies (SubstractionGrayscale, SubstractionBinarization, Substraction Canny), two object identification methods (Grayscale Connected, FelzenwalbHuttenlocher) and two object recognition methods (Nearest Object Distance, Certain Recognition Matching. In the preprocessing step of our system we select parameters like the interesting area and the object size. We use the segmentation process for the recognition of potential objects. After the identification of objects as turtles we track their movements. In the last step we count the turtles. In order to have a representative test number of moving objects we use videos with 1661 moving cars. The best results of 98.98 percent we reached with the combination Canny Edge Detection, Grayscale Connected, and Certain Region Matching Strategy.**

Keywords—automatic tracking; moving objects; image processing; video; background separation; object recognition; labeling

I. INTRODUCTION

With our system we want to count automatically the number of mother turtles that move to the beach for egg deposition.

To track moving objects there are different approaches. Adaptive Background Subtraction and images differences are e.g. used in [1],[2], and [3]. Reference [4] uses edge detection for the parameter extraction and classification of pollen types, and [5].

K-means clustering algorithm and histogram analysis, Weighted average method, Median filter, and Gaussian Mixture Model are employed in [6] to separate the objects from the background. Reference [7] identifies moving objects by determining the center of mass, [8] proposes a dual-layer particle filter for detecting and tracking multiple moving objects.

Our system has focused on a real-time system to track turtles that move into a defined direction - to or from the sea. To reach good identification results and a good system performance we implemented tree segmentation strategies, two object identification methods and two object recognition methods. We used all possible combinations of these methods to find out the best strategy for the object recognition and counting.

Because of the absence of a representative test number of turtles we evaluated our system with moving cars.

Our article is divided into four sections.

We begin with the description of our concept and the implemented methods. Section III deals with our experiments, and in section IV the results are discussed, and we recommend a combination of methods to track and count moving objects.

II. CONCEPT OF THE SYSTEM

We propose a model of six steps:

- Preprocessing
- Background separation and segmentation
- Object identification
- Labeling of objects
- Recognition of objects
- Tracking of objects

These six steps are explained in the following sections.

A. Preprocessing

We use the preprocessing to split the video into images and for the setting of user defined parameters. The preprocessing of video data is shown in fig.1.

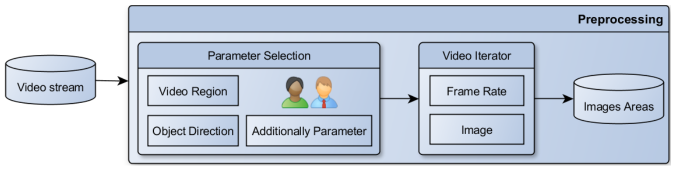

Fig. 1. Preprocessing

The process starts with the image extraction from videos. The user can select an interesting area, the frame rate, and define the moving direction of the turtles. We count the turtles only in one moving direction (from or to the sea) in order to avoid double counting of the same turtle. With the frame rate the user can determine in which detail and speed the video will be analyzed.

978-1-4799-6173-0/14/$31.00 ©2014 IEEE

Now we have separate images for the further image processing. Our next step is the segmentation.

B. Segmentation

In the segmentation step we separate the background and extract the segments of moving objects. We use the combination of three segmentation strategies to filter the moving objects from two sequential images (see fig. 2).

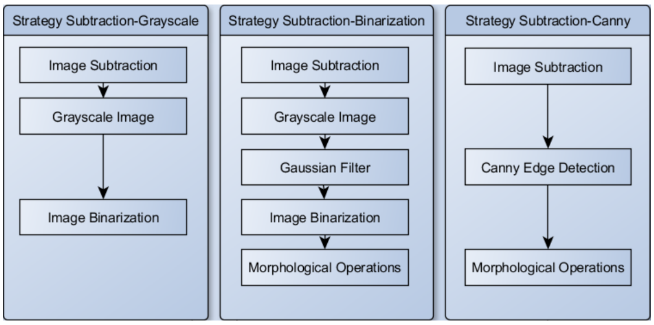

Fig. 2. Segmentation strategies

Fig. 3 shows the different results of the three segmentation strategies, applied to the original (fig. 3(a)). Fig. 3(b) shows the manipulation by the SubstractionGrayscale, 3(c) by the SubstractionBinarization, and 3(d) by SubstractionCanny without morphological operations.

(a) Original Image | (b) SubstractionGrayscale | (c) SubstractionBinarization | (d) Substraction Canny

Fig. 3. Segmentation results

The Substraction Binarization uses a filter to eliminate noises, e.g. vibration of the camera. An edge detection filter is used by the Substraction Canny strategy. To the two last strategies we added morphological operations.

At the end of the segmentation process we have connected segments in the foreground of the images. Now we have to identify object from the segments.

C. Identification

In this section we want to discuss the used strategies to get objects from segments. The identification process marks the object in a segment with labels. With the Felzenszwalb-Huttenlocher Algorithm [9] we identify objects by connecting pixels with similar properties.

The algorithm connects all pixels with n edges e. The edges get weights w of similarity of the pixels v_j and v_j from their Euclidean distance of RGB-values features r (red), g (green), and b (blue).

$$w(v_i, v_j) = \sqrt{(r_i - r_j)^2 + (g_i - g_j)^2 + (b_i - b_j)^2} \quad (1)$$

The identification of objects starts with a segmentation S^0, where each pixel v_i is in its own component C_i^0. The algorithm merges into q steps (q = 1,..,n) two components C_i^{q-1} and C_j^{q-1}, if the minimal difference between both components

$$Dif(C_i, C_j) = \min_{v_i \epsilon C_i,\ v_j \epsilon C_j} w\,(v_i, v_j) \quad (2)$$

is small compared with the internal differences $Int(C)$

$$Int(C) = \max w\,(e) \quad (3)$$

of each component C_i^{q-1} and C_j^{q-1}. We define a threshold function for our merging decision M to merge C_i^{q-1} and C_j^{q-1}

$$M(C_i, C_j) = \begin{cases} true\ if\ Dif(C_i, C_j) \le MInt(C_i, C_j) \\ false \end{cases} \quad (4)$$

with $MInt(C_i, C_j)$ as the minimum of the internal differences

$$MInt(C_i, C_j) = \min\left(Int(C_i) + r(C_i)\,, Int(C_j) + r(C_j)\right) \quad (5)$$

The Grayscale Components Labelling Strategy uses the Grayscale Component Segmenter, that marks connected pixels with the same label. The labeled objects get additional information of size and position.

To reduce noise and wrong labels we use label Constraints Checking. We use the maximum and minimum label height for the verification. Labeled objects with intersections are eliminated, because it is not possible, that we have two or more objects at the same position.

Now we can identify objects in the image. To count objects in a video we have to track the objects from image to image.

D. Recognition and Tracking

In the recognition process we compare new objects with known objects (see fig. 4). So a tracking of the moving objects is possible.

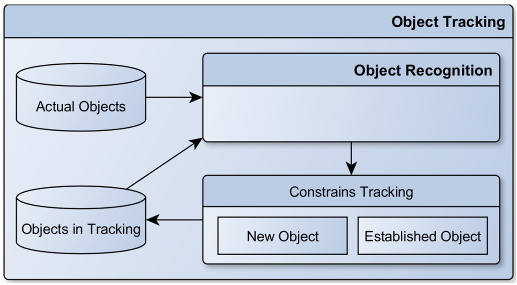

Fig. 4. Recognition and Tracking

For the recognition problem we use two strategies, the Nearest Object Distance and the Certain Region Matching. The Certain Region Matching strategy is a combination of Nearest Object Distance and Feature Matching strategy.

The Basic Matcher tests the distance between the feature using thresholds. The result is the comparison value in percent that describes the equality of two objects.

Recognized objects get a colored border and a unique number to track them. In fig. 5(c) the right car is changing its border color. That means, that it is recognized as another car than in fig. 5(a) and 5(b), the tracking of this car failed because of noises.

(a) (b) (c)

Fig. 5. Object Recognition

Now if we can track objects we want to count them considering constraints.

E. Object Counting

For each object that is detected we check now the counting conditions. The algorithm tests the movement direction and the expected velocity (see fig. 6).

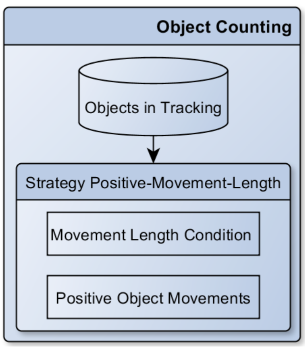

Fig. 6. Object Counting

III. Experimental Settings

In order to have a representative test number of moving object we have used videos with 1661 moving cars, motorbikes and other moving objects at highways. We made one video in the rain and three in the sunshine on four different locations. We used a video resolution of 800 to 600 pixels. Problems were vibration, rain, near object capturing and object deformation because of different distance.

We recorded the first video of 15 minutes in front of a roundabout (see fig. 7, video1) with 352 moving objects. This video was biased by

Video 1 Video 2 Video 3 Video 4

Fig. 7. Recoded Videos

rain and vibration caused by trucks and other large vehicles.

Test video 2 contains 204 moving objects and has 10 minutes process time. The third video runs 15 minutes and contains 445 moving objects. The last video contains 660 objects and has a length of 15 minutes (see table I).

TABLE I. Test Videos

Video No	Time [min]	No of objects
1	15	352
2	10	204
3	15	445
4	15	660

As test systems we used the three test systems Notebook Lenovo (NBL) Lenovo Thinkpad T61 Core 2 Duo 2 x 1,80 Ghz 2048 MB, Notebook Dell (NDell) Dell Studio 1749 Intel Core i3 4096 MB, and Desktop (DK) Desktop system AMD Phenom Quad-Core 4096 MB.

So we get 144 (4 videos * 12 strategies * 3 test systems) test scenarios.

To evaluate the best strategy we combined each strategy of every step and test the accurateness of the counting result.

We have tested 12 strategy combinations (see fig. 8).

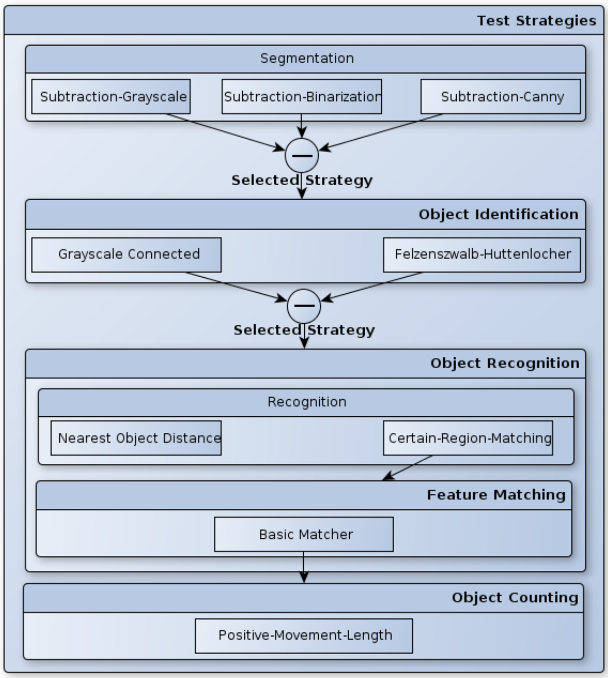

Fig. 8. Test Strategies

IV. RESULTS

All combinations provided counting results better than 72 percent [10].

The best results of 98.98 percent we reached by the Canny Connected Matching Matcher Strategy (see table II).

TABLE II. RESULTS

Strategy	**No Recognized Objects**	**No Missed Objects**
Grayscale Connected Distance	1414	247
Grayscale Connected Matching Matcher	1421	240
GrayscaleHuttenlocher Distance	1581	80
GrayscaleHuttenlocher Region-Matching Matcher	1554	107
Binarization Connected Distance	1241	420
Binarization Connected Matching Matcher	1242	419
BinarizationHuttenlocher Distance	1556	105
BinarizationHuttenlocher Region-Matching Matcher	1562	99
Canny Connected Distance	1642	19
Canny Connected Matching Matcher	1644	17
Canny Huttenlocher Distance	1608	53
Canny Huttenlocher Region-Matching Matcher	1577	84

The results of the performance tests in seconds are shown in table III with NBL for Lenovo, NDell for Dell notebook and DK for Desktop system.

TABLE III. PERFORMANCE TESTS

Strategy	Time NBL	Time NDell	Time DK
Grayscale Connected Distance	273	206	187
Grayscale Connected Matching Matcher	282	204	191
GrayscaleHuttenlocher Distance	281	204	197
GrayscaleHuttenlocher Region-Matching Matcher	288	204	196
Binarization Connected Distance	256	206	197
Binarization Connected Matching Matcher	269	208	202
BinarizationHuttenlocher Distance	269	209	206
BinarizationHuttenlocher Region-Matching Matcher	278	206	205
Canny Connected Distance	246	182	175
Canny Connected Matching Matcher	247	184	178
Canny Huttenlocher Distance	283	187	183
Canny Huttenlocher Region-Matching Matcher	277	186	185

REFERENCES

[1] Q. M. Zhou & J. K. Aggarwal,"Trackingand Classifying Moving Objects from Video", In PETS 2001 Benchmark Data, 2001.

[2] Carlos M. Travieso, Juan C. Briceño, Jaime R. Ticay-Rivas & Jesús B. Alonso, "Pollen classification based on contour features", 15th International Conference onIntelligent Engineering Systems,2011.

[3] O. Javed & M. Shah, "Tracking and ObjectClassification for Automated Surveillance", In CCV, pp IV: 343 ff., 2002.

[4] Q. Zang & R. Klette, "Object Classificatio and Tracking in Video Surveillance". In CAIP, pp. 198–205, 2003.

[5] Y. Zhang, Linkai Luo, "Background extraction algorithm based on K-menas clustering algorithm and histogram analysis", IEEE International Conference in Computer Science and Automation Engineering (CSAE), pp. 66-69, 2012.

[6] Z. Xu, Q. Qiu, L. Wang, "The Solution to Background Segmentation and Extraction in Traffic Video Detection", 11th International Symposium on Distributed Computing and Application to Business, Engineering & Science (DCABES), pp 392-395, 2012.

[7] T.Y. Lai, J. Y. Kuo, C. Liu, Y. W. Wu, Y. FanJiang, S. Ma, "Intelligent Detection of Missing and Unattended Objects", in Complex Scene of Surveillance Videos, International Symposium on Computer, Consumer and Control (IS3C), pp. 662-665, 2012.

[8] K. Jung, N. Kim, S. Lee, J. Paik, "Dual-Layer particle filtering for simultaneous multiple objects detection and tracking", IEEE International Conference on Acoustic, Speech and Signal Processin (ICASSP), pp. 2307-2311, 2013.

[9] P. F. Felzenswalb, D. P. Huttenlocher, „Efficient Graph-Based Image Segmentation“, International Journal of Computer Vision, pp. 167-181, 2004.

[10] J. Philipps, Automatic detection and tracking of moving objects, Bachelor Thesis, Hochschule Lausitz (FH) University of Applied Sciences, 2012.