Detection and Identification of Animals Using Stereo Vision

Olivier Lévesque and Robert Bergevin
olivier.levesque.1@ulaval.ca robert.bergevin@gel.ulaval.ca
Computer Vision and Systems Laboratory of Laval University
Québec City, Canada G1K 7P4

Abstract

Automating the detection and identification of animals is a task that has an interest in many biological research fields and in the development of electronic security systems. We present a system based on stereo vision to achieve this task. A number of criteria are being used to identify the animals, but the emphasis is on reconstructing the shape of the animal in 3D and comparing it with a knowledge base. Using infrared cameras to detect animals is also investigated. The presented system is a work in progress.

1. Introduction

Biologists from our community study the impact of human construction on fauna by producing statistics on the presence of animals in the surrounding region before and after these constructions. This paper introduces a system based on stereo vision to observe a fixed scene where animals may be present. The system under development is to search for identification criteria on all the moving objects observed, for instance, motion speed, size, 3D shape, color and texture to compare each observed object with an animal description in a database and achieve classification. Observations span nearly three weeks, obtained during winter and summer of 2009 in a natural outdoor environment where different species such as birds, squirrels, dogs and humans are visible. The objective of this paper is to explain the approach we use to achieve the identification and this project's current progress status. Existing techniques are integrated in order to solve a number of standard computer vision problems such as segmentation of objects, tracking, camera calibration and 3D reconstruction. The main contribution of this project is to overcome challenges arising from the variety of animal species and weather conditions. The intended output of the project is an efficient software usable by a novice in the computer vision field. In the next section, the main algorithm is introduced. Each step is then explained in detail in the following subsections. Technical information is given at the end of the paper.

2. Main algorithm

The main algorithm (see Figure 1) comprises eight steps: initialization, image acquisition, background model update, background segmentation, labeling and grouping, modelling and feature extraction, comparison and identification, global statistics computation.

2.1. Initialization

The initialization task is divided into two parts, computation of the calibration matrix and compution of the background model. A calibration matrix M needs to be computed for each camera to map its coordinates to world coordinates. That is, for each camera : $\hat{u} = M\hat{X}$ where $\hat{u} = [u, v, 1]^T$ and $\hat{X} = [X, Y, Z, 1]^T$. The matrix M can be found using different well known calibration techniques [5, 8]. In order to compute the calibration matrix for each camera, the user enters the world coordinates [X,Y,Z] of a minimum of six points and their corresponding coordinates [u,v] in the image. From these values, we can derive these equations : $u_i = (M_1 * \hat{X}_i)/(M_3 * \hat{X}_i)$ and $v_i = (M_2 * \hat{X}_i)/(M_3 * \hat{X}_i)$ where i = 1,2...n (number of points). Those equations can be restated in the following way : $(-u_i * M_3 + M_1) * \hat{X}_i = 0$ and $(-v_i * M_3 + M_2) * \hat{X}_i = 0$. Using these two equations, we obtain :

$$A = \begin{pmatrix} \hat{X}_1^T & O^T & -u_1\hat{X}_1^T \\ O^T & \hat{X}_1^T & -v_1\hat{X}_1^T \\ & ... & \\ \hat{X}_n^T & O^T & -u_n\hat{X}_n^T \\ O^T & \hat{X}_n^T & -u_n\hat{X}_n^T \end{pmatrix}$$

where $O^T = [0, 0, 0, 0]$. Finally, we solve the following : A $\begin{pmatrix} M_{11}M_{12}M_{13}M_{14}M_{21}...M_{33}M_{34} \end{pmatrix}^T = 0$

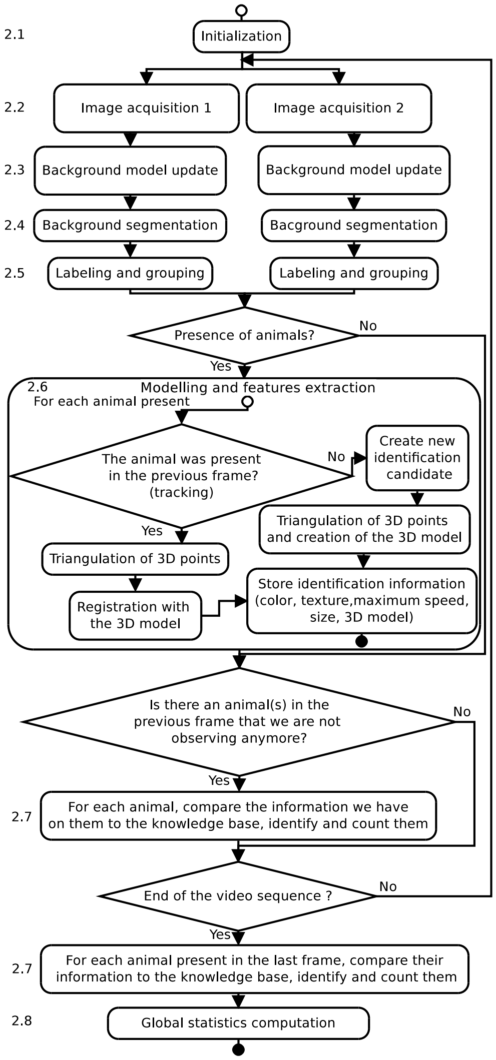
2.1
Initialization
2.2
Image acquisition 1
Image acquisition 2
2.3
Background model update
Background model update
2.4
Background segmentation
Bacground segmentation
2.5
Labeling and grouping
Labeling and grouping
Presence of animals?
No
Yes
2.6
Modelling and features extraction
For each animal present
The animal was present in the previous frame? (tracking)
No
Create new identification candidate
Triangulation of 3D points and creation of the 3D model
Yes
Triangulation of 3D points
Registration with the 3D model
Store identification information (color, texture,maximum speed, size, 3D model)
Is there an animal(s) in the previous frame that we are not observing anymore?
No
Yes
2.7
For each animal, compare the information we have on them to the knowledge base, identify and count them
End of the video sequence ?
No
Yes
2.7
For each animal present in the last frame, compare their information to the knowledge base, identify and count them
2.8
Global statistics computation

Figure 1. Flowchart of the main algorithm

using the smaller eigen vector corresponding to the smaller eigen value in the singular value decomposition (SVD) to obtain M. $[M_{11}M_{12}M_{13}M_{14}M_{21}...M_{34}]^T$ corresponds to the last column of the matrix V in the decomposition. To compute the initial background model, five frames are needed at the beginning of the video sequence without any animal in it. The mean and variance of each pixel in the three color channels is computed as the model.

2.2. Image acquisition

At this point, the system needs to obtain one frame from each camera. Ultimately, the system is meant to work in real time. Currently, the video sequences are stored in a database and the algorithm retrieves the frames as required. The possible camera configurations are : two visible cameras, two infrared cameras or only one of them. Using two cameras, frames are aquired simultaneously in order to use stereo vision for 3D reconstruction. It is also currently possible to only detect animals using a single camera. One can observe in Figure 2 that aquiring from an infrared camera is likely to simplify the background segmentation. However, using infrared cameras may make the task of 3D reconstruction harder, because it could be more difficult to match the corresponding points in the two views.

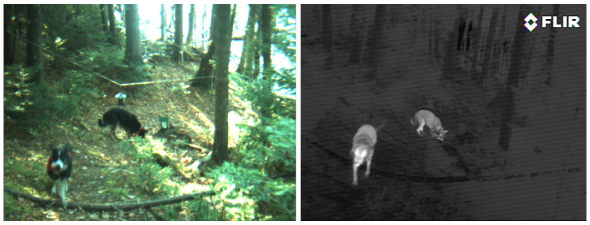

Figure 2. Visible and infrared images

2.3. Background model update

Illumination conditions change over time. Besides, non-living things may be moved by the animals. Hence, the system needs to slowly modify its background model. Currently, the background model is updated at each fifty frames to avoid slowing down the entire algorithm. The values of one of the five background images are modified (the five are sequentially modified, but only one at each iteration), by taking 90% of the value of the background image and 10% of the current analyzed frame. This avoid capturing too much noise or value of pixels currently occupied by an animal. After this, the background model is computed from the average of the five background images and the variance background model is similarly updated.

2.4. Background segmentation

At this point, the background pixels are identified. The values of each pixel are compared using Euclidean distance with the values of the background model. The variance of the pixel is also taken into consideration in the comparison. If the result of the comparison is greater than a threshold, we consider it as foreground and a candidate to be identified. The value of the threshold depends on the global illumination of the scene. It is computed from the average intensity of the pixel and the difference between the darker and lighter pixels.

2.5. Labeling and grouping

Once the segmentation is obtained, foreground pixels need to be grouped for each animal. First, the system labels each pixel, doing an eight connectivity label propagation. Each pixel and its eight neighbors are sequentially checked to assign a label to each pixel that is connected. Any pixel with no foreground neighbors is assigned to the background. Since the noise may produce disconnected objects, blobs closer than 10 pixels are also connected. Figure 3 presents a segmentation and grouping result.

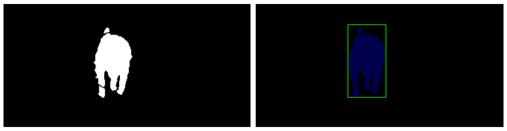

Figure 3. Segmentation and grouping

2.6 Modelling and feature extraction

2.6.1 Tracking

Each currently detected object may needs to be associated with an object detected in the previous frame. Object bounding boxes are tracked in 2D on the basis of proximity. An extrapolation of the motion of bounding boxes is obtained from the five previous frames before associating a box to the nearest one in the current frame. Figure 4 presents a tracked bounding box result.

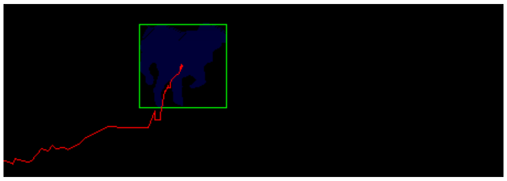

Figure 4. Tracking

2.6.2 Triangulation of 3D points

We are currently working on the 3D reconstruction stage of the algorithm. The principle is simple: the projectors of corresponding image points intersect at the projected 3D point. When the cameras are in canonical configuration, retrieving the 3D coordinates is easier because it corresponds to a simple 2D trigonometric problem, as shown in Figure 5. With two images in canonical form, world coordinates are computed as follows [7]: $Z = bf/d$ and $X = -b(2u + u')$ and $Y = bv/d$ where $d = (u' - u)$. Knowing the real world coordinates of one point is enough to isolate the baseline distance and the focal length : $b = Yd/v$ and $f = Zd/b = Zd/(Yd/v) = Zv/Y$. In order to reduce errors, all known point coordinates used in the calibration step are used to compute an average result. In the canonical configuration both image planes are in the same plane and their respective epipolar lines are parallel and at the same height. A transformation is computed to rectify corresponding frames [3]. Figure 6 presents the result of the rectification.

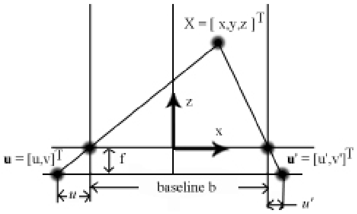

Figure 5. Canonical configuration

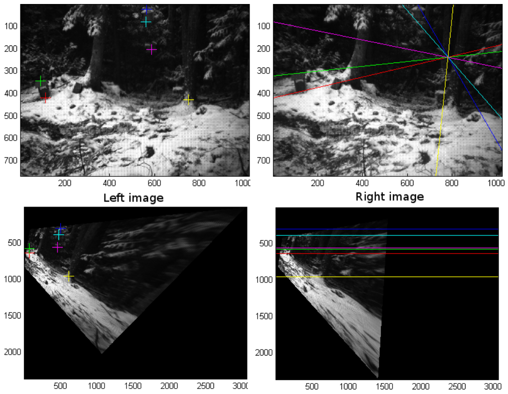

Figure 6. Result of the rectification

Once images are rectified, it is known from the epipolar constraint that a point in camera 1 may only be found in camera 2 at the same row. The score of a putative correspondence is related to the similarity of the local appearances. A small value of the following equation corresponds to a high matching score.

$$\sum_{i=-3}^{3}\sum_{j=-3}^{3}(Im_1(i,j,:) - Im_2(i,j,:))^2 * (e^{-\frac{i}{2}} * e^{-\frac{j}{2}})$$

where $Im_b(i,j,:)$ is the value of all the channels of the pixel at the position (i,j) in image b. Once all best matches are identified, the system triangulates using only the points with highest matching scores. The system also stores neighbor values of each correctly matched point in memory to be able to track local regions in future frames.

2.6.3 Registration with the 3D model

3D points obtained at a given frame need to be registrated to the current model before it is updated. A RANSAC approach [2, 6] is to be used to estimate the transformation. Reconstructed 3D points from the current model and those in the current frame are matched again using the local appearances. Then three points are randomly selected from the list of best matches and a global transformation is computed from their coordinates in the current frame and the previous one. After that, the system verifies if the transformation fits other good matches. If not, the system takes three other randomly selected points and repeats the process until a good result is found. If the approach does not converge to a good result after a number of trials, the best result is kept. Because an animal is a deformable object, the system needs to detect if some parts of the animal have moved differently. The approach taken is based on the skeletal signature of the 3D object [1]. If the system has at least 3 points from a skeleton branch in the list of good matches, it can compute the specific motion of that branch. At the end of this step, the new 3D points (those that are not in the list of good matches) are added to the 3D model of the object and the points from the deformed parts are updated.

2.7. Comparison and identification

A comparison is made between the knowledge base and the features extracted from the observed objects. We plan to merge existing approaches to achieve the 3D comparison. One of the most interesting techniques is the one proposed by S. Biasotti in reference [1]. A skeletal signature of the 3D object is computed. That signature is then used as a size graph to compute discrete size functions, giving a similarity measure between shapes. A template matching algorithm [4] will also be used. Here the idea is to generate 2D projections of 3D models in key views to be matched to observed 2D silhouettes. Template matching can be difficult because animals are deformable objects, but the system is to try both approaches and keep the stronger result. Every identification criterion we retrieve from these techniques and from our earlier observations will be taken into account in a probabilistic equation to achieve identification of the objects. The system also needs to score its confidence in the identification, so the user can verify the most uncertain identifications. At the end of this step of the algorithm, it will be determined whether the observed object is an animal and wich one in its knowledge base is the most accurate match.

2.8. Globals statistics computation

The final step of the algorithm is computing statistics and producing a graphic showing the number of detected animals over time. Currently, the user can interact with the graphic to view each detected animal on the video sequence. He can also observe the result of the 3D reconstruction algorithm. In the future, this graphic will be grouped by animal class and the user will be able to obtain the confidence in each identification.

3. Technical information

The code is written in C++ inside the Qt environment using OpenCV and OpenGL. Releasing one's work and knowledge with a free and permissive license allows people all around the world to optimize, modify and use results according to their needs. The lib3ds is used to integrate our 3D models. For the acquisition, Point Grey Dragonfly2 color cameras and Flir PathFindIR infrared cameras (spectral range 8 to 14 um) are used.

4. Conclusion

The approach presented integrates existing computer vision techniques to automate the detection and identification of animals. This research field still presents interesting challenges to be solved in the future.

References

[1] S. Biasotti, D. Giorgi, M. Spagnuolo, and B. Falcidieno. Size functions for comparing 3d models. *Pattern Recognition*, 41(9):2855 – 2873, 2008.

[2] M. A. Fischler and R. C. Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6):381–395, June 1981.

[3] A. Fusiello, E. Trucco, and A. Verri. A compact algorithm for rectification of stereo pairs. *Machine Vision and Applications*, 12(1):16–22, July 2000.

[4] N. Gupta, R. Gupta, A. Singh, and M. Wytock. Object recognition using template matching. http://stanford.edu/∼nikgupta/reports/cs229-report.pdf.

[5] Y.-H. Kwon. Direct linear transformation camera calibration. http://www.kwon3d.com/theory/dlt/dlt.html.

[6] M. Leventon, W. Wells, and W. Grimson. Multiple view 2d-3d mutual information registration. *Image Understanding Workshop*, 1997.

[7] M. Sonka, V. Hlavac, and R. Boyle. *Image Processing, Analysis, and Machine Vision, Third Edition*. Thompson Learning, Toronto, Ontario, Canada, 2008.

[8] Z. Zhang. A flexible new technique for camera calibration. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(11):1330–1334, 2000.