

A Method to Detect Victims in Search and Rescue Operations using Template Matching

Carlos Castillo
Grupo de Inteligencia Artificial
Universidad Simón Bolívar
Caracas 1080, Venezuela
Email: carlos@gia.usb.ve

Carolina Chang
Grupo de Inteligencia Artificial
Universidad Simón Bolívar
Caracas 1080, Venezuela
Email: cchang@gia.usb.ve

Abstract—We present an approach to vision-based person detection in robotic applications that uses template matching. We detect people using templates of the human silhouette. In our approach, we detect borders on each image, then apply a distance transform, and then match templates at different scales. Our system integrates silhouette, corners (point of interest) and skin presence in order to obtain more robust results in the detection of victims in search and rescue operations. Further, we describe the automatic generation of templates from a set of photographs of the object of interest.

I. INTRODUCTION

Detection and recognition of objects from images disregarding orientation, scale and view is a very important research subject in computer vision [3], [14]. People detection in images and video sequences is a research subject in this area. We are interested in this problem from a robotic application point of view since we are currently in early development stages of a robotic application for search and rescue operations [2], [6].

Some challenges with person detection are:

- **Pose:** The human body is a non rigid object that can take a wide variety of poses. From a pattern recognition point of view, each type of pose is a different classifier.
- **Structural components:** Structural components such as clothes are different from person to person, so there is great variability from pattern to pattern.
- **Occlusion:** Objects can be partially occluded by other objects.
- **Image orientation:** The human body varies for different rotations with respect to the camera's axis.
- **Image conditions:** Conditions such as lighting and camera characteristics affect the appearance of the human body.

The problem of people detection is very complex and has not been solved in its generality, but there have been advances where the pose is fixed, such as in the case of pedestrians [14], [1], [13]. There have also been advances in integrating greater tolerance to variations using component-based detection. Many of the existing systems for people detection [8], [14] use movement as a focus of attention (assuming a stationary camera) and use a large scale

classification system (such as SVMs) directly around the region that registered movement. In the case of Wren et al. [14], domain specific scene analysis is required. These methods are clearly unfeasible for a robotic application. Recognizing objects in images taken by a moving camera (in this case a fixed camera carried by a moving robot) is much more complicated than real time tracking with a stationary camera.

Our approach uses fast template matching as a focus of attention similar to the system to detect pedestrians in frontal poses described by Castillo and Chang [5]. Basically, it discards locations where there is no silhouette matching the human body. Our operative definition of a victim includes two cues: a silhouette and some visible skin. Further, our method seeks to detect potential victims in non-vertical poses.

The contributions intended are two-fold: first, the design and implementation of a general vision system that integrates multiple cues to detect potential victims, and second, a concrete implementation on board a robot in an embedded application.

The rest of the paper is organized as follows. First we describe distance transforms for template matching, and the corner detection approach we used, after that, we describe the system details, then the results are presented. Finally, the discussion and conclusions are presented and ideas for future work are given.

II. DISTANCE TRANSFORM FOR TEMPLATE MATCHING

A distance transform (DT) converts a binary image (containing values 0 and ∞) to an image where each pixel value denotes the distance to the nearest feature pixel. From this definition of the distance transform problem, a $O(n^4)$ algorithm can be readily constructed (for an $n \times n$ image). However, over the last 20 years the state of the art has advanced either approximating the EDT (Euclidean Distance Transform) in a $O(n^2)$ time or providing an exact solution in a $O(n^3)$ time.

Many DT algorithms exist, the differing characteristic is the distance metric and the propagation of local distances. In particular, we use Euclidean distance and Maurer's line-column scanning method [9]. Figure 1 shows the conversion

process from an initial frame to distance transformed image ready for template matching.

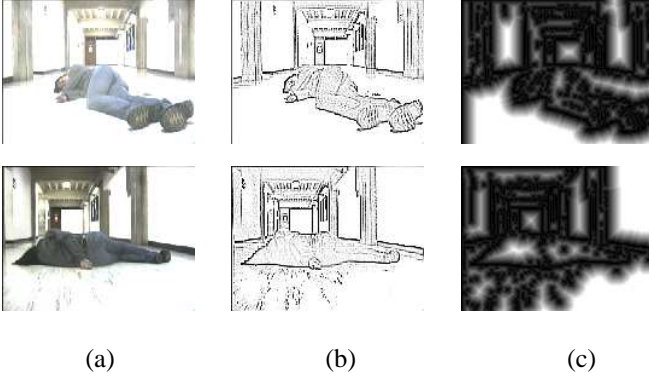


Fig. 1. (a) is the original image, (b) is the contoured and grayscale image and (c) is the distance transformed image ready for template matching.

After the image has been adequately preprocessed, the template matching step begins. As described by Gavrilu [7], a given image I is said to be matching a template T when:

$$D(T, I) \leq \theta \quad (1)$$

where θ is a user defined threshold on the maximum acceptable dissimilarity between the DT image and the template. $D(T, I)$ can be defined in several ways. In our system we experimented with two different definitions: mean alignment error and maximum alignment error. In the case of mean alignment error, $D(T, I)$ is given by:

$$D(T, I) = \frac{1}{|T|} \sum_{t \in T} d_I(t) \quad (2)$$

where $|T|$ is the number of features in T and $d_I(t)$ is the distance between feature $t \in T$ and the closest feature in I .

In the case of maximum alignment error, $D(T, I)$ is given by:

$$D(T, I) = \max_{t \in T} d_I(t) \quad (3)$$

$d_I(t)$ is the distance between feature $t \in T$ and the closest feature in I as calculated by the EDT (Euclidean Distance Transform).

The mean alignment error quantifies the alignment error as the average (mean) distance of all the individual pixels in the template while the maximum alignment error quantifies the alignment error as the distance of the worst aligned pixel in the template.

III. CORNER DETECTION

Our criterium for interest point (region) is a point that is not highly correlated to its neighbors. Perceptually these regions are called salient. There are many algorithms to

detect salient points; we use a very simple (and fast) one which is described in this section.

To detect corners, we use Sobel first derivative operators [10] to take derivatives of the image. A small region of interest is defined to detect corners in. A 2×2 matrix of the sums of the derivatives is created as follows:

$$C = \begin{pmatrix} \sum D_x^2 & \sum D_x D_y \\ \sum D_x D_y & \sum D_y^2 \end{pmatrix} \quad (4)$$

The eigenvalues are calculated by solving $\det(C - \lambda I) = 0$. If $\lambda_1 > t$ and $\lambda_2 > t$, where t a constant threshold, then there is a corner (point of interest) at that location.

The matching process is done only over the regions of interest. To do this we incrementally compute the number or corners in a given matching window.

IV. SYSTEM DETAILS

In this section we describe how our system works. The operation of the system is divided in two phases. The first phase is the generation of victim templates from photographs, the second phase is the online victim detection from these templates. Our system was developed in C++ using the Intel's OpenCV Library.

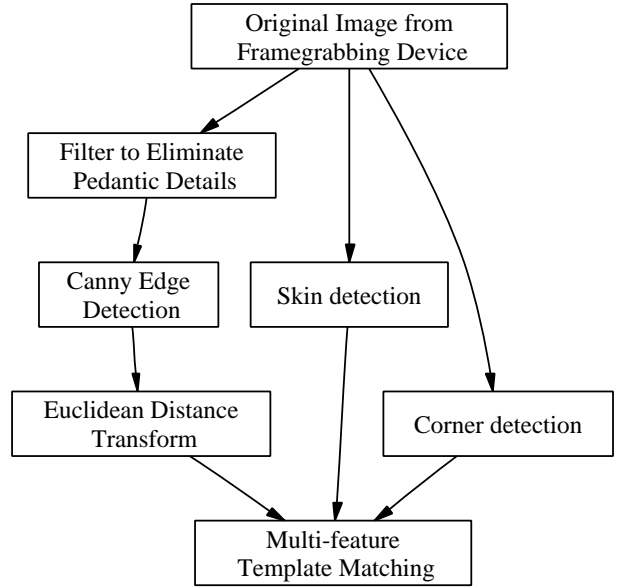


Fig. 2. A diagram of our victim detection procedure.

A. Generating Templates from Photographs

The image of the object of interest is contour filtered and thresholded. This process defines a resulting image. The resulting image is seen as a graph G where the data (black) pixels are the nodes and the arcs are the immediate neighborhood relation to other data pixels. On this graph, we calculate strongly connected components using DFS (Depth First Search) and we proceed to eliminate connected components that include less than τ nodes.

Figure 3 shows this process applied to several images. Figure 4 shows an example of the calculation of connected components to obtain a template of interest in a given image. The top part of Figure 4 shows the original thresholded image and the bottom part shows the equivalent neighborhood graph. On the equivalent graph, strongly connected components are calculated. The large connected components (larger than a experimental set threshold τ) are considered to be a template of the object of interest (see for example the (c) column of Figure 3).

B. Victim Detection

To detect victims we integrate two cues: a silhouette and skin presence.

1) *Detecting the silhouette:* At its core, the system uses template matching employing Euclidean distance transform (EDT) to evaluate candidate victims using templates obtained from the photographs.

The very first step is preprocessing. Each input image is grayscaled and contour-filtered using the Canny edge detector [4], [12]. After that, the contoured and grayscaled (CG) image is transformed using an EDT.

The image is scanned for matching silhouettes. We have devised two simple methods for image scanning:

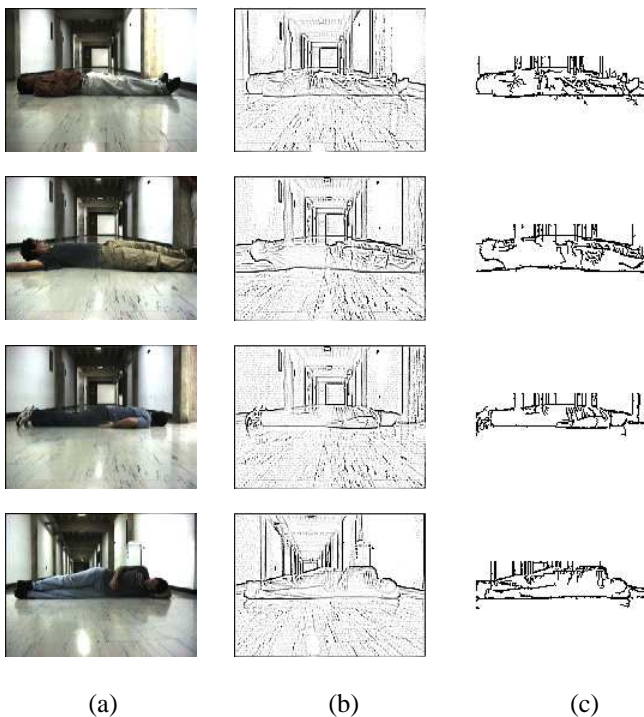


Fig. 3. Template extraction from a photograph: (a) is the original image, (b) is the contoured and grayscaled image and (c) is the strongly connected component from the thresholding of the neighborhood graph shown in (b); this image is used as template.

- Using exhaustive scanning. In an $X \times Y$ image with an $N \times M$ template, we first try to match the window

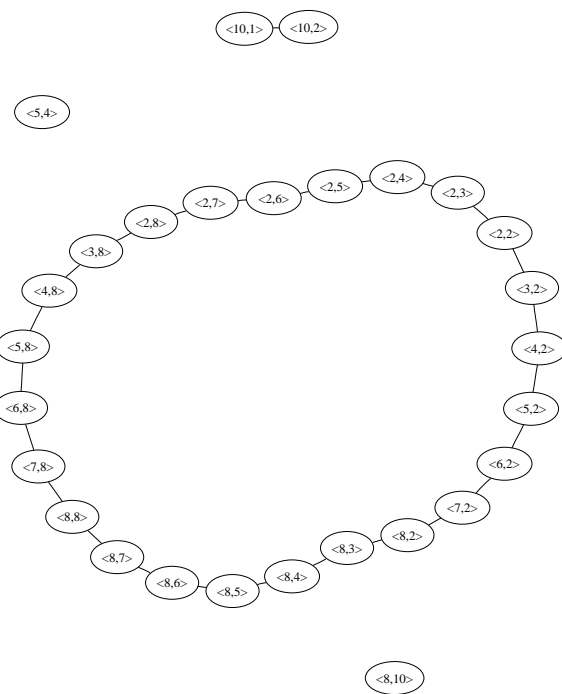
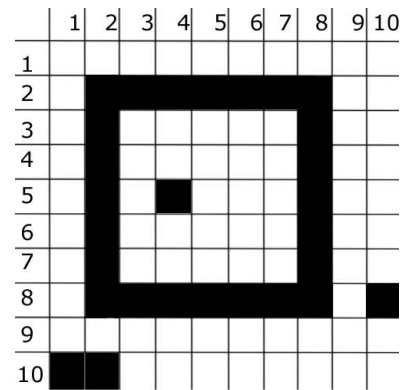


Fig. 4. An image (top) and the equivalent neighborhood graph (bottom).

defined by the rectangle $(0, 0, N, M)$; after that the one defined by $(1, 0, N + 1, M)$, and so on until reaching the end of the image at that scale.

- Using random sampling. In an $X \times Y$ image with an $N \times M$ template, we select a fixed number of samples proportional to the size of the image. This scanning method accelerates the process with a sacrifice in precision.

In the offline experiments we use exhaustive scanning because runtime performance is not an issue. The online version also uses exhaustive scanning. However, note that the online version could be made faster by using random sampling, but in this case not all positions in the image will be scanned in each frame.

After experimentation we settled with 12 templates. The 12 templates include victim lying in both directions (left to

right and right to left) and therefore can detect in close variations of left-to-right and right-to-left. The matching process over a distance transformed image allows to capture great variability in the pose of the victims.

More templates means a better definition of the class of interest but also translates into a slower matching process. The templates are taken from photographs of the object of interest after contour filtering it and obtaining the relevant connected components.

2) *Detecting skin:* We use a segmentation method to detect skin using YCbCr chrominance information[11]. This method is more robust because it uses more domain information than when using an SVM (Support Vector Machine) as previously proposed [2]. In our current implementation, the presence of skin is used to relax the matching threshold (θ) in case of presence of skin.

If in the evaluation of a given rectangle of area a there is between 2% and 20% skin, the matching threshold is relaxed (increased) by 10%. The skin presence cue helps solve some occlusion problems very elegantly.

It is important to note that the presence of skin helps in the victim detection but it is not a limiting factor.

We tested the system onboard an ActivMedia Robotics Pioneer 2 mobile robot. The online version (onboard the robot) uses the randomized scanning method previously described.

The performance (as measured by false positives and false negatives) degenerated significantly. To handle this we adjusted (downwards) the value of θ in the template matching step. Further, to enhance the precision of the system in our office environment (our Pioneer II robot is not designed for outdoor use), we measured the correlation of the value pixels on the DT image over the template as described in equations 2 and 3 (we called this value β) and measured the percentage of matching non-data points in the template compared to the contoured image (we called this value α). So the matching criteria is:

$$\frac{\alpha}{\beta} > \gamma \quad (5)$$

where γ is an experimentally set threshold value. The matching criteria seeks a balance of many matched points with low matching error (derived from the distance measure of the EDT image). This refinement of the matching criteria significantly decreases the false-positive rate (in our case by over 12%).

The online version of the system works at 3 Hz. This speed is slow because we exhaustively scan the image at each frame.

The offline version (same online version but instead of loading the image from the framegrabber, loads it from a file) was tested with our 700 image database¹; the correct detection rate was 78%.

¹The image database is available, please contact the authors for details.

V. EXPERIMENTAL EVALUATION

In this section we describe the experiments we performed to assess the feasibility of our approach.

A. Initial experiments

The first part of our experimental evaluation consisted in trying to detect victims. After verifying that it worked with real victims, we tried to make it look at things that had the look of victims but actually weren't. From the initial experimentation we learned that the templates obtained didn't match much more than the object of interest (victims) in our office environment and that it was quite hard to fool (create an artificial scene) to make the system detect non-victims as victims.

However, in a highly cluttered scene, victim detection is harder and the system tends to fail more. The parameters of the Canny edge detector and the corner detector could be modified so that the clutter affects the detection process less but it is still a problem that is hard to avoid.

The other type of error the vision system could make was to not detect victims when present. In this regard, we found that our multi-scale scanning algorithm was very sensitive to the contexture (or build) of the person in the scene. We found two solutions to this problem: scanning in more scales (using a smaller step between scales) or including a wide variety of body builds in the templates used. Both solutions entail more processing time per frame.

B. Evaluation in a more realistic situation

We put a victim in some place in a long hall; the lighting conditions, pose and rescue scenario were all realistic. The robot was remotely operated by someone who knew in great detail the way the system functioned.

We evaluated if the robot could successfully make the detection and if it did, we measured the minimum and maximum distance to the victim where the robot made the detection, and also evaluated the performance of the robot in the presence of debris occluding the victim.

In Fig. 5, the system proposed adequately detects at 2 meters the presence of a victim when there is office debris occluding the view.

In Fig. 6, the system proposed detects a victims in a range from: 2.6m to 4.5m.

In Fig. 7 the system's resilience to occlusion around non-critical partial features is shown. The system cannot correctly detect a person when the object of interest is more occluded than (c).

VI. CONCLUSIONS

We have presented an architecture for recognition of victims in search and rescue situations. It is the first time this problem is studied from a computer vision point of view. The method applied is very simple and scales well to larger datasets and can be used in real time applications. The method copes well with natural variations in the pose of the object of interest.



Fig. 5. Detecting a victim around office debris. (a) office debris, (b) victim occluded by the debris and (c) the system detecting the victim

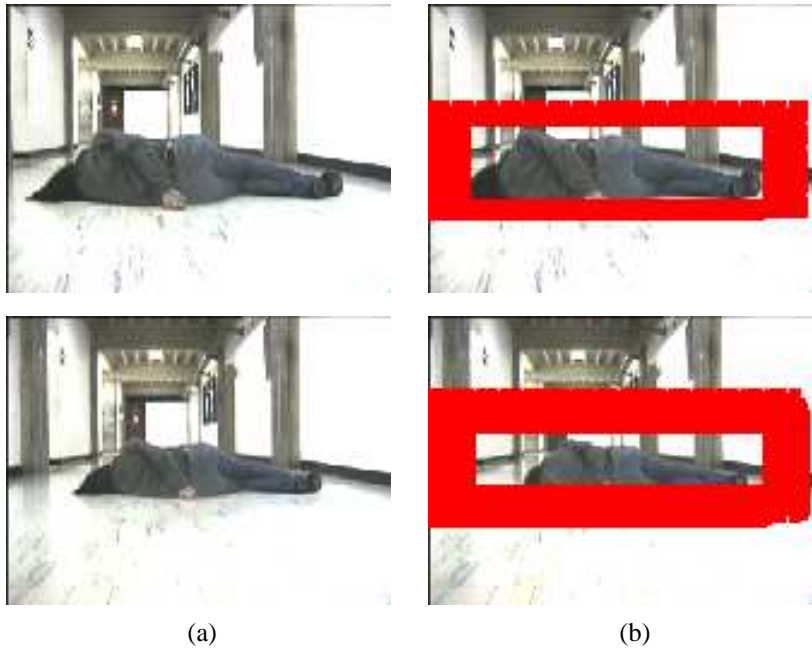


Fig. 6. Detecting a victim at various distances. (a) original image, top: nearest detection distance, bottom: furthest detection distance, (b) marked image

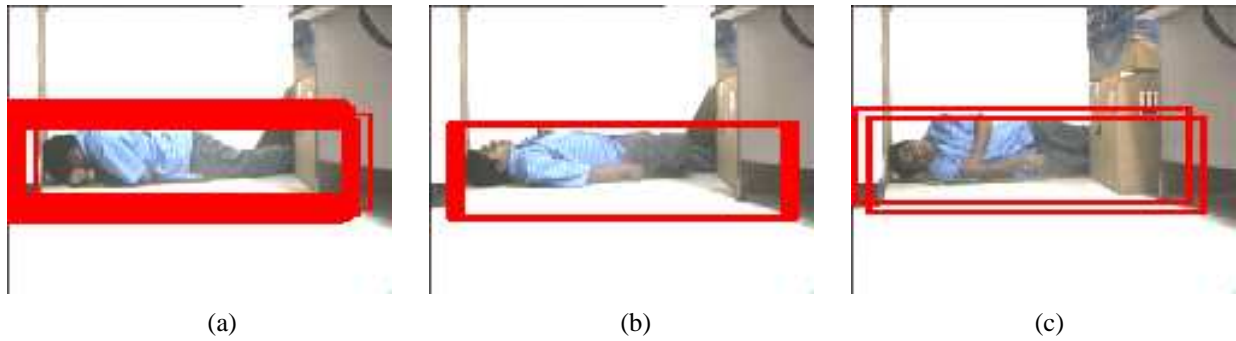


Fig. 7. Resilience to occlusion around non-critical partial features. (a) not very occluded view, (b) moderately occluded view and (c) very occluded view.

This application needs to be improved in order to become more useful since it would fail under many conditions: occlusion, low contrast, debris. It constitutes, however, a first step in using computer vision systems to detect victims in search and rescue operations.

Our indoors robot is not suited to perform outdoors search and rescue tasks, yet this research is allowing us to gain some insight about the USAR robotics domain.

ACKNOWLEDGEMENTS

We would like to thank the various model victims for their patience and cooperation, in particular: Miguel Castro, Eduardo Ruiz, Julio Castillo, and David Ojeda.

REFERENCES

- [1] M. Bertozzi, A. Broggi, R. Chapuis, F. Chausse, A. Fascioli, and A. Tibaldi. Shape-based pedestrian detection and localization. *Procs. IEEE Intl. Conf. on Intelligent Transportation Systems*, pages 328–333, 2003.
- [2] A. Brando and C. Chang. Firefighter-robot interaction during a hazardous materials incident exercise. In *11th International Conference on Advanced Robotics*, volume 2, pages 658–663, 2003.
- [3] A. Broggi, M. Bertozzi, R. Chapuis, F. Chausse, A. Fascioli, and A. Tibaldi. Pedestrian localization and tracking system with kalman filtering. *Procs. IEEE Intelligent Vehicles Symposium*, pages 584–589, 2004.
- [4] J. F. Canny. A computational approach to edge detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 8(6):679–698, 1986.
- [5] C. Castillo and C. Chang. An approach to vision-based person detection in robotic applications. In *2nd Iberian Conference on Pattern Recognition and Image Analysis*, volume 2, pages 209–216, 2005.
- [6] C. Chang and A. Brando. Semi-autonomous victim search. In *Proceedings of the 3rd IEEE International Workshop on Safety, Security and Rescue Robotics*.
- [7] D. Gavrilu. Pedestrian detection from a moving vehicle. *Proc. of the European Conference on Computer Vision*, 2(8), 2000.
- [8] B. Heisele, C. Nakajima, M. Pontil, and T. Poggio. People recognition in image sequences by supervised learning. Technical Report CBCL-188, MIT Artificial Intelligence Laboratory, June 7 2000.
- [9] C.R. Maurer Jr. and V. Raghavan. A linear time algorithm for computing the euclidean distance transform in arbitrary dimensions. In *IPMI*, 2001.
- [10] E. P. Lyvers and O. R. Mitchell. Precision edge contrast and orientation estimation. *IEEE Trans. Pattern Anal. Mach. Intell.*, 10(6):927–937, 1988.
- [11] C. T. Hsu M. J. Chen, M. C. Chi and J. W. Chen. Roi video coding based on h.263+ with robust skin-color detection technique. *IEEE Transactions on Consumer Electronics*, 2003.
- [12] D. Marr and E. Hildreth. Theory of edge detection. *Proc Roy. Soc. London*, page B207:187, 1980.
- [13] C. Papageorgiou and T. Poggio. Trainable Pedestrian Detection. In *Proceedings of the 1999 International Conference on Image Processing (ICIP-99)*, pages 35–39, Los Alamitos, CA, October 24–28 1999. IEEE.
- [14] C. R. Wren, A. Azarbayejani, T. Darrell, and A. Pentland. Pfunder: Real-time tracking of the human body. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(7):780–785, 1997.