SEVENTH FRAMEWORK PROGRAMME

VIT
Vision for Innovative Transport

Project partly funded by the EC
Grant agreement no. *222199*

SP4-Capacities - Research for SMEs

REPORT ON BETA VERSION OF SOFTWARE PACKAGE (OBJECT CLASSIFICATION AND TRACKING FROM SINGLE CAMERA)

Deliverable D5.1

Release date 7 July 2009

23 October 2009 - Reviewed after EU Technical Expert comments

Work package number WP5
Work package title People security with video analysis
Activity Type RTD

About the Document

This document reports technical details of beta version of software package for robust object classification and tracking from single camera; it reports the content of the developed software modules and the documentation about how to use it. Also it reports the activity of *Tasks 5.2 Statistical learning to support decision* and *Task 5.3 Video analysis and robust tracking* from months 4 to month 12 inside the Workpackage *WP5:* People security with video analysis.

The document has been produced by the collaboration of the workpackage WP5, the participants of the workpackage have all duly contributed to the activity of the workpackage and the production of this document and they endorse the final version as the conclusion of the workpackage.

Workpackage leader

Alberto Lovato (IMA)

Document authors

Francesca Odone (DISI)
Alberto Lovato (IMA)
Emanuele Trucco (DUN)

Document reviewers

Fabio Tarantino (ILOG)
Thomas Keese (WITT)

Table of contents

INTRODUCTION **4**
STRUCTURE OF THE REPORT 4
AUDIENCE **4**
1. TESTING SCENARIOS **5**
2. OBJECT TRACKING **5**
RESULTS 6
REFERENCES 6
3. STATISTICAL LEARNING TO SUPPORT DECISIONS **6**
PEDESTRIAN DETECTION 7
RESULTS 8
BLOB CONTOUR REGULARITY 9
RESULTS 10
SIZE FUNCTION 10
REFERENCES 11
4. COMPARISON AMONG CLASSIFICATION METHODS **12**
5. HOW SOFTWARE RESPONDS TO USER REQUIREMENTS **16**
HOW SOFTWARE RESPONDS TO USER REQUIREMENTS 16
HOW SOFTWARE MEETS CONSTRAINTS 16
HOW SOFTWARE MEETS MEASURABLE OBJECTIVES 16
6. PROCESSING PIPELINE **17**
7. USER MANUAL **18**

INTRODUCTION

WP5 aims to study and develop video analysis methods which are able to locate people in off-limit areas during load/unload operation of the Metrocargo system.

Following user requirements, the off-limit area is monitored by several video cameras which cover all the area and also the adjacent perimeter. The devised processing pipeline analyzes in real time every single video flow, tracking moving objects and classifying them as people or mechanical devices. The vision algorithm sends alert messages to the system when people presence is detected.

Three different approaches to the problem of classification has been carried out and a comparison between the three has been done, choosing one of them for the test filed phase.

The purpose of the document is two-fold: to give an explanation of the developed method in order to fit the technical specification released in Deliverable 2.2 and to give a documentation about how to use the software package inside the system Metrocargo for the purpose of people security.

Structure of the report

The report is organized with the following structure:

1. Testing scenarios
2. Object tracking (task 5.3)
3. Statistical learning to support decisions (task 5.2)
4. Comparison among classification methods (task 5.2)
5. How software responds to user requirements
6. Processing pipeline
7. User manual

AUDIENCE

The present deliverable is filed as Confidential, as it contains critical information for the VIT project and also for the Metrocargo system.

Therefore the audience of the document is restricted the project participants --- the SME’s who will find the technical details following their user requirements and the RTD performers who will use the present report as a guideline of their research and development activity.

1. Testing scenarios

This section reports briefly the scenarios adopted during the first 12 months of RTD activity within WP5. At the beginning of the project, since the Vado Ligure plant was not yet available, we evaluated real testing scenarios reproducing the main features of Metrocargo environment. These scenarios have been the reference for the first risk assessment phase and for the first prototype of video analysis for people security.

We recall the main objective of the WP: to classify humans against mechanical man-made autonomously guided devices in motion. Since to the knowledge of the RTD performers involved in the WP there are no functioning terminals with automatic handling units, an agreement among the project partners was reached on the best approximation being a urban environment.

We have identified a testing area (besides one of Imavis commercial partners premises) where the RTD performers could have a full (direct and remote) access and all privacy issues had been dealt with, by means of pre-existing agreements.

Such an environment hosts a minority of walking humans, other people riding bikes and motorbikes, and a majority of man-made objects of different sizes (trucks, cars, motorbikes) moving at different velocities (some of them passing by, other parking) --- see Figures 3 - 8.

A video-surveillance system monitoring the area was available at the beginning of VIT project and was already recording motion events. This allowed for a live test since the very beginning of the development process. The system produces to the benefit of VIT RTD activity two kinds of material:

1. Video recordings used in the development and laboratory testing phase: a set of about 100 manually labeled videos, the average length of which is 2.6 minutes, corresponding to about 4000 frames each. Such videos have been acquired in different conditions (cloud, bright sun, fog, rain) and different times of the day (including night) have been collected. 50 videos form the so called *training set,* used for training the modules, validating them, performing parameter tuning. The remaining videos are used for testing.
2. Live video stream: thanks to a 24H full access to the system, the developed modules can be evaluated directly on the acquired video stream live. This is useful to check real-time performance and robustness to long time processing.

2. Object tracking

As outlined in D2.2 Technical Specifications, the first module developed within WP5, is an object tracking module, to the purpose of modelling actions performed by objects moving in the scene.

The developed tracking algorithm receives as an input a video stream for each camera and performs *(i)* background update*, (ii)* motion detection and *(iii)* object tracking.

According to methods well established in the literature, the background updating model takes into account the variance of each pixel in grayscale in a fixed temporal interval and updates only those pixels with a small variance.

Then a background subtraction technique permits to detect the image regions moving with respect to the background. Afterwards connected components of pixels in motion are built, thus identifying moving blobs or units.

Once the algorithm has detected objects which are not part of the background, a tracking algorithm based on the position of the centroid of each blob is used to follow the trajectory of each of these objects in the scene.

To compensate common issues like segmentation errors, noise or occlusions, the algorithm uses a dynamic filter based on Kalman [Kal60]. This filter tries to exploit in an optimal way all the information given by the measurements, taking into account a certain amount of noise and also giving a prediction on the future state of the blob, given its history until current observation.

The most innovative part of this low level processing module, is an efficient data association method allowing for a correct tracking in situations where a group of moving objects overlap. For more details on this, the reader is referred to the paper [Noc09], presented by the RTD performers to an International Conference of the field.

As a special case, sudden stops of objects have been modelled. The temporal interval of the background update process can be configured to regulate the elapsed time before an object enters in the background, thus disappearing from the list of moving objects. Within this time, the algorithm keeps on detecting the stopped object, also assisted by tracking that expects to find the object near to the predicted position.

Results

The algorithm has been developed starting from a robust version of tracking already used by the involved RTD in other applications. The reliability of this algorithm is well established and the test performed in the chosen scenario has confirmed it.

We performed the test on the collected data presented in the Section 1.

On the single frame (without the use of tracking and Kalman filter) we have the following error percentage on the number of detected objects in motion:

- 7,5 % false positive (mainly due to illumination effects – shadows etc);
- 9,5 % false negative (mainly due to a failure on the background update, caused by still objects);

Using the temporal continuity induced by tracking and Kalman filter the error percentage (on the object trajectory) decreases as follows:

- 2,2 % false positive (due to spurious objects splits);
- 3,6 % false negative (mainly caused by occlusions).

Tests on unlabeled live videos qualitatively confirmed these results.

References

[Kal60] R. Kalman, A new approach to linear filtering and prediction problems. Trans. of the ASME journal of basic engineering, 82, series D, 1960.

[Noc09] N. Noceti, A. Destrero, A. Lovato, F. Odone. Combined motion and appearance models for robust object tracking in real-time. IEEE International Conference on Advanced Video and Signal based Surveillance AVSS 2009

3. Statistical learning to support decisions

Once a moving object has been detected, the next objective is to automatically decide whether it is or not a person. This task can be cast in the more general problem of *object classification.*

This section reports the experimental evaluation carried out on a set of different approaches to the classification of blobs appearance. According to the user requirements the various approaches have been evaluated with respect to:

- their discrimination power and ability to generalize to new situations and new scenarios;
- their computational cost.

We started with object classification based on statistical learning, implementing feature-based pedestrian detection methods. The devised methods produced promising results, but to the price of a rather high computational cost. For this reason we resorted to use simpler classification rules, based on shape descriptions. In this case we developed and evaluated two different techniques both applied to the blob contour: the analysis of contour regularity and the use of the Size Function as a contour shape description. The reminder of the section presents a brief account of the research carried out.

Pedestrian detection

All methods presented are designed to locate objects of interest by scanning each image using a window with a fixed aspect ratio and then selecting the windows that contain the object of interest.

Rectangular features

As a first approach we have developed a method using rectangular features, first introduced in [Vio02] in the context of face detection algorithms. An over-complete set features has been computed by shifting and repositioning a set of masks into all the possible steps.

To address the large dimensionality of the data, we applied a process of selecting variables in a sequence of smaller sub-problems obtained by sampling feature vector. In a second phase we combine the results of previous runs by selecting only the features that have been chosen in all the sub-problems in which it was drawn. This set of features is then further reduced in the final round of feature selection to obtain the final set of descriptors. The RTDs involved in the project are experienced with this feature selection + classification pipeline [Des07].

A thorough experimental evaluation carried out in the first months of the project, underlined the fact the performance of these features for the problem of pedestrian detection is not satisfactory. At the same time, the nature of these “focus of attention approaches” make them a good candidate to simply dismiss the negatives and then to identify areas of interest on top of which to apply slower more complex methods. For this reason this approach, associated to a different set of features, has been considered in successive developments, as confirmed in the reminder of the section.

Histogram of oriented gradient (HoG)

As a second approach we analyzed the image window dividing it into cells over which the histogram of the direction of the gradient is calculated. The cells are then analyzed in overlapping groups: all histograms of cells belonging to the block are concatenated and the content of the carrier is normalized. The final descriptor is the concatenation of vectors calculated over all the blocks [Dal05].

The performance of these descriptors are far superior to those obtained using the feature of [Vio02]. In the tests we noticed a strong instability in areas of particularly saturated images that are particularly susceptible to register false positives. This problem has been faced filtering all areas with a low standard deviation of the intensity of grey.

A tricky point of this algorithm is parameter tuning (bin of histogram of cells, cell size, number of cells per block, overlay block) which depend on the size of the window to be classified and on the characteristics of the sought object. We have used a validation set to select all the parameters before testing performance on the test set.

The main problem of this approach remains the high computational cost. The method can be combined with the pre-processing steps to limit the areas on which the descriptor must be calculated, but this reduces the goodness of results.

Covariance features

As a third approach we used a descriptor with characteristics similar to those presented in [Vio02]: for all the possible windows in different positions and scales within the image to be classified, the covariance matrix of some variables (position, gradient, form the gradient, second derivatives and gradient direction) is computed.

The set of features is then given by the matrices calculated in all configurations. As in [Vio02], in this case the calculation of these descriptors in a reasonable time is made possible through the widespread use of the technique for calculating the integral image feature of covariance.

For the classification stage in the literature there are two approaches: a first method classifies the covariance matrices on a manifold (because the covariance matrices are not a vector space) [Tuz08]. In the second case, the array is unwound in a carrier and is used directly for classification by building a cascade of classifiers with Adaboost [Yao08]; using a weighted version of LDA the descriptor is reduced to one-dimensional features, in order to use it within Adaboost.

In our work we have followed the second approach to meet the performance requirements in the classification stage. Serious convergence problems forced us to abandon this approach, in spite of the promising performances reported in the literature.

Results

Considering the data driven nature of this approach an appropriately sized pedestrian dataset has to be used for the training and validation phase). We exploited annotated data from benchmark datasets (Caltech pedestrian dataset[1], Inria person dataset[2], VISOR video-surveillance repository[3].

Figure 1 reports a comparison of the Receiver Operating Curves (ROC) obtained training two classifiers as in [Des07] and [Dal05] on the benchmark Caltech dataset and validating them on different videos from the Visor repository. The HoG descriptors are the best performing and therefore have been adopted for the following testing phases.

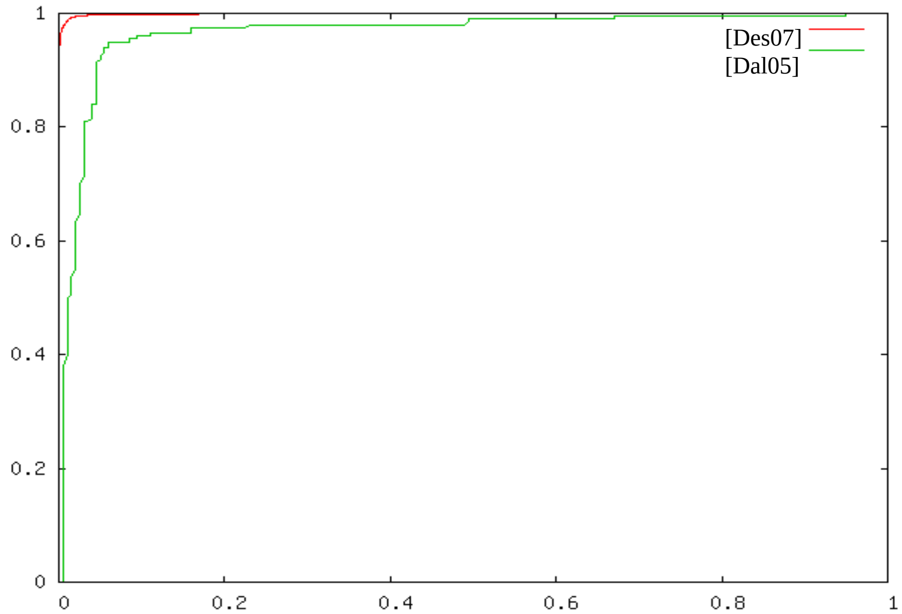

Figure 1 – ROC curve for two classifiers as in [Des07] and [Dal05]

We performed tests on the collected data presented in the section 1. Moving objects were located according to the method described in section 2 and then classified according to [Dal05].

The error percentage are:

- 2,3 % false positive (moving objects erroneously classified as people);
- 5,4 % false negative.

Tests on live images has confirmed these results.

While the results presented above are very satisfactory, we noticed that all these methods tend to be slow and cannot be cast in a real-time pipeline. An important point to make is that pedestrian detection can be a rather challenging problem if no prior is available. On the other hand the application we are considering can take advantage from the fact that the relative angle between camera and observed scene is known, and from the fact that objects moving in the scene are rarely people. Based on these considerations a set of simpler and more efficient methods based on shape analysis have been studied, developed, and validated. The reminder of the section reports the obtained results.

[1] http://www.vision.caltech.edu/Image_Datasets/CaltechPedestrians/
[2] http://pascal.inrialpes.fr/data/human/
[3] http://www.openvisor.org/

Blob contour regularity

The scenario where the analysis will be performed is the Metrocargo plant where there will be a few “authorized” moving objects, observed by cameras the position of which is well known, with specific and well known characteristic (in particular size and appearance). On the opposite side of the spectrum, with respect to feature-based statistical learning methods, we evaluated a very simple shape based classifier, that fully exploits all prior information on the environment.

The idea behind the algorithm we developed is that the shape of a moving mechanic object is more stable and constant than the shape of a person walking in the scene.

The algorithm analyses the shape of the tracked object. For each blob the algorithm sets a reference shape after a certain number of observations, when we can assume the object is not a false alarm (i.e., a false positive).

Then the algorithm compares the shape of each view with the shape of the reference one, returning a shape distance D, computed it as follows:

$$D(A, B) = \sum_{i=1...7} |m_i^A - m_i^B|$$

where

$$m_i^A = sign(h_i^A) \cdot \log h_i^A \quad m_i^B = sign(h_i^B) \cdot \log h_i^B$$

and h_i^A, h_i^B are the Hu moments [Hu62] of object A and object B respectively.

Then the algorithm updates a running variance of the distance.

Notice that, the idea of using a reference shape instead than a comparison between consecutive shapes also addresses the problem of slow motion or objects suddenly stopping.

At each frame the classifier decides if the object is mechanic object or not on the basis of a threshold: below the threshold the object is mechanic, above is a person. The threshold has been set on an appropriately selected small training set, (on the final prototype it can be built during the system calibration time).

The algorithm scheme is described by the flow chart in *Figure 2*.

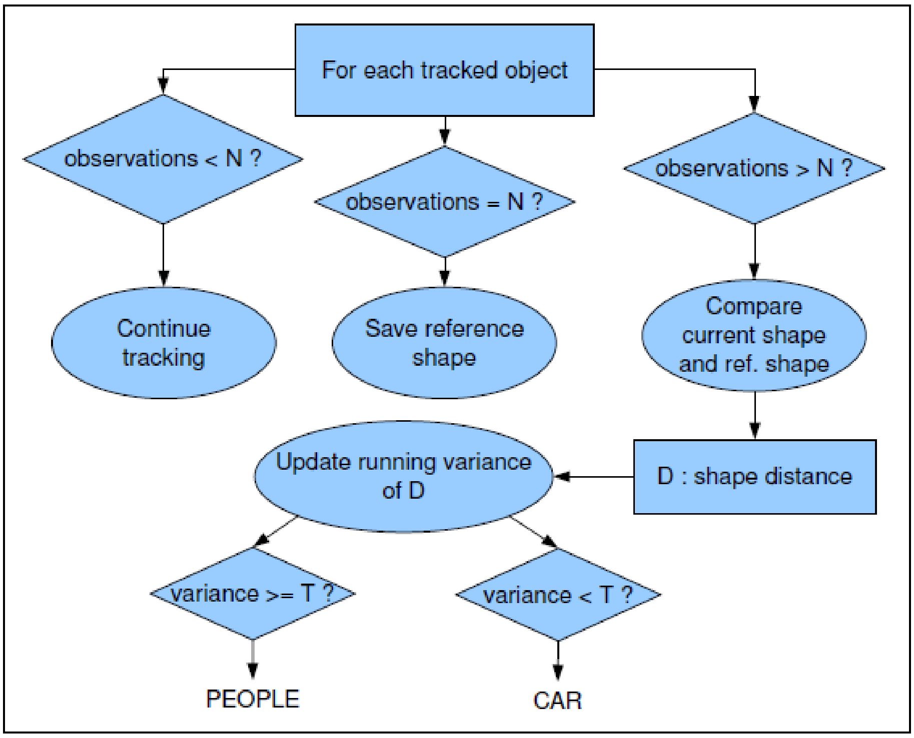

Figure 2 – Blob contour regularity algorithm scheme

To improve the method robustness temporal continuity is exploited: before an alert is given a certain number *N* of consecutive coherent answers have to be associated to each moving objects. Again, a meaningful *N* is estimated on an appropriately selected training set. The value has been currently set to 3, again after a parameter tuning phase performed on the training set.

It is maybe the case to underline that, given the fact that the method discriminates on the basis of shape stability, it makes confusion between people and animals. The SMEs have been updated on this issue, and reached the conclusion that, since plants could be visited by animals, a video-surveillance system should notice them and send the appropriate alert. This is particularly important to spare animals lives and also to limit the amount of damage to Metrocargo equipment.

Results

We performed the test on the collected data presented in the section 1.

The error percentages over 46 test videos on each detected blob are the following:

- 2,7 % false positives (moving objects erroneously classified as people);
- 7,2 % false negatives (mainly due to objects merge).

A significant improvement has been observed if temporal continuity is exploited. In the case the number of consecutive coherent observations required is N=3 frames, the estimated errors drop to 0,9% false positives and 3,6% false negatives.

Tests on live images has confirmed these results.

The results are very satisfactory, as the ones obtained with pedestrian detection, but with this method also the computational performances are satisfactory and they have successfully been proved in a real-time pipeline. Therefore this method has been implemented in the current software version and a field test evaluation, currently in process, is speaking in favour of this approach.

The main limitation of this approach is due by the simplicity of its decision function. This method does not meet entirely the requirements of having a good generalization ability with respect to new scenarios or unexpected behaviours. This requirement was indeed the reason why a learning from examples approach was proposed in the first place. To address this limitation, a rather more complex, but still computationally effective, shape classifier based on size functions is currently under evaluation. A description of it is reported in the remainder of this section.

The results obtained testing the algorithm on the remaining 4 videos, acquired under extreme weather conditions (1 with fog and 3 with very low illumination), are worst: 6% false positives and 9,2% false negatives (considering the small amount of data including in this set the percentages are less meaningful). This may be due to the fact that the devised classification procedure needs an improvement from the generalization stand point.

Size Function

A possible approach that was evaluated and proved interesting was the use of size functions as a shape descriptor to be associated to a classifier.

Size Function [Fro99] is a very powerful shape descriptor which extracts from the contour of the blob some specific information (through the computation of a so called *“measuring function”*) and describes them in terms of a limited number of points of a 2D plane exactly called Size Function of the blob contour according to a specific *measuring function*. A pair of blob contours can be compared computing the distance between the Size Functions of the two blobs.

We have built a training set of blobs belonging to the two classes (mechanic objects / people) and we have computed all the Size Functions of these blobs.

The current implementation of the classification module is based on a simple nearest neighbour classifier: given an object, the algorithm calculates its Size Function and compares it to those of the examples in the training set using the Euclidean distance between their nearest points as a metric. The object is then associated to the class (mechanical objects/ people) containing its nearest neighbour among the training set.

If the size of the training set is limited, the computational cost is under control: for each blob the computation of Size Functions has limited costs and the comparison with the training set weighs as an Euclidean distance of small set of points. As the training set size grows, other classification algorithms can be taken into account, e.g., Support Vector Machines [Vap98] (in this case the number of comparisons will be proportional to the number of support vectors and, for an appropriate choice of data representation and of a kernel function, this number will be much smaller than the whole training set size). It is worth noticing, though, that a nice feature of size functions is that they usually capture quite effectively appearance features from a limited set of data.

This approach aims to be an improvement of the previous approach on the regularity of the blob contour, through the use of a set of measuring functions able to analyse various aspects of the shape of the blob contour.

At month 12 only a few measuring functions have been implemented and tested: distance from the shape centroid, distance from a set of other cardinal points. Thus, the algorithm is not ready for being fully evaluated and compared with the other methods.

References

[Vio02] Viola, P. and Jones, M.: Robust real-time object detection. International Journal of Computer Vision. 2002.

[Des07] Destrero, A. and De Mol, C. and Odone, F. and Verri, A.: A regularized approach to feature selection for face detection. Lecture Notes in Computer Science. Springer 2007

[Dal05] N. Dalal and B. Triggs. Histogram of oriented gradient for human detection. In CVPR, 2005

[Tuz08] Tuzel, O. and Porikli, F. and Meer, P.: Pedestrian detection via classification on riemannian manifolds. IEEE Transactions on Pattern Analysis and Machine Intelligence. 2008.

[Yao08] Yao, J. and Odobez, J.M.: Fast human detection from videos using covariance features. 2008

[Fro99] Frosini P. and Landi C., Size Theory as a Topological Tool for Computer Vision, Pattern Recognition And Image Analysis, 9(4):596–603, 1999.

[Vap98] V.N. Vapnik, *Statistical Learning Theory.* Wiley, 1998.

[Hu62] M-K. Hu. Visual pattern recognition by moment invariants. *IRE Trans. on Information Theory*, IT-8:pp. 179-187, 1962.

4. Comparison among classification methods

This section briefly summarizes the results obtained in the first evaluation phase of the methods under consideration. We recall that the evaluation is based on a quantitative measure based on a set of labelled videos and a qualitative evaluation based on analysing the outputs obtained by the various methods on video streams. See Section 1 for details on the data used.

- **The feature based approach with statistical learning** has a great potential, as testified by the amount of scientific literature devoted to it. A positive element of this approach is that temporal information is not required, although it represents an interesting add-on to improve efficiency. This is not a plus in the environment under consideration where a video sequence is always available (with the exception of still objects classification that could be detected with a purely appearance based approach). The poor efficiency of the methods, and the fact it is not easily scalable to real-time performance make it not appropriate for the task.
- **The shape regularity description** has the important advantage of being very fast to compute. It produced satisfactory results on the set of real videos used in the first evaluation phase, although it shows a limitation in its generalization ability (a small variation of the environment, for instance due to a different organization of the monitoring area, may cause a performance degradation).
- **Size functions** represent a promising compromise between the two above solutions. Preliminary results based on very simple measuring functions convinced the involved RTD performers to invest more in this direction as the field tests carry on. Anyway, if the second approach will confirm its robustness in the test field of the last six months, this third approach will not be tested.

The software beta version is based on shape regularity description.

To give a qualitative flavour of the results it produces we have selected six snapshots extracted from videos analysed by the algorithm: the profile of moving objects is coloured with yellow in case of rigid object (car) and with blue in case non-rigid object (person).

The six snapshots are shown in the next *Figures 3-8*

Figure 3 – A person is walking right under the camera and it is correctly classified

Figure 4 – A car approaches the person and the algorithm distinguishes them

Figure 5 – A motorbike passes ride by a person: the object is rigid and the algorithm correctly classified it

Figure 6 – A car is approaching correctly classified as a rigid object

Figure 7 – A person walks frontally towards the camera and for a while it is recognised as a rigid object

Figure 8 – The person gets near to the camera and the algorithm correctly classifies it as a non-rigid object

5. How software responds to user requirements

We are going to summarize user requirements, constraint and measurable objectives and how the software responds to them.

How software responds to user requirements

- **Detect the presence of people in the work area**
  - The software segments the moving objects through a change detection module with a background updating model based on the variance of each pixel in grayscale
  - The software tracks the moving objects (blob) with the help of the Kalman filter, taking into account the stopped object thanks to a slow background updating
  - The software classifies every blob using a shape regularity description; a configurable number of consecutive frame with a coherent classification makes the method robust.
- **Keep under control the whole terminal to monitor and automatically report any operational irregularity and safety breach, in the first place regarding human safety**
  - Month 18 requirement
- **Issue a signal than can be used to stop all equipment in the area where the human presence was detected (safety) and to alert the control room (security)**
  - When human presence is detected in the off limit area, the algorithm sends an immediate alert to the system management in order to immediately stop the operations of load/unload and to alert security human operators.
- **Let the human operator monitor the area through video cameras**
  - Month 18 requirement

How software meets constraints

- **Work in extreme weather conditions: fog, rain, snow, wind up to 100 km/h, temperatures between -30° and +50°C.**
  - The hardware components will be definitively selected within the month 18
  - Image processing methods to contrast noise due to bad weather has been implemented and successfully tested in the first 12 month testing scenario.
- **Operate in daylight and at night, preferably using its own illumination although artificial lighting can be provided**
  - The first 12 month testing scenario was equipped with day-night video-cameras and was sufficiently illuminated for this purpose. Test have been successfully carried on.
  - The final choice of the hardware components and illumination will be done within the month 18.

How software meets measurable objectives

Following the above mentioned motivations, considering that people intrusion in the plant may be seen as a very rare event (considering the complexity of the anti-intrusion system an estimate of 1 intrusion per month, mainly due to errors of human operators is an upper bound) the following error percentages are acceptable:

- **Maximum error percentage -- people detection in the work area:**
  - **in standard conditions: false alarms 5%, misses 1%**
  - **in extreme conditions: false alarms 8%, misses 3%**
  - **at night: false alarms 8%, misses 3%.**

The tests above described meet the objectives defined by the SME in the User Requirements, even though the number of people in the test environment is much higher (every video include from 1 to 3 people) than the estimated 1 per month; therefore the impact of the error is obviously different.

6. Processing pipeline

The Metrocargo plant will be covered by several video cameras which will monitor all the sensible area: off-limit zones where the mechanical devices work and adjacent areas where unauthorized actions may start.

All the cameras will be connected with the analyzer server, each one identified in the server either as an “off-limit zone” camera or an “adjacent area” camera.

The interaction between cameras will be helpful because it allows to the data management software to send a pre-alert if the movement is in an adjacent area, enabling thus to try to stop the moving object before it enters in the off-limit area.

Moreover in the last six months of the project it will be investigate the opportunity of using the alert of “adjacent area” cameras as a in-system pre-alert for “off-limit zone” camera, helping the vision system to be more confident in the critical areas.

For each camera the software receives as an input a video stream.

The first part of the software (object detection and tracking) follows this pipeline for each frame:

- background update;
- change detection between video frame and updated background;
- blob matching based on Kalman filter;
- Kalman filter update.

The output of this first part of the software is a list of objects detected on the scene of camera, each one with its history from its first appearance to the current video frame.

Together with the video stream, the list of blob for each frame is the input of the second part of the software (object classification); the analysis of the blob contour regularity follows this pipeline for each blob and for each frame:

- Choice of the reference blob (after five observations of the tracked blob)
- from the 6^{th} observation of the blob, computation of the distance of the shape between the current blob and the reference one;
- updating of a running variance of the distance;
- classifier decision if the object is a person on the basis of the overcoming of a configured threshold;
- alert after 3 consecutive coherent decision.

The output is the alert message when a person is detected on the monitored area.

7. User manual

The software prototype is an executable compiled for Linux OS with the following requirements:

- server equipped with CPU Pentium IV 3.0Ghz or higher, RAM 512Mb min, Ethernet 10/100 Mbit, Hard Disk 80 Gb min;
- frame grabber for video PAL acquisition and digitalization;
- Linux OS, kernel 2.6.23.9, Debian etch distribution
- low level libraries distributed with the software

WP Leader IMA has placed its ICSVision Analyzer server in order to run and test the software.

The program takes in input and analyzes a standard YUV420P video stream captured from a CCTV camera.

The configuration of the program is done by an **XML file** like the following:

```
<peopledetection>
  <background interval="100" alpha="0.075" threshold="60" />
  <tracker maxDist="200" min-area="200" maxColorDist=”0.3” />
  <regions>
    <rectangle name="work area">
      <point y="296" x="72" />
      <point y="332" x="416" />
    </rectangle>
    <background width="640" height="480" />
  </regions>
  <shape-analysis min-observation=”5” variance-threshold="0.3" alarm-
        threshold=”3” />
  <rpcclient listener="10.0.0.23:9000"/>
</peopledetection>
```

The background node controls the behaviour of the change detection module used by the algorithm, here follows a brief description of each parameter:

- *interval="100"* the number of frames an object should be static to be included in the background
- *alpha="0.075"* the weight given to new frames in the background updating
- *threshold="60"* threshold used in the difference between current frame and the background to detect moving objects

The tracker node controls the behaviour of the object tracker used by the algorithm, here follows a brief description of each parameter:

- *maxDist="200"* maximum distance (in pixel) allowed for an object in two consecutive frames to be followed by the tracker
- *min-area="200"* minimum area of objects to track
- *maxColorDist="0.3"* maximum distance between colour histograms of objects to track

The regions node in the XML controls the regions of interest in the scene. In this way it is possible to select only some regions of the scene to control for the presence of people.

- *Rectangle points* set the vertexes of rectangles where analysis has to be performed
- *BG width and height* the coordinates of the points are calculated with respect to this size of the frame

The shape-analysis node sets the parameters of the blob contour regularity algorithm, here follows a brief description of each parameter:

- *min-observation="5"* the minimum number of observation of the blob needed to set the reference shape and start the blob contour regularity process
- *variance-threshold="0.3"* threshold in the variance of the shape of a tracked object.
- *alarm-threshold="3"* if the variance in the shape goes above the threshold for this number of times, an alarm is issued by the program.

The ***rpcclient*** node sets the host and port of the XMLRPC server which will receive the alarm.

The alarm is sent through the network using the XML-RPC protocol[4] with the path *event.alarm*

Here follows an example of XML message.

```
<param><value><struct>
  <member>
    <name>type</name>
    <value><string>people</string></value>
  </member>
  <member>
    <name>source</name>
    <value><string>camera0</string></value>
  </member>
  <member>
    <name>event_id</name>
    <value><int>1234</int></value>
  </member>
  <member>
    <name>timestamp</name>
    <value><int>1234567890</int></value>
  </member>
</struct></value></param>
```

In order to better clarify the use of the XML, we have prepared a simple table where we give some examples about how to change XML parameters to configure the software in different scenarios.

Parameter	Test scenario	Slow objects	Big objects
interval	100	250	200
alpha	0.075	0.02	0.03
maxDist	200	50	150
min-area	200	200	800
min-observation	5	10	7
alarm-threshold	3	6	4

[4] http://www.xmlrpc.com/