MVA2005 IAPR Conference on Machine VIsion Applications, May 16-18, 2005 Tsukuba Science City, Japan
10-1

On Road Simultaneous Vehicle Recognition and Localization by Model Based Focused Vision

Noel Trujillo, Roland Chapuis, Frédéric Chausse and Christophe Blanc
LASMEA, UMR 6602 CNRS/Université Blaise Pascal,63177 Aubière, FRANCE
{trujillo,chapuis,chausse,blanc}@lasmea.univ-bpclermont.fr

Abstract

In this paper we present an application of model based focused vision in order to recognize and localize on road vehicles rear part. A good recognition rate and vehicle distance estimations were observed. Some results are presented here.

1 Introduction

In recent years, vehicular safety has taken special attention in the intelligent transportation comunity. To tackle this problem, driver's assistance systems have been developed in order to reduce the number of accidents. For such assistance systems, differents devices for environement perception are used. In particular, object detection is important for obstacle avoidance. For the detection and localization tasks, we need to recognize or to identify the object we want to detect. Our objective is, once the object is recognized, to find the 3D parameters of obstacles observed by road scene images. In the next section we will describe the recognition method and the principle we use to localize the object in the 3D world.

2 Object Recognition

As described in [3], this recognition method is based on a recursive recognition process driven by an object's probabilistic model, which takes into account the relationship between the object's features. A detailed description is given next.

2.1 Model definition

For object representation we can find several approaches in the literature like geometric (line segments, corners, wavelets descriptors, etc), functional-based, multi-scale, graph representations, etc. The proposed methodology allows us to build a model by using the functional approach for shape representation as shown in [1] for the case of road recognition. Notice we can describe an object by using other kind of representations like geometric and multi-scale approaches, resulting in an hybrid representation.
This process uses both structural and appeareance based approaches taking advantage of each one. The model definition is maybe one of the most crucial points for the recognition process. For this method, the model is constituted of N object features supposed to have a gaussian pdf. Each feature $\mathcal{F}_i$ is represented by a vector $\mathbf{p}_i = (o_{i1}, ..., o_{iM_i})^t$ with M_i parameters o_{ij} which can be the coordinates of the feature, color/gray level regions, a spatial localization of a tuned-filter response, etc. This modeling allows to work in the general *feature space* and not only in a geometric space. The vectors $\mathbf{p}_i$'s are grouped into a vector $\mathbf{x}=({\mathbf{p}_1}^t, ..., {\mathbf{p}_N}^t)^t$ which is considered as a multidimensional *r.v.* with normal probability density function and covariance matrix $\mathbf{Cx}$. The couple $(\mathbf{x}, \mathbf{Cx})$ constitutes the *model* of the object.

2.2 Learning stage

Learning phase is a simple off-line training procedure. Its goal is to give an initial value of the model with the objective of limiting the search of features in the feature space. Mainly, our task is to find the vector $\bar{\mathbf{x}}$ which is the mean value of the object features and its covariance matrix $\mathbf{Cx}$ which has the relational knowledge between features. Both can be calculated by simple statistics.
In order to deal with the problem of scale variant objects, a perspective projection of its geometrical parts could be done. This will be explained in detail in the next section.

2.3 Recognition process

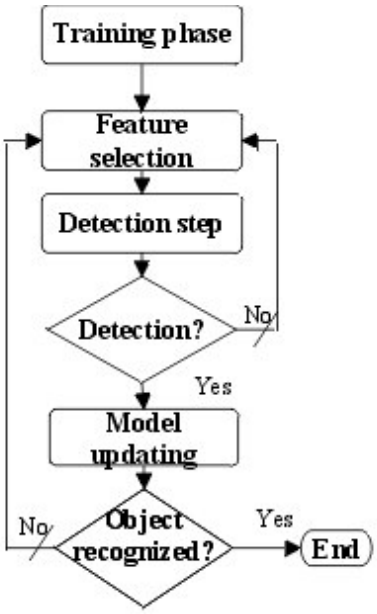
The two main characteristics of the algorithm is (1) the recognition process is achieved in the feature space, (2) it is guided by the model in order to focus a particular subspace in the overall feature space. We are therefore able to deal with a more complete representation of an object avoiding problems when working with large feature spaces and combinatorial problems for feature correspondance.
For a given feature $\mathcal{F}_i$, we define

- a vector $\mathbf{p}_i$ and its covariance matrix $\mathbf{Cp}_i$
- a detection function $f_i(\mathbf{p}_i, \mathbf{Cp}_i)$ associated to feature $\mathcal{F}_i$ able to provide an estimation $\widehat{\mathbf{p}}_i$ of $\mathbf{p}_i$.
- a cost function $\chi_i(\mathbf{x}, \mathbf{Cx})$

2.4 Detection and cost functions

The detection function associated to each feature is a low level image processing operator and it is strongly dependant on application and on the nature of the feature. We have to remark that this detection functions are *parametrable functions*. We mean that the function receives as parameters the model $(\bar{\mathbf{p}}_i, \mathbf{Cp}_i)$, of a current recognition state, in order to focus the detection. The task of these detection functions is to extract an estimation of the parameters o_{ij} in an specific region of interest **(ROI)**, centered on $\bar{\mathbf{p}}_i$ and taking into account the permissible variations given by $\mathbf{Cp}_i$. We have to remark that this **ROI** is defined in the feature space.
The cost function $\chi_i(\mathbf{x}, \mathbf{Cx})$ represents the search cost of the feature $\mathcal{F}_i$ on the recognition process.

Figure 1: Simplified organization chart of the recognition process



2.5 Algorithm's Evolution

Processing all this information could be extremely time consuming if we use a typical rigid model matching algorithm. In order to overcome with this problem, a recursive model-driven procedure is used. Then, the algorithm processes only the information the model needs (a special feature in a certain region in the image for a given resolution, for example). After a successful detection, the recognition process updates the statistical model, by means of a Kalman filter, in order to reduce the permissible variations of the resulting features. The process starts from an initial value $(\mathbf{x}(0), \mathbf{Cx}(0))$ corresponding to the value given from the learning stage. Then for a particular level or fecognition k, the process detects the feature $\mathcal{F}_i$ having the smallest cost calculated with the cost function χ_i. The algorithm is stopped when a certain criterion is reached. Figure 1 shows the simplified procedure chart of the algorithm.

2.6 Recognizing objects at different scales

In order to recognize objects of different sizes in an image, a variation of typical approaches was developped. In the bibliography we can find several approaches dealing with this problem [4] [5], more often, a multi-resolution analysis is used. When working with rigid models, a very fine search over scale must be done, step of $1/4$ is typical for good results (typical 11 scales to analyze) [5]. The inconvenient is that we need to analyze several scales to detect our object and consequently a high processing time is involved. In order to tackle this problem, we have used the typical 2^n resolution analysis. The fact that we did the learning stage using the perspective projection transformation, as a result we have a deformable model and allows us to deal with small variations in scale. Taking advantage of this permissible variations, we only need to do the search in four scales.

3 Application: On road vehicle Recognition and Localization

We have defined a vehicle's model as a set of 18 features $\mathcal{F}_i$ corresponding to the contour vehicle's rear part as shown in figure 2. All the features are composed by a three parameters vector $\mathbf{p}_i = (u_i, v_i, \zeta_i)$, where u_i and v_i are the coordinates of the feature's center relative to the image coordinate system and ζ_i is the direction of the edge information which will be explained later. In order to built a scale invariant model, a perspective projection of the 3D coordinates points of the vehicle was made.

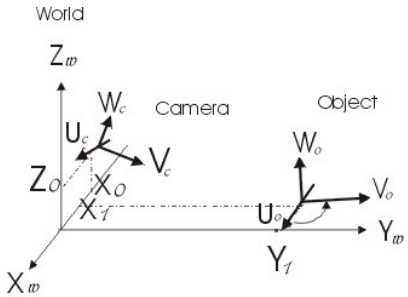
Figure 2: 18 features $\mathcal{F}_i$'s conforming the vehicle's model

3.1 Object modeling in the 3D world

Let us define the three coordinates systems taken into account for this application:

- World coordinate system with axis X_w, Y_w, Z_w
- Object coordinate system with axis U_o,V_o,W_o and rotations angles ϕ, θ.
- Camera coordinate system with axis U_c,V_c,W_c and rotations angles ψ, α

Figure 3: World, object and camera coordinates systems



Our objective is to represent the points of the object into the world coordinate system, and then to represent the 3D points of the world into the image plane by means of perspective projection.
Let us define $\mathbf{p}_w$, $\mathbf{p}_o$ and $\mathbf{p}_c$ points belonging to the world, object and camera coordinates system respectively. The representation of a 3D point of the object to the world coordinate system is carried out by the following rigid transformation

$$\mathbf{p}_w = [\mathbf{R}\phi\mathbf{R}\theta\mathbf{p}_o] + \mathbf{t}_o = [X_w, Y_w, Z_w]^t, \qquad (1)$$

$$\mathbf{p}_o = [U_o, V_o, W_o]^t \; and \; \mathbf{t}_o = [X_o, Y_o, 0]^t$$

Where $\mathbf{R}$ and $\mathbf{t}$ correspond to a rotation matrix and translation vector respectively.
Thereafter, we have to represent a 3D point of the world into the camera coordinate system by means of

$$\mathbf{p}_c = \mathbf{R}\psi\mathbf{R}\alpha[\mathbf{p}_w - \mathbf{t}_c] = [U_c, V_c, W_c]^t \quad (2)$$

with,

$$\mathbf{p}_w = [X_w, Y_w, Z_w]^t, \;\; \mathbf{t}_c = [X_c, 0, Z_c]^t$$

Object representation into the image plane In order to represent the 3D object into the image plane, we use the well known equations for perspective projection.

3.2 Learning Phase

Since we have a normalized vehicle's image database, we are able to extract directly from images the values of U_o and W_o. The 18 points were selected by hand and extracted from a set of 150 vehicle's images of different appeareance. The ζ_i parameter is calculated making an accumulation histogram of the energy, of the phase component, at the output of a quaternion Gabor filter mask [2]. For each point U_{oi} and W_{oi} 3 filter masks are applied, each one corresponding to horizontal, vertical and diagonal orientations. In order to learn the local appearance of the object, SVM's classifiers are used taking information at differents scales of analysis. At this point, the mean vector $\bar{\mathbf{x}}$ and the covariance matrix $\mathbf{Cx}$ can be calculated.
Because we need the couple $\mathbf{m}_i = (u_i, v_i)^t = f(U_{oi}, W_{oi}, X_0, \alpha, \psi, X_1, Y_1, Z_1, \theta, \phi)$. The covariance matrix can be calculated as:

$$\mathbf{Cm}_i = J_f \mathbf{C}l {J_f}^t \quad (3)$$

where J_f is the Jacobian defined as:

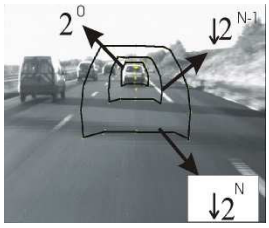
$$J_f = \begin{pmatrix} \frac{\partial u_1}{\partial X_0} & \frac{\partial u_1}{\partial \alpha} \cdots & \frac{\partial u_1}{\partial \phi} \\ \frac{\partial v_1}{\partial X_0} & \frac{\partial u_1}{\partial \alpha} \cdots & \frac{\partial v_1}{\partial \phi} \\ \frac{\partial u_2}{\partial X_0} & \frac{\partial u_2}{\partial \alpha} \cdots & \frac{\partial u_2}{\partial \phi} \\ \vdots & \vdots & \vdots \\ \frac{\partial v_{18}}{\partial X_0} & \frac{\partial u_{18}}{\partial \alpha} \cdots & \frac{\partial v_{18}}{\partial \phi} \end{pmatrix}$$

with $\mathbf{C}l = diag(\sigma^2_{X_0} \sigma^2_\alpha \cdots \sigma^2_\phi)$ because of the terms are un-correlated, where diag is a diagonal matrix. Notice that $\mathbf{p}_i = (\mathbf{m}_i, \zeta_i)$ and $\mathbf{x} = (\mathbf{p}^t_1, ..., \mathbf{p}^t_N)^t$. Having calculated the mean value $\bar{\mathbf{x}}$ and its covariance matrix $\mathbf{Cx}$ of the model, we are able to perform the search process for features in the recognition task.

3.3 Detection and cost functions

The detection functions $f_i(\mathbf{p}_i, \mathbf{Cp}_i)$ are SVM's based detectors. The region in the image to be scanned and the orientations of the low level image operators are given by $(\mathbf{p}_i, \mathbf{Cp}_i)$. An estimated value $\hat{\mathbf{p}}_i$ is returned . The cost functions are defined in function of the sliding window's size and in the scale of analysis. For this application, it is more expensive to search a small feature in a high resolution image than a low resolution overall appeareance feature of the vehicle. Figure 4 shows different initial models when searching vehicles over scale.

Figure 4: Vehicle recognition over scale.



3.4 Vehicle Localization

For the localization purpose, we can deduce the 3D parameters by means of a Kalman filter by using the following equations:

- A measurement equation given by
  $\hat{\mathbf{x}} = f_o(\mathbf{x}_l)$
  $\approx \mathbf{H}\mathbf{x}_l$ with $\mathbf{H} = \frac{\partial f_o}{\partial \mathbf{x}_l}|\mathbf{x}_l = \mathbf{x}_{l_o}$
- and an evolution equation given by
  $\mathbf{x}_l(k+1) = f_e(\mathbf{x}_l, \mathbf{u})$

The filter is fed with $\mathbf{x}(k)$ and $\mathbf{C}(k)$.
Having the estimation of $\hat{\mathbf{x}}$ we can deduce $\mathbf{x}_l$ in order to predict $\mathbf{x}_l(k+1)$.

3.5 Vehicle Tracking

The tracking phase is a very simple procedure which consists of keeping the model of the last state, after a successful recognition, $(\mathbf{x}(p), \mathbf{Cx}(p))$ and to re-define the new permissible variations of the object. In the case of vehicle tracking, the new permissible variations could be given by the possible lateral position of the vehicle and the possible variation in depth. For the next image, the initial model corresponds to $(\mathbf{x}(p), \mathbf{Cx}(p))$ and the process is repeated until the recognition criterion is achieved. We have to note that the possible variations of the model are strongly reduced in relation with the initial model $(\mathbf{x}(0), \mathbf{Cx}(0))$ and thus the recognition process is faster. In our tests, the recognition process time in the tracking stage is between $80 - 150ms$ running the program under Matlab. A velocity vector could be taken into account in order to better predict the trajectory of the vehicle.

Figure 5: Approximated distances of vehicle.

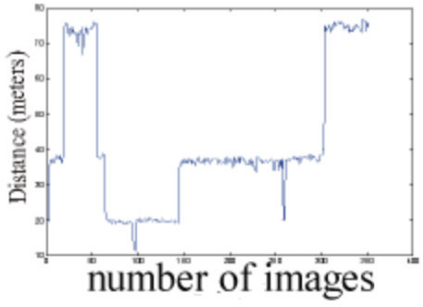
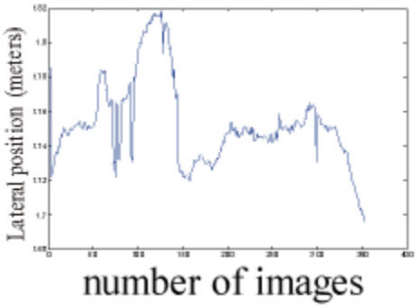


Figure 6: Vehicle's lateral position.



3.6 Improving object recognition by using LASER telemeter data

Having devices that helps to localize objects, we can take advantage of that in order to accelerate the recognition process. For example, without localization information the recognition process must do a search in the image over scale. If we know the 3D localization of the object, it's easy to feed this information, in a natural way, into the system and to focus the attention into a particular part of the image and also in a specific level of resolution. As a result, a faster recognition time is obtained.

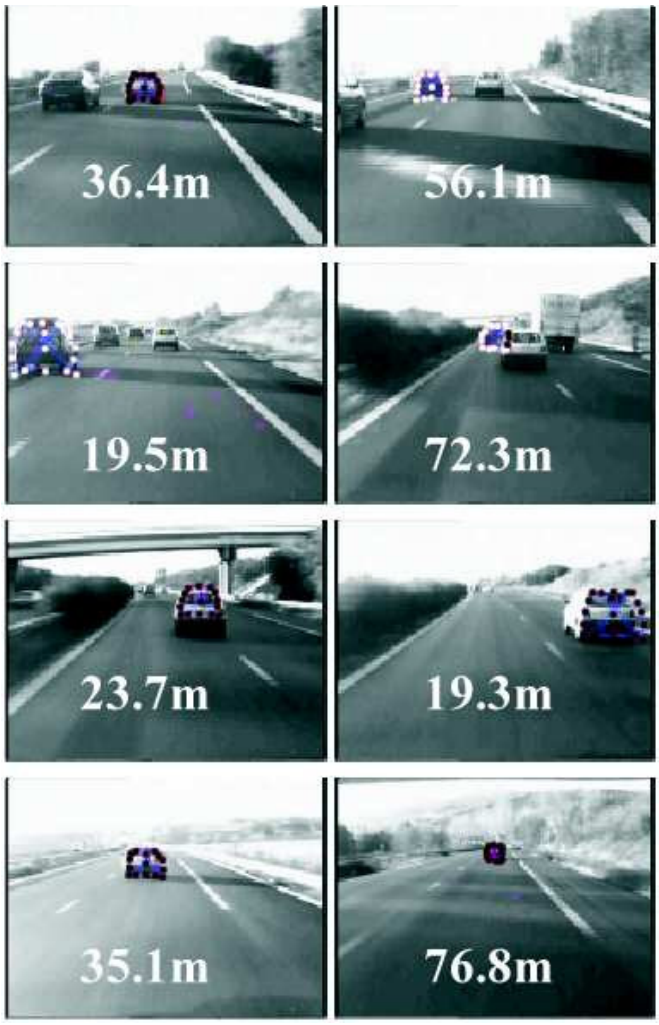
4 Results

Figure 7 shows examples of vehicle recognition and the distances estimated by the present algorithm. The algorithm was tested in a video sequence with more than 500 images. A good recognition rate of 96% was observed. In figure 6 we observe the evolution of the distance aproximated by the algorithm in a video sequence, and the estimated lateral position of the vehicle in figure 5. Due to model defects, many different configurations (mainly in size) can be adapted to a same size vehicle. Obviously this results an error in the estimated distance.

5 Conclusions and Future Work

An image based recognition and localization method was presented. The results show a good recognition rate in the recognition stage and a good approximation of the object's distance in relation with LASER telemeter information. Estimation errors in our image based approach are due mainly in the fact that the recognition system mistakes the recognition due to the simplicity of the choosen vehicle's model. Another problem is that, small errors in the estimated model are translated into large variations in distance.
In the future, an integration of the lidar to our system is necessarily in order to focalize it to a specific direction depending on model's needs. A cost function must be defined in order to calculate the probability of good and bad detection of each primitive. Thereafter, an automatic and quasi-optimal primitive selection could be done in order to make the algorithm converges more fast.

Figure 7: Examples of vehicle recognition and localization.



References

[1] R. Aufrère, R. Chapuis and F. Chausse. A model-driven approach for real-time road recognition. *Machine Vision and Applications*, 2001.

[2] Trujillo N. Bayro-Corrochano, E. and Naranjo M. The role of the quaternion fourier descriptors for preprocessing in neuralcomputing. 2003.

[3] Chapuis R. Chausse F., Trujillo N and Naranjo M. Object recognition by model based focused vision. 2004.

[4] M. Papageorgiou, C. Oren and T. Poggio. A general framework for object detection. *Proc. Int. Conf. Computer Vision*, 1998.

[5] Michael Jones Paul Viola. Rapid object detection using a boosted cascade of simple features. Conference on Computer Vision and Pattern Recognition, 2001.

[6] Takeo Schneiderman, Henry. Kanade. Object detection using the statistics of parts. *International Journal of Computer Vision*, 2002.

[7] Keiji Yanai and Keiji Deguchi. A multi-resolution image understanding system based on multi-agent architecture for high-resolution images. 2001.