TRANSPORT PROBLEMS 2009
PROBLEMY TRANSPORTU Volume 4 Issue 2

Wiesław PAMUŁA
Department of IT Systems in Transport, Faculty of Transport,
Silesian University of Technology
Krasińskiego St. 8, 40-019 Katowice, Poland
Corresponding author. E-mail: wieslaw.pamula@polsl.pl

OBJECT CLASSIFICATION METHODS FOR APPLICATION IN FPGA BASED VEHICLE VIDEO DETECTOR

Summary. The paper presents a discussion of properties of object classification methods utilized in processing video streams from a camera. Methods based on feature extraction, model fitting and invariant determination are evaluated. Petri nets are used for modelling the processing flow. Data objects and transitions are defined which are suitable for efficient implementation in FPGA circuits. Processing characteristics and problems of the implementations are shown. An invariant based method is assessed as most suitable for application in a vehicle video detector.

METODY KLASYFIKACJI OBIEKTÓW DLA ZASTOSOWANIA W WIDEO-DETEKTORZE POJAZDÓW OPARTYM NA FPGA

Streszczenie. Artykuł przedstawia dyskusję własności metod klasyfikacji obiektów stosowanych w przetwarzaniu strumieni wideo z kamery. Omówione są metody oparte na wydzielaniu cech, dopasowaniu do modeli i wyliczaniu niezmienników. Zastosowano sieci Petri do zamodelowania przetwarzania. Zdefiniowano obiekty danych i przejścia pozwalające na sprawną implementację z użyciem układów FPGA. Przedstawiono cechy i trudności w implementacji metod. Oceniono metodę opartą na niezmiennikach, jako najbardziej przydatną dla zastosowania w wideo detektorze pojazdów.

1. INTRODUCTION

The crucial task of ITS is, gathering information on road traffic flow. The gathered data is used for traffic control, scheduling of traffic diversions and planning changes in organization of transport systems. Reliable information means accurate and exhaustive estimation of traffic parameters. These parameters include not only simple data on the number of moving vehicles but also parameters describing the dynamics of movement. One of the important vehicle characteristics, related to movement dynamics, is the size and type of vehicles [1].

The knowledge of the size and type, in short class vehicles contributes to a more precise traffic flow model formulation. Additionally discriminating objects, other than vehicles is also of much interest, specially for ensuring safety in interaction between for instance: pedestrians and vehicles, in traffic situations.

Road embedded vehicle detectors provide signals usually for determining the presence of objects above detection areas. Equipped with proprietary hardware for analyzing signal changes in time, so

called vehicle signatures, may function as classifiers [2]. Such solutions require considerable capital investment for the deployment on existing road infrastructure and are cumbersome in maintenance especially on roads with heavy traffic.

Vision based systems offer a flexible alternative to such devices. Easy to install but as yet difficult to tune for achieving reliable measurement requirements. Known systems relay on utilizing presence function evaluation for a set of detection fields covering object sizes for discrimination [3,4].

The problem of vehicle classification requires a consideration on subjects such as: definition of vehicle classes, aspects of world scene to camera image projection, complexity of discrimination algorithms, processing resources for implementing devised actions.

2. VEHICLE CLASSIFICATION

The main determinants for distinguishing classes are physical dimensions of the vehicles. This is not adequate for estimating traffic flow dynamics. Physical size presents only the area a vehicle occupies on a road while for assessing the character of movement one needs knowledge on the way the vehicle potentially moves.

It is desirable to distinguish vehicles of different acceleration capabilities and different average moving speed. For instance differentiating TIR lorries from busses, both are of similar sizes but move differently. Lorry drivers minimize the number of manoeuvres trying to keep a constant travelling speed while bus drivers make frequent stops and starts. Such entirely different moving manner should be accounted for in estimating the traffic dynamics.

2.1. Vehicle classes

Classifying vehicles for estimating traffic flow dynamics is also dependent on the model of traffic which is utilized for carrying out control and planning schemes in ITS networks [5]. Number of classes is derived on the basis of required accuracy of modelling. It is usually between 3 and 8 the most common being 4.

Considering a rough approximation of traffic flow model based on cellular automata four vehicle classes is adequate [6].

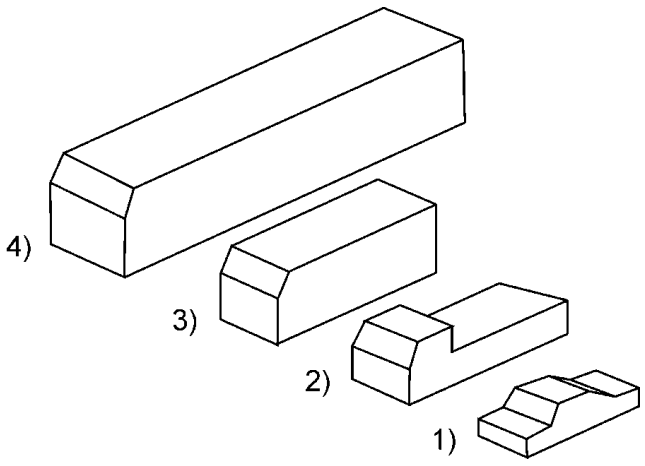

Fig. 1. Vehicle classes – 3D models: 1) cars, 2) trucks-d, 3) trucks-v, 4) long vehicles
Rys. 1. Klasy pojazdów – modele 3D: 1) samochody, 2) wywrotki, 3) ciężarówki, 4) długie pojazdy

The most important class is the class of cars being an average substitute of the largest range of vehicles used for personal travel. This is the most diversified group of objects including slow city cars, on one pole of movement agility through company cars, family saloons and fast sports cars on the other pole.

Other important group of vehicles constitutes two classes of trucks van shaped and dumpers. Although these move in a similar manner and are of comparable size they have quite different shapes

and so are usually separated. The largest vehicles permitted on public roads are first of all long and so appropriately constitute the class of long vehicles. Buses, TIR lorries, trams belong to this class.

Fig.1. presents an overview of defined vehicle classes. These basic classes are characterized by average physical dimensions and rough object shape. Further classification can be performed to enhance the accuracy of estimating traffic parameters [7].

Size proportions in the figure are retained so it illustrates the range of size differences between the smallest and largest vehicle which will mount a challenge for automatic classification designs.

2.2. Camera field of view

Traffic is monitored with CCTV cameras working in the most popular TV standard PAL. Cameras are situated above or beside traffic lanes. This leads to a distorted view of travelling vehicles. Vehicle sizes are not preserved as they proceed along observed traffic lanes. To determine a space transform system for proper classification of objects a camera model is utilized. Using this model the field of view of the camera is calibrated to obtain the correct vehicle size estimation.

Fig. 2 shows the position of the camera relative to the traffic lane. A camera is assumed located at a height h above the ground and a perpendicular distance d from the edge of the lane. It is directed at an angle β with respect to the road axis and tilted α to the ground.

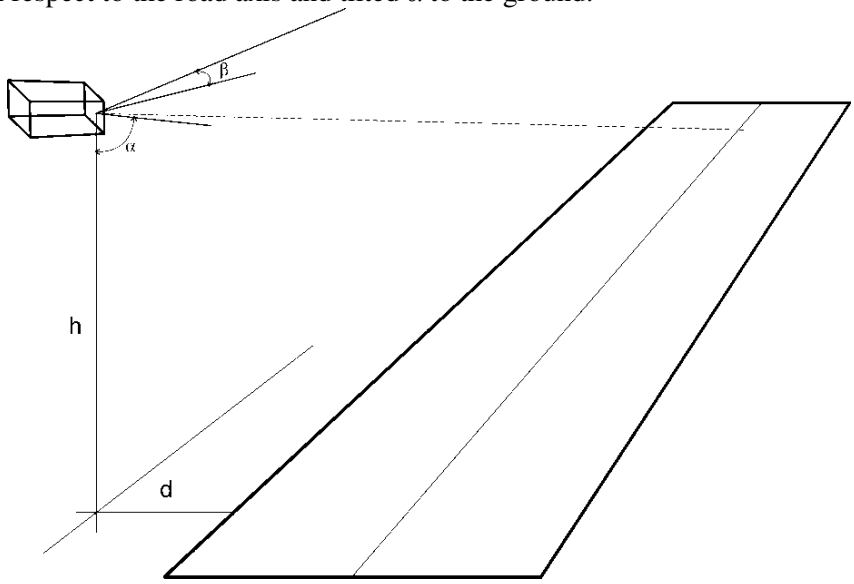

Fig. 2. Camera field of view
Rys. 2. Pole widzenia kamery

The projection of a point in the coordinate system of moving vehicles onto an image plane of the sensor of the camera is additionally determined by hardware factors such as focal length of used optics, sampling rate, aspect ratio of the integrated light sensor [8]. These factors in many real world video monitoring systems are poorly known and sometimes unintentionally are changed because maintenance work is done without competent supervision.

The camera view model can be efficiently represented by the projection matrix **M** using a homogeneous coordinate system in which the space point $\mathbf{P_W}\ [X,Y,Z,1]^T$ and the projected image point $\mathbf{P_I}\ [x,y,1]^T$ are related:

$$\boldsymbol{P_I = MP_W} \tag{1}$$

Decomposing **M** into a hardware factors matrix **H** and a relative camera position matrix **C**, shows the transform details:

$$\boldsymbol{M = HC} \tag{2}$$

where:

$$\boldsymbol{H} = \begin{bmatrix} fSx & 0 & p_x \\ 0 & fSy & p_y \\ 0 & 0 & 1 \end{bmatrix} \qquad \boldsymbol{C} = \begin{bmatrix} \boldsymbol{r}_1 & t_x \\ \boldsymbol{r}_2 & t_y \\ \boldsymbol{r}_3 & t_z \end{bmatrix} \tag{3}$$

and f is the focal length, Sx and Sy are the scaling factors, p_x and p_y are the offset parameters. Vectors $\boldsymbol{r}_i$=(r_{i1}, r_{i2} ,r_{i3}) represent the components of camera rotation and are functions of α, β angles while t_x, t_y , and t_z are the translation parameters related to h and d.

2.3. Processing resources

The basic building block of a FPGA is called by different manufacturers Configurable Logic Block, Logic Element or Slice. It consists of a 4 to 6 input LUT based function generator and storage configured as flip-flops or latches. LUTs are designed to function both as logic and memory elements. Blocks contain additional logic, for efficient implementation of carry chains and to enable joint functioning of neighbouring LUTs [9, 10, 11].

Logic blocks do not provide pure arithmetic functions.

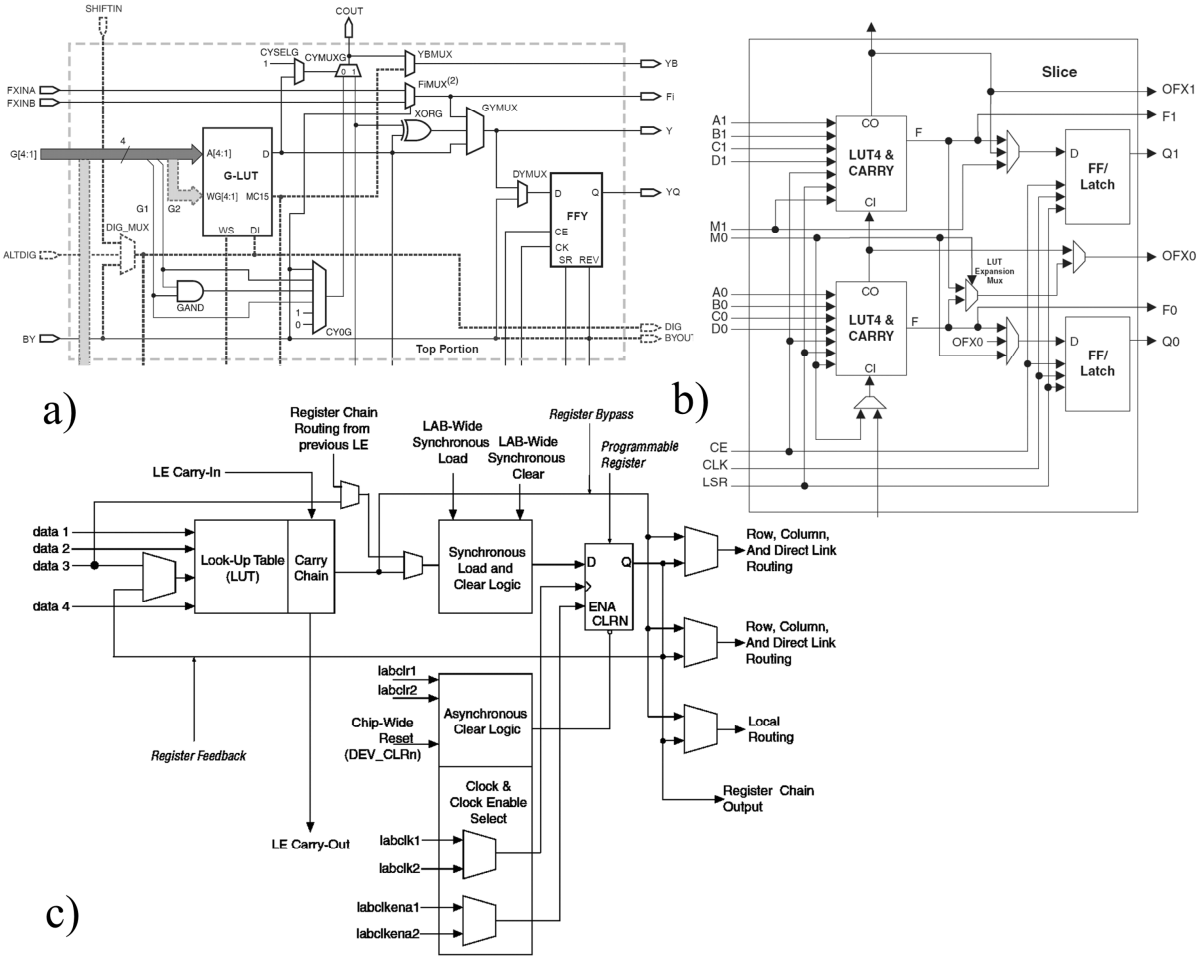

Fig. 3. FPGA logic resources: a) Xilnx ¼ of CLB, b) Lattice slice, c) Altera LE
Rys. 3. Organizacja bloków logicznych w FPGA: a) Xilinx ¼ CLB, b) Lattice slice, c) Altera LE

FPGAs contain complex configurable interconnect structures consisting of routing lines of different speeds enabling the wiring of processing components. Routing components is the most

demanding task in designing a processing structure. Manufacturers provide proprietary routing tools which require laborious tuning to attain desirable performance especially when high processing speeds are envisaged. A way to ease the tuning work is to optimize exchange of data between processing components.

A different approach should also be undertaken to the task of decomposing the processing algorithm. Abundant logic resources must compensate scarce memory, although some FPGAs have embedded block memories they are insufficient to store data of the size of an image frame.

Another distinctive feature is that controlling algorithm steps transitions requires a construction completely different from a programme with which it is usually associated. FPGA content is defined in VHDL. Processing algorithms prepared using high level programming languages like C may be almost directly converted to VHDL code [12]. Almost means that the programme must mimic a hardware design otherwise the result is unpredictable.

Widely used control structure is a finite state machine which extorts extra effort when devising algorithm steps.

3. CLASSIFICATION METHODS

The aim of classification is to match an object to one from a set of templates. A template may be defined in a number of ways: by a set of features, using a model, with a transform invariant. Matching is performed by analyzing the distance of the object description to the template. The distance measurement is defined in the appropriate feature, model or transform space.

Considering the implementation in a FPGA device, classification methods have to be decomposed into operations acting on data structures. Data structures define the complexity of routing of components performing processing tasks. Simple structures contribute to efficient processing and allow for easy debugging of designed architectures.

Data driven algorithms definition facilitate conversion of processing tasks into components embedded in FPGA.

3.1. Feature based classification

Successful classification requires a careful choice of features characterizing objects [13]. Objects moving in the field of view of the camera change their dimensions and appearance. A correct classification must use movement invariant features or somewhat limit the area of movement to minimize distortion.

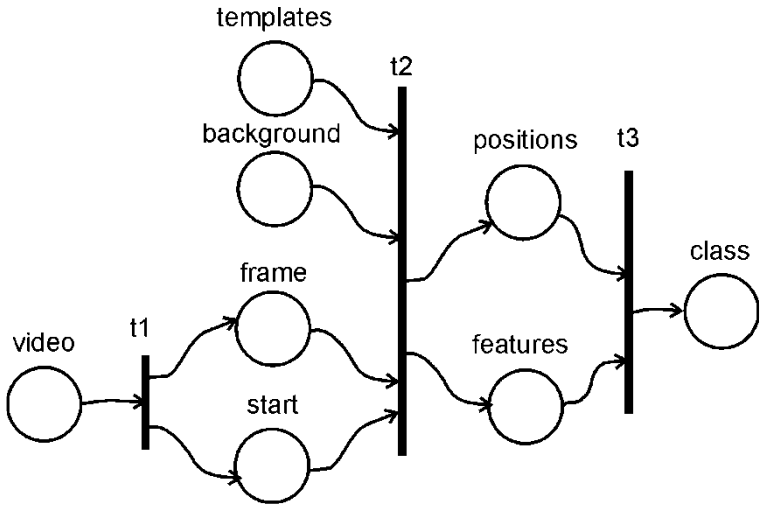

Fig. 4. Feature based classification
Rys. 4. Klasyfikacja na podstawie cech obiektów

Features such as corners, edge segments, patches of uniform pixel values and fork shapes are utilized for characterizing objects. Classes are defined by sets of relative positions of chosen features. A measure of matching – distance for the prepared set is devised which is optimized for implementation incorporating at most logical operations.

Fig. 4 presents an example of image processing for deriving feature based classification of objects. Petri net is used for decomposing the processing flow [14]. Camera delivers a stream of video represented by the data object *video*. Transition *t1* extracts the signal *start* and *frame* which combined with objects *background* and *templates* are processed in transition *t2* to form objects *feature* and *position.* Following transition *t3,* performs distance calculations and comparisons giving as result the object *class*.

Features are detected utilizing a matching procedure based on comparing template points with object points derived by subtracting background from current frame contents. Transition *t2,* a simple comparison, may be substituted by for instance a corner detection or geometric moments calculation procedure. In such cases the object *templates* will represent appropriate template spaces of corners or moments required for classification.

Background is modelled using a simplified Mixture Gauss model [15] This is not expanded to keep the diagram clear. It can be noted that transitions work on different sized data objects which gives a paramount advantage over conventional processor based implementations.

3.2. Model fitting

When the field of view of the camera covers a large area and object features are hard to discern due to low camera resolution applying model fitting delivers a solution for classification. This approach is especially useful when tracking objects with occlusions in highly cluttered environment.

Model fitting is based on matching wire models of objects [16,17]. The models represent average object contours. Matching consists of finding models that coincide with object edges or segment approximations of object contours. The wire models are prepared off-line by projecting 3D polyhedrals as in fig.1 onto the image plane using eq.1 when the camera parameters are known or utilizing a field of view calibration program [18].

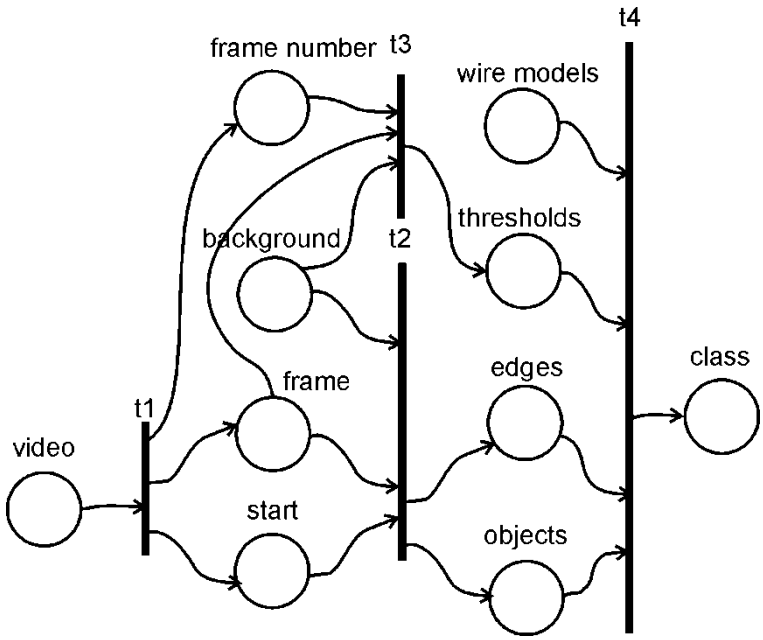

Fig. 5. Model fitting
Rys. 5. Dopasowanie modeli

The resulting models are cut into segments, hidden segments are eliminated. A set of segments for each class is defined. The segments are additionally characterized by weights indicating their significance in total length of segments of the model.

The matching process should also take into account the contents of interior of the model to avoid erroneous matching to road features instead of vehicles.

Transition *t1* as in previous section prepares current image data. Model matching requires more sophisticated data preparation. It involves edge detection which is dependent on accurate evaluation of whole frame characteristics to determine discrimination thresholds. Therefore two transitions are defined to separate frame and current processing tasks. Transition *t2* derives current object data while *t3* calculates image data characteristics over a number of frames. Although different these transitions are shown at the same level because it is possible to perform them concurrently.

Specific to FPGA, matching may be done on all classes simultaneously and this is represented by transition *t4*.

3.3. Invariant evaluation

The imaging process of camera may be modelled by projective transform as shown in section 2.2. It does not preserve distances nor ratios of distances, a measure that is preserved is known as cross-ratio [19]. Cross ratio is defined using points, lines or planes. In the case of image processing a points definition is most convenient as pixel coordinates are the basic elements of an image. It is a ratio of ratios of distances between four collinear points:

$$C_r(p_1, p_2, p_3, p_4) = \frac{d_{13} d_{24}}{d_{14} d_{23}} \tag{4}$$

where: p_i - collinear points, d_{ij} - distances between point p_i and p_j.

Using this measure an invariant is derived suitable for characterizing rigid polyhedral objects moving or variously placed in the field of view of a camera [20, 21].

$$I = \frac{T_{123} T_{456}}{T_{124} T_{356}} \tag{5}$$

where: T_{ijk} are areas of triangles with vertices at points p_i, p_j, p_k. There are six points placed on two planes intersecting at p_3 and p_4 as illustrated on fig. 6.

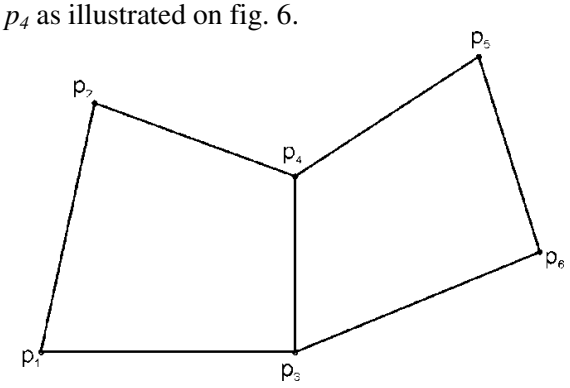

Fig. 6. Reference points for calculating moving object invariant
Rys. 6. Punkty dla wyliczania niezmiennika poruszającego się obiektu

Expressing areas in terms of vertex coordinates on the image plane the invariant:

$$I = \frac{\begin{vmatrix} \boldsymbol{p}_1 & 1 \\ \boldsymbol{p}_2 & 1 \\ \boldsymbol{p}_3 & 1 \end{vmatrix} \begin{vmatrix} \boldsymbol{p}_4 & 1 \\ \boldsymbol{p}_5 & 1 \\ \boldsymbol{p}_6 & 1 \end{vmatrix}}{\begin{vmatrix} \boldsymbol{p}_1 & 1 \\ \boldsymbol{p}_2 & 1 \\ \boldsymbol{p}_4 & 1 \end{vmatrix} \begin{vmatrix} \boldsymbol{p}_3 & 1 \\ \boldsymbol{p}_5 & 1 \\ \boldsymbol{p}_6 & 1 \end{vmatrix}} \tag{6}$$

An object class is defined by a set o vertices selected to cover characteristic points which distinguish the class objects. These points must lie on two intersecting planes.

Movement invariants for objects are sensitive to vertex coordinate extraction errors. It is vital to diminish transform and calculation errors. Eq. 8 presents an estimation of invariant sensitivity to coordinate determination errors. It is the sum of sensitivities of estimating the triangle areas S_{ijk}:

$$S_{ijk} = \frac{1}{T}\sum_{i=1}^{6}\frac{\partial T}{\partial c_i}\frac{dc_i}{c_i} = \sum_{i=1}^{6}\frac{1}{1+\Delta_i}\frac{dc_i}{c_i} \tag{7}$$

where:

$$\Delta_1 = \frac{c_3(c_6-c_2)+c_5(c_2-c_4)}{c_1(c_4-c_6)}, \Delta_2 = \frac{c_1(c_4-c_6)+c_5(c_2-c_4)}{c_3(c_6-c_2)}, \Delta_3 = \frac{c_1(c_4-c_6)+c_3(c_6-c_2)}{c_5(c_2-c_4)}$$

$$\Delta_4 = \frac{c_4(c_5-c_1)+c_6(c_1-c_3)}{c_2(c_3-c_5)}, \Delta_5 = \frac{c_2(c_3-c_5)+c_6(c_1-c_3)}{c_4(c_5-c_1)}, \Delta_6 = \frac{c_2(c_3-c_5)+c_4(c_5-c_1)}{c_6(c_1-c_3)}$$

and *c* are coordinate values of vertex points.

$$S_I = S_{123} + S_{456} + S_{124} + S_{356} \tag{8}$$

Rough estimate of Δ_i value, done for a series of video traffic images, is about 2. The value of triangle area error therefore is 2 times the error of vertex coordinate extraction.

The resultant invariant sensitivity reaches a value about 8 which means that correct separation of classes requires invariant values differing more than 8 times the error of vertex coordinate determination. Using standard resolution CCTV cameras this amounts to about 8%.

It can be noted that a way to diminish errors is to define reference vertex points which lies as far apart as possible on the examined object. Fig. 7 shows preferable vertex points on vehicle models. These are roof top, front window vertices, head light centres. In order to enhance the robustness of classification a model is defined by two to four sets of reference points.

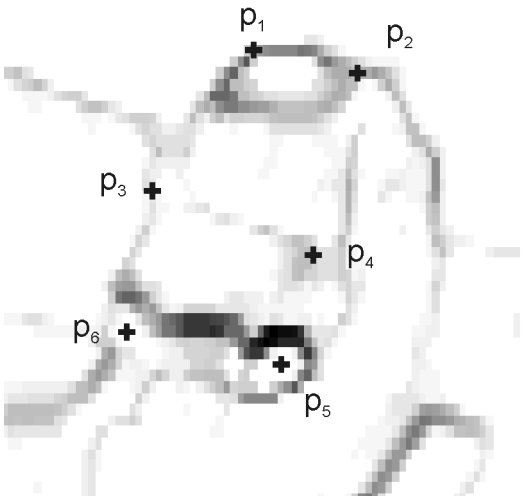

Fig. 7. Vertex points on vehicle model
Rys. 7. Punkty wierzchołków dla modelu pojazdu

Fig. 8. presents a design of the processing flow of invariant based classification. Similarly as in model matching the transitions *t1* and *t3* prepare objects *start, frame* and *thresholds*. Transition *t2* derives a slightly different data entity called *objects* which is a set o characteristic pixel coordinates for objects. Transition *t4* extracts object vertex coordinates and calculates invariants. In the following step *t5* these are compared with a set of class invariant values prepared off-line and the classes of the objects are determined. Again the specifics of FPGA allows for simultaneous processing of a number of objects detected in the field of view of the camera.

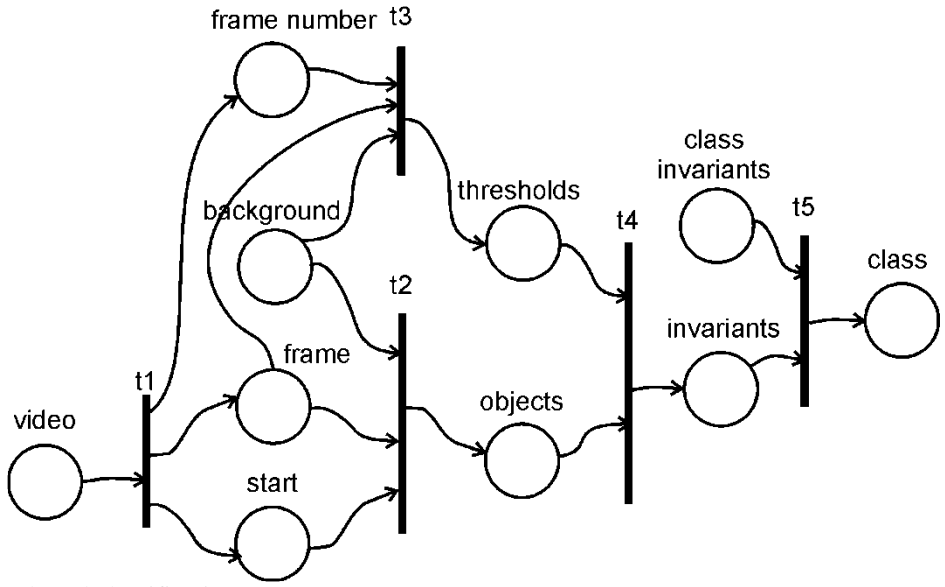

Fig. 8. Invariant based classification
Rys. 8. Wyznaczanie klas na podstawie niezmienników

The off-line preparation of invariant values does not require the knowledge of camera parameters. 3D line drawings of models are adequate for estimating invariant values.

4. CONCLUSIONS

The design of a video vehicle classifier requires special care in the area of user interface. A user interface must be kept as simple as possible, devoid of any intricate configuration procedures. Taking this demand into consideration classification based on features or models may be hard to be accepted by users. Feature based classification gives satisfactory results when the camera view is small or the vehicles are large and during movement change little their appearance. It may be difficult to satisfy these conditions without special camera placement or adapting camera optics.

Model based classification requires calibration of the field of view of the camera. This process may be aided with 3D modeling software but as such requires an experienced user to perform.

Invariant evaluation based classification is less demanding for camera placement and does not require calibration of the field of view. Processing requirements of all the methods are similar. Analysis of the diagrams shows that there are up to five transitions to achieve a classification.

Currently produced FPGA devices have logic which works with clocks of 100 to above 300MHz. The complexity of transition operations prohibits one cycle implementations. Judging on design experience gained during FPGA vehicle detector implementation these transitions will require up to 4 clock cycles [18].

The main source of time loss is however inefficient frame based memory access. Streamlining data exchange is the key to successful algorithm implementation.

Pixel by pixel processing is an alternative solution. At first it may appear infeasible.

A modification of net diagrams by redefining data entities is adequate to model this alternative solution. New entities refer to single pixel data and associated auxiliary variables. The scope of transitions changes.

Video data is usually acquired using a standard video decoder IC which provides data synchronously with a 27 MHz clock. Data is coded in compliance with ITU-R BT 656 standard. Luminance is outputted every second clock cycle and this is the time interval in which all transitions must fit in. The pixel processing cycle is therefore 74ns long. Utilization of catching memory operations and logic circuits for conducting data operations should meet this timing constraint.

Preliminary implementations confirm the usefulness of applying invariant evaluation for classifying vehicles. Work is required to refine the extraction of object vertices. Currently used method is not accurate enough for small objects moving far away from the centre of view of the camera.

5. ACKNOWLEDGMENTS

This research is a part of a project financed by the Faculty of Transport in Silesian Technical University ref. BW515/RT6/2008.

References

1. Datka S., Suchorzewski W., Tracz M.: *Inżynieria ruchu*, Warszawa WKŁ, 1999.
2. Oh C., Ritchie S.G..: *Recognizing vehicle classification information from blade sensor signature*. Pattern Recognition Letters vol. 28 , 2007, pp. 1041–1049.
3. *Autoscope terra*. Data sheet. Image Sensing Systems USA, 2008.
4. *Traficam vehicle presence sensor*. Traficon N.V. Belgium, 2008.
5. Bretherton R.D., Bodger M., Baber N.: *SCOOT – Managing Congestion Communications and Control.* Proceedings of ITS World Congress, San Francisco, 2005.
6. Maerivoet S., De Moor B.: *Cellular automata models of road traffic*, Physics Reports 419, 2005, pp.1- 64.
7. Song B.S., Lee K.M., Lee S.U., Yung I.D.: *3D target recognition based on projective invariant relationships*, Journal of Visual Communications and Image Representation vol.14, 2003, pp 1-21.
8. *FPGA Features and Design*. Xilinx Inc. San Jose, CA USA, 2007.
9. *Cyclone III Device Handbook*. Altera Co. San Jose, CA USA, 2008.
10. *LatticeXP Family Handbook*. Lattice Semiconductor Co., 2007.
11. *Handel-C Language Reference Manual*. Agility Design Solutions Inc., 2007.
12. Sun Z., Bebis G., Miller R.: *Object detection using feature subset selection*. Pattern Recognition vol. 37, 2004 pp. 2165 – 2176.
13. Murata T.: Petri Nets: *Properties, Analysis and Applications*, Proceedings of the IEEE, 1989, vol.77,pp. 541-580.
14. Stauffer C., Grimson W.E.I.: *Adaptive background mixture models for real time tracking*, Proc CVPR, vol.2, 1999, pp 246-252.
15. Polat E., Yeasin M., Sharma R.: *A 2D/3D model-based object tracking framework*, Pattern Recognition vol.36, 2003, pp. 2127 – 2141.
16. Shina J., Kima S., Paika J., Abidid B., Abidi M.: *Optical flow-based real-time object tracking using non-prior training active feature mode*l, Real-Time Imaging vol. 11, 2005, pp. 204–218.
17. Project Report: *Modules of Video Traffic Incidents Detectors ZIR-WD for Road Traffic Control and Surveillance.* WKP-1/1.4.1/1/2005/14/14/231/2005, Katowice, Poland, 2007.
18. Mundy J.L., Zisserman A.: *Geometric Invariance in Computer Vision*, Cambridge MA, MIT Press, 1992.
19. Song B.S., Lee K.M., Lee S.U.: *Model-Based Object Recognition Using Geometric Invariants of Points and Lines,* Computer Vision and Image Understanding vol.84, 2001, pp. 361-383.
20. Zhu Y., Seneviratne L.D., Earles S.W.E.: *New algorithm for calculating an invariant of 3D point sets from a single view,* Image and Vision Computing vol.14, 1996, pp. 179-188.

Received 17.11.2008; accepted in revised form 3.06.2009