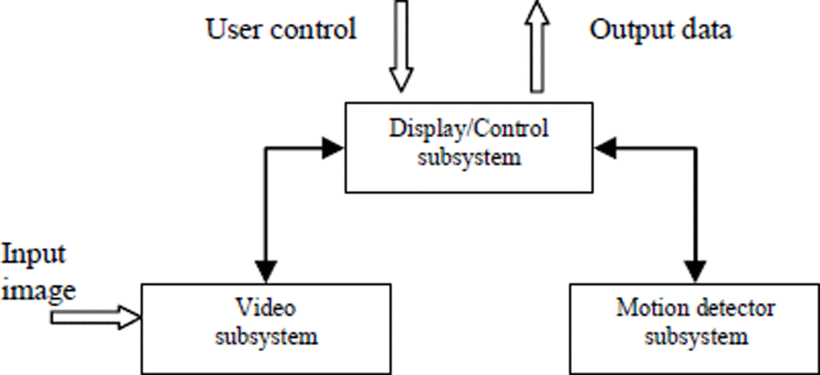
**Traffic Monitoring Using Computer Vision:**

**Summary:**

* **Abstract:**
  + Aim is to build a traffic monitoring system that tracks and counts different vehicles using computer vision and camera
  + The real time process (15-30 FPS) of the video works at daylight
  + Accepts video from a file or camera, marks moving vehicles with rectangle and counts them
  + System contains three sub-systems:
    - Video Sub-System
    - Motion-Detector Sub-System
    - Display/Control Sub-System
  + For max Speed, each Sub-System runs on different threads
  + For accepting Visual Information: DirectShow is used
  + Image Processing: Done partially with DirectCV rapper for OpenCV
  + Motion Detection: Gaussian Mixture with background Segmentation
  + A mask is created to filter out sidewalks and opposite direction traffic flow
  + Mask creation is done by placing points in polygon
  + Polygon is created using Heuristic Method and Genetic Algorithm
  + Labelling Algorithm is used for object detection which uses Decision tree and Union-Find data structure
  + Trying to add shadow removal algorithm
* **Section 1:**
  + **Introduction, System Design:**
    - Online and Offline operation is available
    - Offline operation: Analyzes prerecorded video
    - Online operation: Used recording device connected to computer
    - Inputs are picture frames, and outputs are different frames about the current frame
    - System is divided into three subsystems: Video Sub System, Motion Detector Sub System, Display/Control Sub System



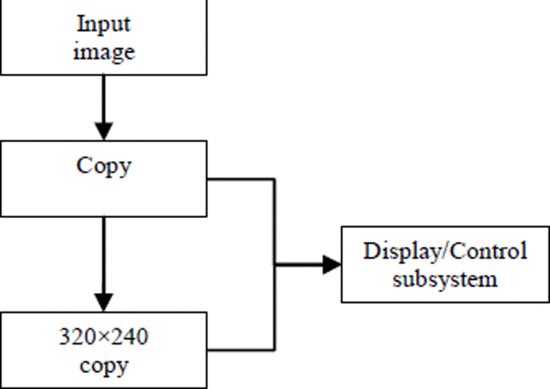
* User interaction using Display/Control Sub System, where user can open desired video file
* This Sub System controls the operation of the system, and tunes individual sub systems to each other
* Video Sub System transfers current frame along with operational data to control sub system
* Motion sub system receives frame, processes it and sends it to control sub system
* Real time operation is a must, each sub systems uses threads to fasten the calculation

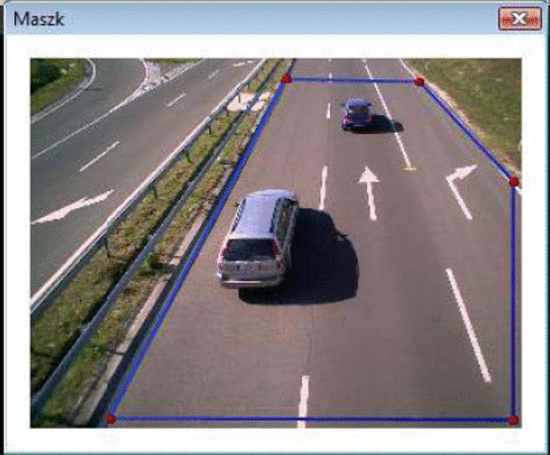
**Video Sub System:**

* Set source of the video stream and start the playback
* During playback sub system creates 2 copies of input frame, One Original Size and One 320x240 size.
* Both are sent to Display/Control subsystem
* Smaller one is used for Display and for motion detection
* In Offline mode we can control the playback i.e., pause, play etc.…

**The Motion Detector Sub System:**

* We can mark the area of interest using a mask
* This system is for motion detection and object detection
* Operation phases
  + Initialization
  + Detection
* **Masking:**
  + There will be a lot of unnecessary elements in the motion detection
  + To remove those, we mark the relevant areas in the image and make our program process only the area within the mask
  + To create the mask, we mark the vertexes of the area and program will create a polygon that matches the vertexes
  + Polygon changes with every vertex added or removed
  + Polygon and Vertex are shown

****



* The mask algorithm creates a sorted set of output points based on the input points
* Solution consists of Sorting the points and Creation of white polygon on a picture with Black Background
* Sorting the points is difficult, where we create a cycle with minimum length of edges, this process is Travelling Salesman Problem or Hamiltonian cycle/Hamiltonian circuit, which is an NP-complete problem
* Suppose we have set of points called “H” with “n” 2 dimensional points (pi) in it.
* We sort the points pi such that distance from each other is the smallest (Distance calculated as Euclidean Distance)
* **n** points have (n-1)! Combinations to try, therefore takes a lot of time
* A greedy algorithm tries to find global optimum while working at each stage of local optimum. Its difficult to find global optimum this way
* Genetic algorithms are optimization algorithms which try to mimic the evolutional progress of the nature to solve problems with no previous knowledge about the target problem, and yet they will eventually find the optimal solution
* Heuristic solutions will give solutions based on previous knowledge, but they may not always give the optimal solution. However, heuristic solutions are faster than genetic, and by merging these two solutions a hybrid method can be creates, that can find the optimal solution with faster speed

**The Algorithm consists of the following techniques:**

* **Initialization:** Generation of M individuals randomly.
* **Natural Selection:** Eliminate p% individuals. The population decreases by M × p/100.
* **Multiplication:** Choose M × p/100 pairs of individuals randomly and produce an offspring from each pair of individuals. The population reverts to the initial population M.
* **Mutation by 2opt:** Choose p% of individuals randomly and improve them by the 2opt method. The individual that has the best fitness value in the population, is always chosen. If the individual is already improved, do nothing, because it cannot be further improved by 2opt.

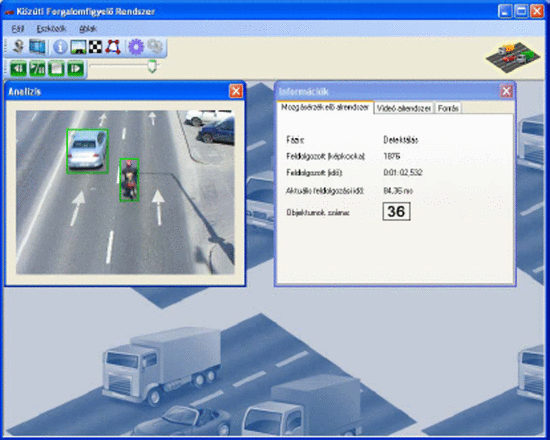
**Motion Detection:**

* Gaussian Mixture Model is used
* For Real Time Segmentation we use Background Subtraction
* **Background Subtraction:** We take a reference frame and subtract it with every new frame and thresholding the result. The result is a binary segmentation which indicates the non-stationary objects
* Time averaged background image as reference image faces many problems and requires training period
* Motion of background images are averaged during training period, while motionless background images are considered permanent foreground objects.
* This method can’t cope with gradual illumination changes in the scene, therefore background model to be constantly re-estimated
* Modelling each background pixel by a mixture of *K* Gaussian distributions (*K* is a small number from 3 to 5).
* Different Gaussians are used to represent different colors. The weight parameters of the mixture represent the time proportions that those colors stay in the scene.
* The background components are determined by assuming that the background contains the *B* highest probable colors.
* The probable background colors are the ones which stay longer on the scene, and which are more static.
* Static single-color objects trend to form tight clusters in the color space while moving ones form wider clusters due to the different angles of reflecting surfaces during the movement.
* To measure this, we introduce a new unit called *fitness value*. To allow the model to adapt to changes in the illumination and run in real-time, an update scheme was applied, which is based on selective updating.
* Every new pixel value is checked against the existing model-components in the order of their fitness value. The first matched model-component will be updated.
* If the process finds no match, a new Gaussian component will be added with the mean at that point and with a large covariance matrix and a small value of weight parameter.

**C. Object detection**

* For the count of vehicles, we detect them in the input frame, this process is called connected components analysis
* The algorithm labels every interconnected pixel group in a binary image with a different label to differentiate different objects.
* Using the differing labels, we can identify the place of an object along with its size, which can even be an input to a pattern recognition algorithm.
* To find an object in an image, Connected Component Algorithm is the base.
* Pattern recognition consumes a lot of time, and there is active research for faster results.
* Input picture is a Binary Matrix, B, for the algorithm. Output matrix is “K”
* “1” in B indicates Object Pixels, “0” Indicates Background Pixels
* To find Connected Pixels, we need to find a neighboring or connected pixel that has the same pixel value as every pixel.
* In 2D we have 4 neighboring and 8 neighboring, we use the latter one
* We use the Fiorio algorithm enhanced with a decision tree and Union-Find data structure.

**Experimental Results:**

****

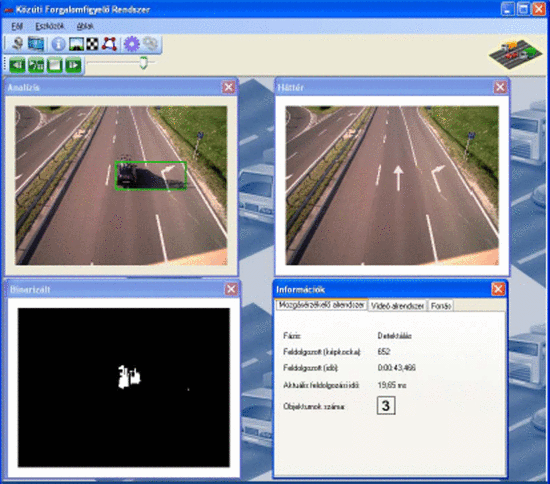
* Type of source: file
* Place: Margaret bridge
* Resolution: 640×480
* Frame rate: 30 frames/second
* Length: 1258 frames
* Type of source: file
* Place: A footbridge over a bypass near Kaposvár
* Resolution: 1280×1024
* Frame rate: 15 frames/second
* Length: 180 frames

The system works perfectly when using these video streams as an input.

We performed additional tests in online mode using a notebook and a webcamera, with the length of 10–20 minutes.

The rate of correct detections was 72% on the average, while the rate of false positives was 8.5%.

These usually occurred at the regeneration of the background or at a sudden change in the illumination.

****

**Reference:** [**https://ieeexplore-ieee-org.ezproxy1.lib.asu.edu/document/4956624**](https://ieeexplore-ieee-org.ezproxy1.lib.asu.edu/document/4956624)