I. Introduction
This project aims to design a video surveillance software solution that can be executed on a single desktop PC running Linux Operating System. The objective of this project is to propose an economical solution for a surveillance package that can recognize faces and keep track of faces over multiple surveillance regions covered by different cameras. It will also backup compressed video recording with time stamp and camera ID for later use. The figure above shows the complete block diagram of the system.
The whole system can be broken down into three subsystems, which will be discussed in subsequent sections, followed by the limitations of this solution.

II. Face Detection System – Haar Cascade Filter
For face detection in a video frame, Haar Cascade Classifier is used. This classifier iterates over the image several times to determine face regions. It is capable of detecting multiple faces in a frame.

Following filter settings were used for face detection:
- Minimum Face size = 48 x 48
- Minimum Hits = 5
- Scale for Increase in ratio = 1.3

Face detection accuracy for this system is 94% (frontal face only), with 6% false negatives and 7% false positives. The computational time for a single frame of size “480x640” is 52 milliseconds.

III. Face Recognition System – Eigenface
A face image size is fixed to 50x50, i.e. a face can be considered to be a point in 2500 dimensional space, where each pixel position represents a dimension. Two faces can be distinguished by their point-to-point distances, but calculating distances in 2500 dimensional space is computationally expensive and it is inaccurate as diminishing a 2500 dimensional value to a single scalar value of distance will also accumulate the noise in each dimension, thus amplifying the noise by the factor of 2500.

Eigenface method uses Principal Component Analysis, to determine mutually orthogonal vectors in directions of maximum separation, called Principal Components (PCs). By using few of these PCs, we can form a subspace and project all face images onto this subspace. We can now calculate distances between two faces in a reduced dimension.

Principal Components are 2500 dimensional vectors and can be visualized by a 50x50 intensity image, as shown in the figure on the right.
This system delivers 93% frontal face recognition accuracy, with 7% false negatives and 5% false positives. Using 50 faces (5 samples per person) in database for comparison and 4 Principal Components, it takes 5-7 milliseconds to recognize a face.

IV. Face Tracking System – Motion Estimation
Motion Estimation Algorithm, generally used in video compression, is a fast search algorithm to determine a translational vector (also called the Motion Vector), which gives least error (e.g. Mean Absolute Distance) from the previous frame to the current frame for a block of image. This technique is applied for face tracking in this project. The Motion Estimation Algorithm used for this system is New Three Step Search (NTSS), which is center biased in its search for matching block.
In comparison to the Full Search algorithm, for salesman test video, NTSS algorithm provides 10.94 speedup ratio, 0.990 probability of catching true motion and 0.044 average distance to true motion.

For the purpose of face tracking, NTSS algorithm was modified to fit the requirements of this project.
Some of the modifications are:
- Variable search window size.
- Checking points at every fourth pixel to increase the search area from 47 pixels to 128 pixels in both horizontal and vertical directions.

V. Limitations
Some of the limitations of the system are:
- Face recognition is highly sensitive to illumination changes.
- System accuracy is very sensitive to changes in face orientations.
- Motion Estimation implementation only allows for discrete movement of 4 pixels in vertical and horizontal directions.
- Cannot track fast moving faces, i.e. more than 28 pixels per frame.
- Using camera with the feature of auto-blurring of frames will also reduce the accuracy of the system.