# CHAPTER-3

# FOREGROUND SEGMENTATION

Foreground segmentation is a task of detecting the salient objects and separating it from rest of the image and this saliency could be due to its motion or, any other feature. So, in simpler sense foreground segmentation could be performed to detect moving objects or to separate parts of objects, such as isolating just the face or the hands of a person in a video sequence. In this chapter Section 3.1 gives an overview of the concept of foreground segmentation. Section3.2 introduces the background modeling. In Section 3.3 different tensor based foreground segmentation techniques are discussed, which is our area of work.

## 3.1 Overview

The need of every tracking problem is to detect the object in the first frame or initially when object appears in the video. Temporal information is being used in many famous methods to reduce the number of false detections which is usually in the form of frame differencing that highlights the changing regions in consecutive frames. As soon as the target objects are segmented, the tracker's task begins to perform object correspondence in the consecutive frames in order to obtain their tracks i.e. the path followed by objects in the subsequent frames. Detection of objects is performed in many ways, one of the most attractive method is the background modeling where firstly a background model is created and then the deviations of each incoming frame is obtained from this model.

If there is any significant change in an image region from the background model it will indicate the presence of a moving object and such pixels are suitably marked for further processing. A connected component algorithm is generally applied in this regard in order to obtain connected regions corresponding to the objects. This process is well known as the background subtraction or, also referred to as foreground segmentation. Foreground segmentation is a fundamental processing stage for vision systems which monitor real-world activity. Foreground segmentation is used in numerous practical applications like surveillance, optical motion capture and multimedia for instance, in video security the background of the scene viewed by the camera most of the time remains same for most of the time and there is an interest in the changes that are taking place in it like when people or vehicles enter the scene. Thus, this background information can be removed and thus saving the computational power by

ignoring the endless hours when nothing is changing. In a stepwise manner background is separated by firstly obtaining a background model then a proper initialization of the background is done. Background is maintained over time, in order to adapt to various dynamic changes and finally the foreground detection is performed. Figure 3.1 illustrates the above discussed steps.

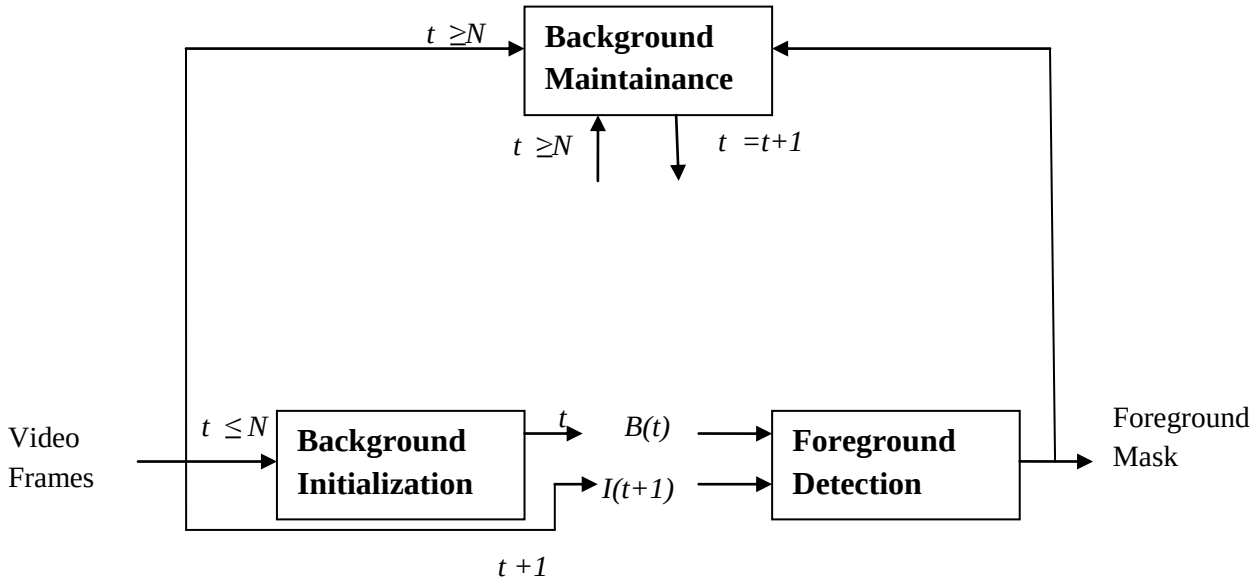


**Figure 3.1 A Pipelined view of Background Subtraction [44]**

## 3.2 Background Subtraction

Background subtraction is the process in which foreground objects are extracted from a particular scene. A foreground object can be described as an object of interest and helps in reducing the data to be processed and at the same time provide important information about the task under the consideration. There are various constraints in the analysis of the natural scenes like changes in the illumination , shadows, poor quality image source, camera jitter, camera automatic adjustments, time of the day, light switch, camouflage, foreground aperture, moved background objects, inserted background objects, multimodal background, waking foreground object and sleeping foreground . In this regard background modeling has sought a lot of attention as these situations possess different spatial and temporal properties. The major difficulties arise from the illumination changes and dynamic backgrounds that is a very common phenomenon and can appear in indoor as well as in outdoor scenes. Figure 3.2 shows an

indoor scene with gradual illumination change. False detections produced in several parts of the foreground mask are shown in the Fig. 3.2(d).

a) Dark b) Middle c) Clear d) Foreground mask

**Figure 3.2 Gradual Illumination Changes in the scene [44]**

a) Light-on b) Light –off c) Foreground mask

**Figure 3.3 Sudden Illumination Changes [44]**

Figure 3.3 shows the case of a sudden illumination change due to a light source changing from on to off. All the pixels are affected by such change and thus, a big amount of false detections is generated as shown in Fig. 3.3(c).Dynamic backgrounds occurs in outdoor scenes due to the trees swaying because of winds; tree leaves movements, waves in oceans, flying birds, sudden rains, shadowing etc. Figure 3.4 shows four main types of dynamics in the scene corresponding to camera jitter,

**Camera Jitter**

**Tree leaves movement**

**Water rippling**

**Waves in water**

**Figure 3.4 Dynamic backgrounds [44]**

tree leaves movement, water rippling, waves in the water. In background modeling a model of the background is learned for performing the background subtraction.

### 3.2.1 Background Modeling

Background modeling is a process of learning a model of the background for accomplishing the goal of background subtraction. Thus, in a comprehensive manner the background modeling generates a model that is used to represent the background. As already mentioned in Figure 3.1 as soon as an appropriate

model is selected the background is initialized during the learning step. The detection of the first foreground is then done by classifying each pixel as a foreground or background pixel. The foreground mask is then applied on the current frame to obtain the moving objects. After this, the background is adapted over time by following the changes that have occurred in the scene with time. There are two major categories of the background modeling techniques that are available namely statistical background modeling and subspace background modeling both of which are summarized in Figure 3.5.

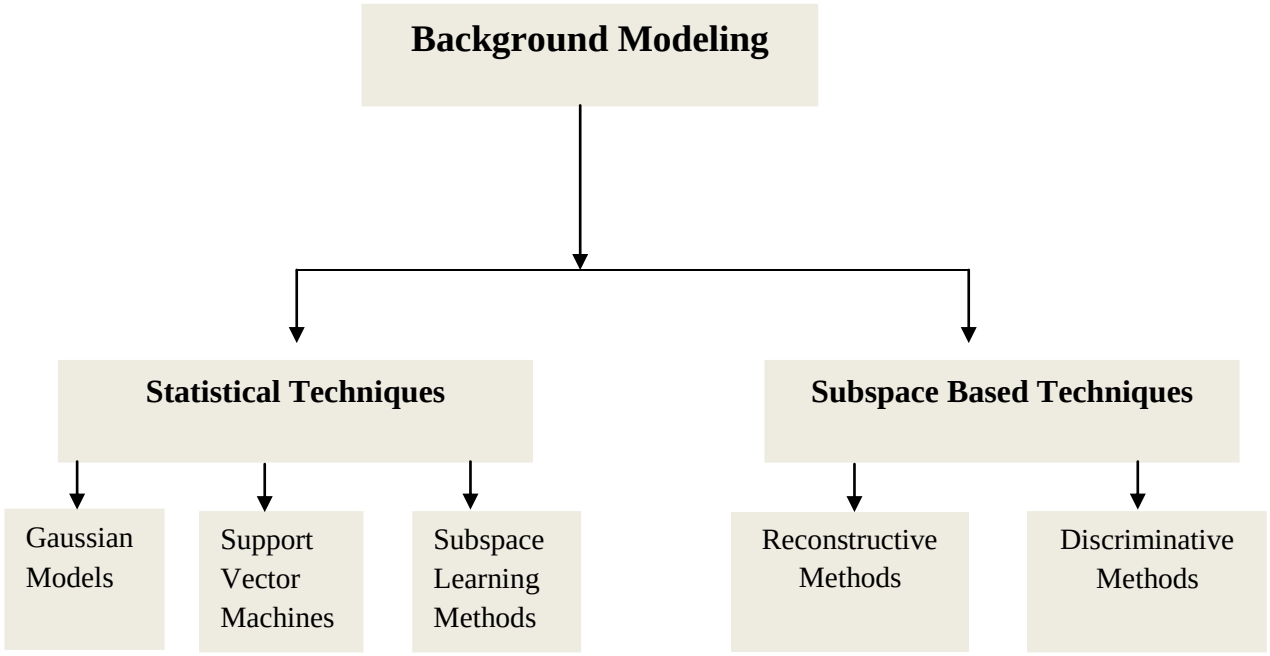


**Figure 3.5 Background Modeling Techniques**

### (i) Statistical Background Modeling

The statistical models are regarded to be more robust for handling illumination changes and dynamic backgrounds. There is a simplest way to represent the background which is by assuming the history of the pixel's intensity values over time and can be modeled by a Gaussian. Following this idea, Wren et al. [7] have proposed to use a Single Gaussian (SG) for background modeling. Kim et al. [45] have generalized the SG using Single General Gaussian (SGG) to alleviate the constraint of a strict Gaussian. Stauffer C, Grimson [8] observed that a unimodal model cannot handle dynamic backgrounds in the presence of waving trees, water rippling or moving algae and solved the problem by proposing the Mixture of Gaussians (MOG) which is used to model dynamic backgrounds.

On the same line, Allili et al. [46] proposed the mixture of general Gaussians (MOGG) as the earlier method [8] has some disadvantages in some circumstances like the background having fast variations which cannot be accurately modeled with just a few Gaussians which is 3 to 5 and thus posing the problems for sensitive detection. A nonparametric technique was developed by [47] known as KDE for estimating background probabilities at each pixel from many recent samples over time. Although this method is more efficient but it is time consuming. The above five models can be classified as first category models due to the fact that they are based on the Gaussian models. The second category makes use of more sophisticated statistical models such as support vector machine (SVM) by Lin H et. al [48], support vector regression (SVR) by Wang J et. al [49] and support vector data description (SVDD) by Tavakkoli et. al [50].This category is not discussed in this study as our work is not related to it in any way.

**(ii) Subspace Background Modeling**

The Subspace Learning methods form the third category of Statistical background modeling technique. Oliver et. al [73] in his work apply Subspace Learning using Principal Component Analysis (SL-PCA) on N images to construct a background model, which is represented by the mean image and the projection matrix comprising of the first p significant eigenvectors of PCA. And, the foreground segmentation is accomplished by computing the difference between the input image and its reconstruction or, in a lucid manner projection on the subspace. Following the same idea, many improvements of SL-PCA are developed to develop more robust and fast methods in the same category, Yamazaki et al.[51] and Tsai et al. [52] have used an independent component analysis (SL-ICA). Another method proposed by Bucak et al. [53] applies incremental non-negative matrix factorization (SL-INMF) to reduce the dimensions. In order to take into account the spatial information, Li et al.[54] have used an incremental rank-(R1, R2, R3) tensor (SL-IRT). In general, the Gaussian models and support vector models are applied for dynamic backgrounds where as subspace learning models are used for illumination changes. The subspace methods can be further classified as reconstructive and discriminative methods as illustrated in Figure 3.6.

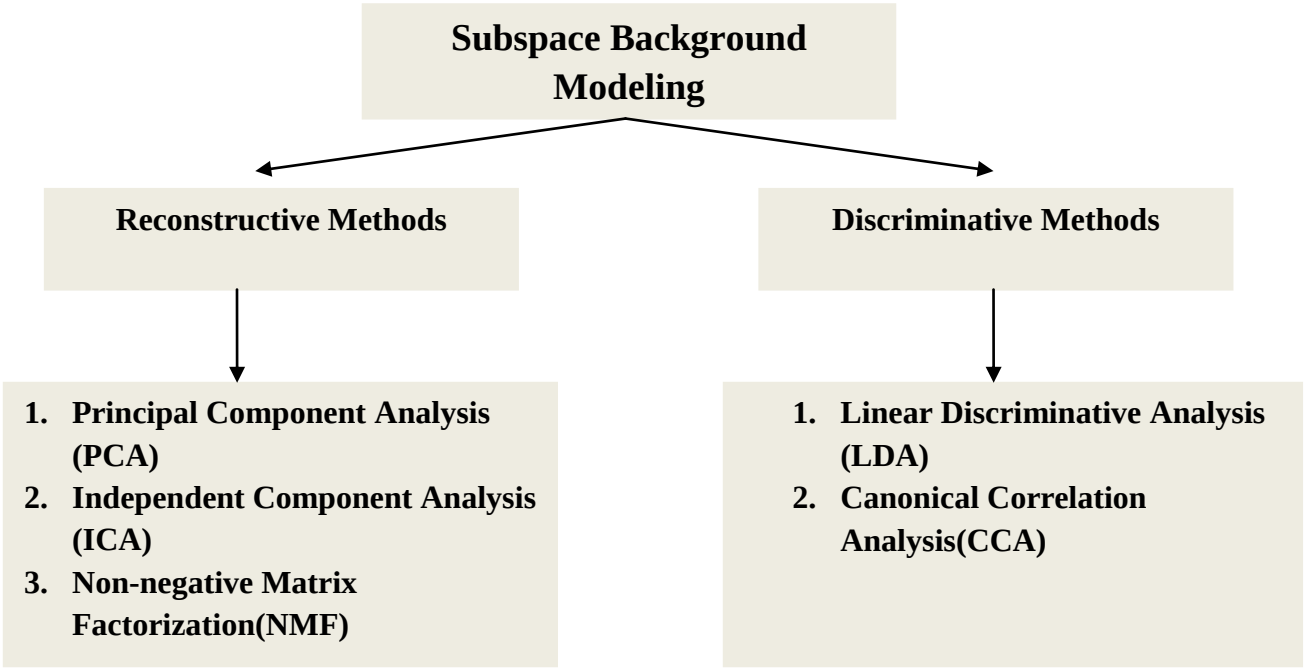


**Figure 3.6 Subspace learning Methods**

## 3.3 Tensor based Foreground Segmentation techniques

Matrices and matrix operations like SVD or PCA have played a vital role in obtaining the patterns in the case when data available is two dimensional i.e can be represented as a matrix. Major applications of it includes pattern recognition in image processing applications, data compression, content based image recognition (CBIR), Hyper spectral Image analysis (HSI), feature selection, information retrieval etc. These are powerful matrix based tools but cannot handle the problems that require information of a higher dimensional space as the matrices have only two dimensions corresponding to the intensity and time in context of a real time video. But, for more accuracy and more real performance there is a need of more dimensions containing features like color, texture, pose etc. A tensor basically fulfills this need. A tensor is a generalization of a matrix and contains large number dimensions of features. For a lot of applications involving higher dimensional space the existing framework of vector and matrix algebra appears to be insufficient or inappropriate.

Higher-order tensors are currently gaining importance due to the developments in the field of higher-order statistics (HOS). The basic HOS includes higher-order moments, cumulants, spectra and cepstra .The HOS of multivariate stochastic variables form a highly symmetric higher-order tensors in a similar way as the covariance of a stochastic vector is a symmetric (Hermitian) matrix. Higher-order tensors are

proved to be of great importance not merely in HOS but also in various other disciplines, like chemometrics, psychometrics, econometrics, image processing, biomedical signal processing, etc. Most often they appear in factor analysis of multiway datasets. Another use is as a formalism to describe linear vector-matrix, matrix-vector, and matrix-matrix transformations in the same manner as the transformations between vector spaces is described by matrices.

With the use of tensors we can attack an even wider range of problems that cannot be solved through matrices. For instance, higher order image analysis, a large dimensional data handling, time-evolving network traffic data etc. In a specific case of network monitoring we can consider a scenario where network flow is arriving very fast and continuously through routers, where each flow consists of source IP, destination IP, port number and the number of packets. In this regard Tensor analysis enables the systems to monitor the dynamic traffic behavior, to identify anomalies which can signify a potential intrusion, or worm propagation, or an attack and thus aids in handling the correlations across various sources, destinations and ports. The network flow example is shown in Figure 3.7, the tensor for a given time period has three modes or the three dimensions containing the source, destination and port and can be viewed as a 3D data cube. An entry (*i, j, k*) in that tensor can be represented like the small blue cube, and contains a variety of information such as number of packets i.e information from the corresponding source *(i)* to the destination *(j)* through port *(k)*, during a given time period.

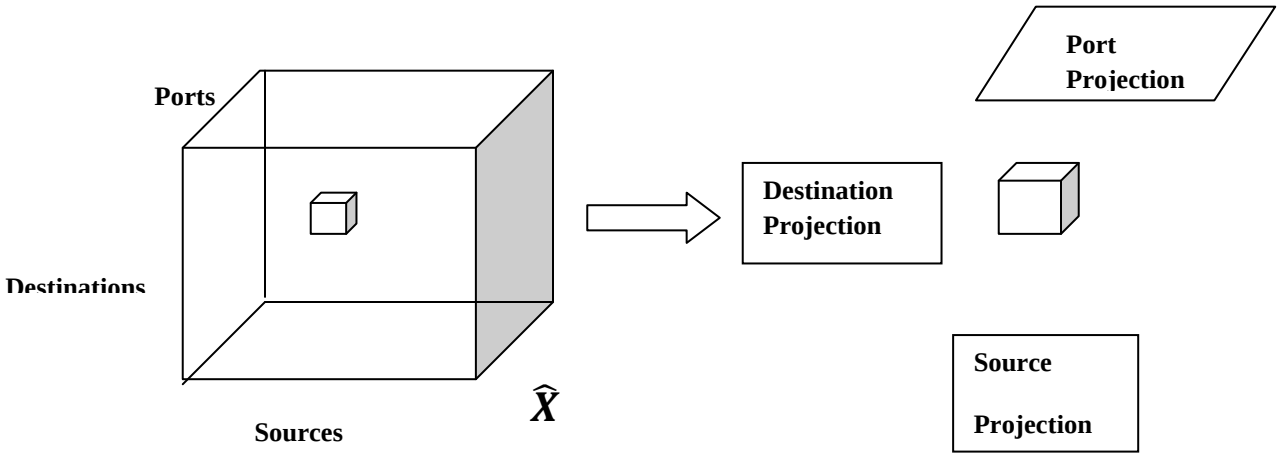


**Figure 3.7 A Third order Tensor representation of network flow data**

As the new tensors are arriving continuously over time thus the dynamic aspect in tensor applications comes in.

Various Tensor based tools are available in literature, using offline (OTA), dynamic (DTA), the streaming (STA) and windowed tensor analysis (WTA) which can incrementally collect and summarize large tensors. DTA and STA are fast and nimble since they avoid storing old tensors. They are also fully automatic, without requiring any user-defined parameters. All of these are discussed in concise way below:

### 3.3.1 OTA

Offline Tensor Analysis is the generalization of PCA (Principal Component Analysis) for the case of higher order tensors. The PCA requires input to be vectors i.e first order tensors. But unlike it, OTA can work on $M^{th}$ order tensors for dimensionality reduction. Figure 3.8 shows an example of OTA over n-$2^{nd}$ order tensors. Unfortunately, the closed form solution for OTA does not exist.

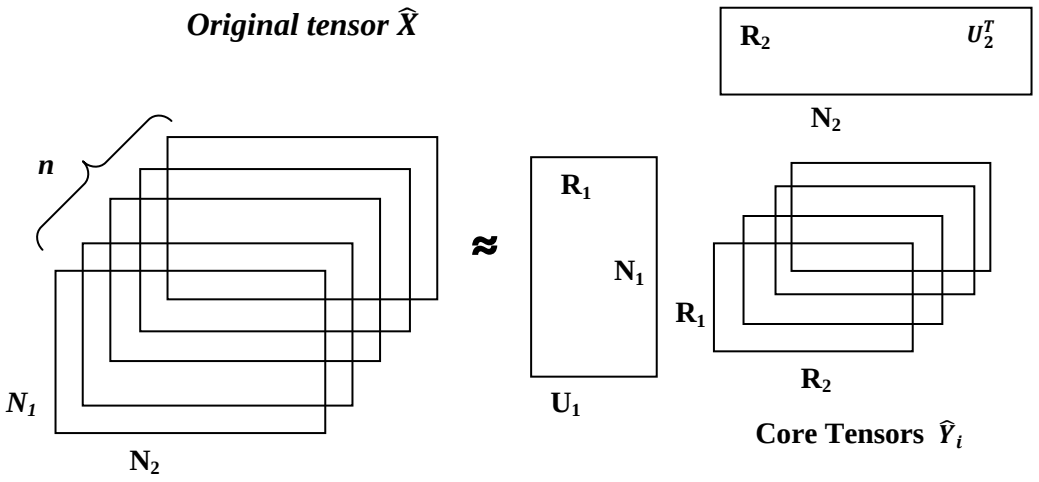


**Figure 3.8 OTA projects n large $2^{nd}$ order tensor $X_i$ into n smaller $2^{nd}$ order tensors Yi with two projection matrices $U_1$ and $U_2$ (one for each mode)[56]**

### 3.3.2 DTA

DTA stands for Dynamic Tensors Analysis and is a dynamical process that provides a compact summary for high order and high dimensional data. Tensor is a k-dimensional data cube. It is a method to compress tensors; an expansion for compressing vectors. In work by Jimeng Sun et al.[56] the authors suggest a method for expanding PCA to a method for choosing the best subset to project

matrices. As we know, PCA is a good and well-known method for data compression and feature extraction. So, having a set of *n* vectors, each containing *m* elements, PCA selects *r* orthogonal directions in which the projection of *n* vectors has maximum variance, or minimum squared error from original data. DTA possesses the following elementary features.

- Scalability
- Space efficiency as, it does not need to store the past, and
- Fully automatic with no need for user defined parameters.

Moreover, there are variants of DTA like STA, a streaming tensor analysis method, which provides a fast, streaming approximation to DTA.

### 3.3.3 STA

Streaming Tensor Analysis (STA) is a speed-up approximation of DTA. The Streaming Tensor Analysis (STA) is a fast algorithm to approximate DTA without diagonalization. As for most of time-critical applications employing tensors the process of diagonalization can be quite expensive. The key component of tracking a projection matrix is first discussed under, and the algorithm describing the process is mentioned in Figure 3.9: The basic idea of the algorithm is to continuously track the changes of projection matrices using the online PCA technique. This is done by reading a new vector (one row of a tensor matrix) and following three steps:

1. *Projection Computation*: Compute the projection (*y)*, based on the current projection matrix (*U)*, by projecting (*x: a row of tensor)* onto *U*;

2. *Reconstruction error*: Estimate the reconstruction error (*e*) and the energy, based on the *y* values; and

3. *Update:* Update the estimates of *U*.

Intuitively, the goal of the tracking algorithm is to adaptively update *U* quickly based on the new tensor. The larger the error *e*, the more *U* is updated. However, the magnitude of this update should also take into account the past data currently captured by *U*. For this reason, the update is inversely proportional to the current energy *s (i)*, which corresponds to the eigen value.

**Input:**

Projection matrix U $\epsilon\ R^{n\times r}$

Energy vector s $\epsilon\ R^{n}$

Input vector x $\epsilon\ R^{n}$

**Output:**

Updated projection matrix U

Updated energy vector s

1. As each point $x_{t+1}$ arrives initialize $\dot{x}$ : =$x_{t+1}$.

2. For $1\leq i \leq r$

y (i) = U (: , i)$^{T}$ $\dot{x}_1$ (projection onto U (:, i))

s(i) ← s(i) + y(i)$^{2}$ (energy ∝ i-th eignval)

e = $\dot{x}$ - y(i) U (:,i) (error, e ⊥ U (:,i))

U(:,i) ← U(:,i) + $\frac{1}{s(i)}\ y(i)e$ (update PC estimate)

$\dot{x}$ (i+1) ← $\dot{x}_t$ - y(i) U (:, i) (repeat with remainder)

**Figure 3.9 Tracking a projection matrix**

### 3.3.4 WTA

Window-based tensor analysis (WTA) for tensor streams deals with summarizing the tensor windows efficiently, using small core tensors associated with different projection matrices, where core tensors and projection matrices are analogous to the singular values and singular vectors for a matrix.

Two variations of the algorithms for WTA are present that are:

1) *Independent-window tensor analysis (IW)* which treats each tensor window independently;

2) *Moving window tensor analysis (MW)* which exploits the time dependence across neighboring windows to reduce computational cost significantly.

There is an important practical application of WTA described as Multi-Aspect Correlation Analysis (MACA), which simultaneously finds correlations within a single aspect and also across multiple aspects.

### 3.3.5 HOSVD

The generalization of the matrix SVD to higher-order tensors is obtained by HOSVD. In the HOSVD the role of the matrix in singular values is assumed by a tensor with the property of all orthogonality [57]. The resulting decomposition i.e. HOSVD is always possible for real or complex $N^{th}$-order tensors. The HOSVD of an $N^{th}$-order tensor can be viewed as an N matrix SVDs. Thus, in sequence several matrix properties possess a very clear higher-order counterpart. There is a link between tensor symmetry and HOSVD that, if a higher order tensor is transformed into itself or into its complex conjugate by a permutation of the indices, then this symmetry is reflected by the HOSVD components.

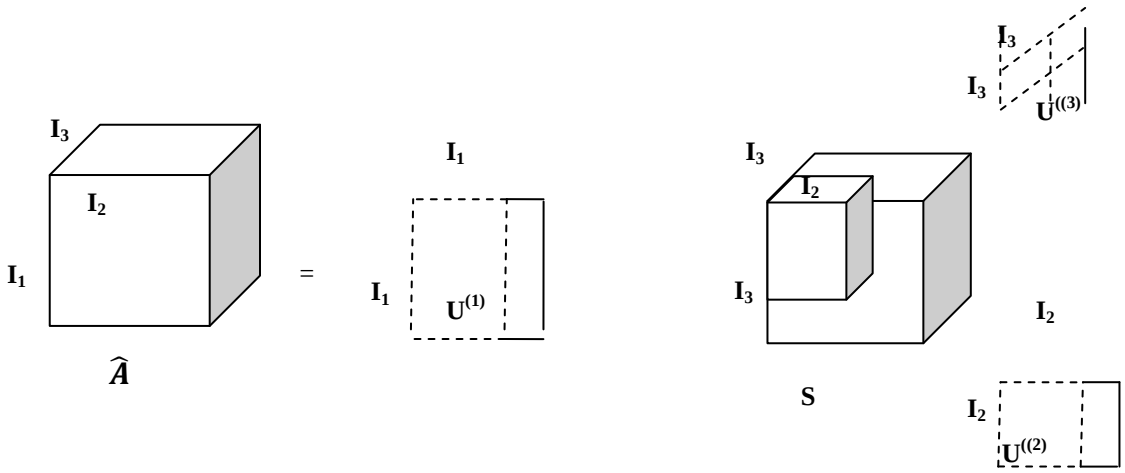


**Figure 3.10 Visualization of the HOSVD for the third order tensor.**

This link of HOSVD with the SVD of matrices makes it a proper tool for the analyzing the n-mode vector space properties. Unlike the matrix case, the core tensor is in general a full tensor, instead of being pseudo-diagonal. This results in two main differences with the matrix case. First, the HOSVD of a given tensor allows us to estimate its n-ranks, which is only a rough lower-bound on the rank; more in general, the HOSVD does not allow for an interpretation in terms of minimal expansions in rank-1 terms. This shortcoming is inherent in the structure of multilinear algebra.

Second, if the smallest n-mode singular values are discarded, it does not necessarily lead to the best possible approximation with reduced n-rank values $(\mathbf{1} \leq \boldsymbol{n} \leq \boldsymbol{N})$. However, if there is a large gap between the n-mode singular values, then the approximation can still be qualified as accurate.

Step 1: Compute

$$\boldsymbol{\mu_{A*}} = \frac{\boldsymbol{I_A}}{\boldsymbol{I_A + I_F}} \boldsymbol{\mu_A} + \frac{\boldsymbol{I_F}}{\boldsymbol{I_A + I_F}} \boldsymbol{\mu_F}$$

Where $\boldsymbol{I_A}$ is the number of the columns in $\boldsymbol{A}'$ and $\boldsymbol{I_F}$ is the number of columns in $\boldsymbol{F}'$.

Step 2: Construct a new matrix E:

$$\text{E} = \left( (\boldsymbol{F}' - \boldsymbol{\mu_F}\, \boldsymbol{l_1} \times \boldsymbol{I_F}) \mid \sqrt{\frac{\boldsymbol{I_A I_F}}{\boldsymbol{I_A + I_F}}} (\boldsymbol{\mu_A} - \boldsymbol{\mu_F}) \right)$$

Where $\boldsymbol{l_1} \times \boldsymbol{I_F}$ is a $\boldsymbol{I_F}$- dimensional row vector whose elements are all "1", i.e. $\overbrace{\mathbf{1}, \mathbf{1}, \ldots\ldots, \mathbf{1}}^{\boldsymbol{I_F}}$.

Step 3: Compute the QR decomposition of E to obtain the eigenbasis É of E. Let matrix $\boldsymbol{U}' \boldsymbol{be}\, \boldsymbol{U}' = (\boldsymbol{U_A} | \text{É})$.

Step 4: Let matrix $\boldsymbol{V}'\ \boldsymbol{be}$

$$\boldsymbol{V}' = \begin{pmatrix} \boldsymbol{V_A} & \mathbf{0} \\ \mathbf{0} & \boldsymbol{\Omega_{I_F}} \end{pmatrix}$$

Where $\boldsymbol{\Omega_{I_F}}$ is the $\boldsymbol{I_F} \times \boldsymbol{I_F}$ identity matrix. Then, matrix $\boldsymbol{\Sigma}'$, is defined as:

$$\boldsymbol{\Sigma}' = \begin{pmatrix} \boldsymbol{\Sigma_A} & (\boldsymbol{U_A})^{\boldsymbol{T}} & \boldsymbol{E} \\ \mathbf{0} & \text{É}^{\boldsymbol{T}} & \boldsymbol{E} \end{pmatrix}.$$

Step 5: Compute the SVD of $\boldsymbol{\Sigma}'$ :

$\boldsymbol{\Sigma}' = \widetilde{\boldsymbol{U}} \widetilde{\boldsymbol{\Sigma}} \widetilde{\boldsymbol{V}}^{\boldsymbol{T}}$. Then, SVD of $(\boldsymbol{A}' | \boldsymbol{F}')$ is obtained: $\boldsymbol{U_{A*}} = \boldsymbol{U}' \widetilde{\boldsymbol{U}}, \boldsymbol{\Sigma_{A*}} = \widetilde{\boldsymbol{\Sigma}}$, and

$(\boldsymbol{V_{A*}})^{\boldsymbol{T}} = (\widetilde{\boldsymbol{V}})^{\boldsymbol{T}} (\boldsymbol{V}')^{\boldsymbol{T}}$.

**Figure 3.11 Algorithm: Incremental SVD**

### 3.3.6 IRTSA-GBM and IRTSA-CBM

As the current pixel based statistical models are sensitive to dynamic scenes such as illumination change, shadow movement, swaying tree etc. there is a need of effective modeling of scenes for foreground segmentation. The method by Li X et al.[58] is a block based background model using incremental Rank ($R_1$,$R_2$,$R_3$) tensor based subspace learning algorithm applicable for grayscale and colored image sequences. It incrementally learns a low-order tensor based eigen space representations that has a full capability to capture the intrinsic spatio-temporal characteristics of the scenes and, proves itself to be a robust foreground segmentation method. An effective subspace analysis is achieved by the use of higher order tensor learning algorithm. And, the problem of modeling the appearance scenes can be reduced to making the tensor decomposition more accurate and efficient. The extended version of the classical incremental SVD algorithm [59] by Ross et al.[4] is used and summarized in Figure 3.11. The IRTSA algorithm has the capability of efficiently computing the SVD of *A*(n)*, when SVD of a *mode-k* unfolding matrix as shown in Figure 3.12 [6].

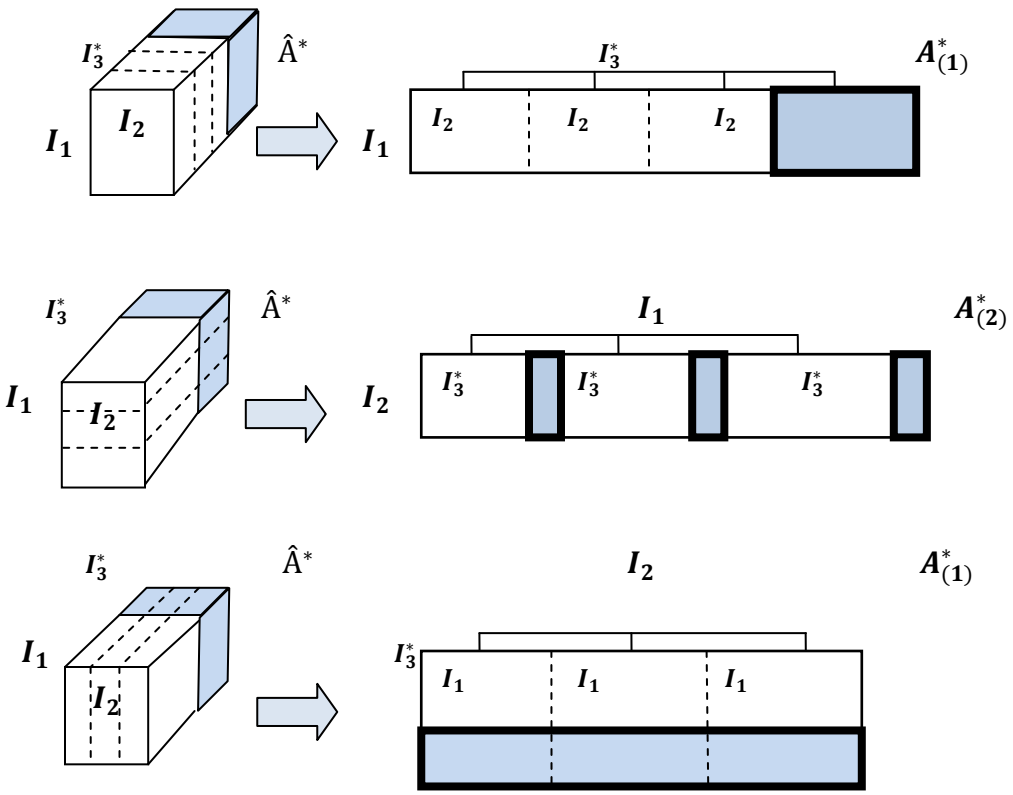


**Figure 3.12 Unfolding an extended 3-order Tensor [6]**

The two steps procedure for foreground segmentation takes place by learning an online background model and then in the second step model matching is performed. To achieve this particular aim, a tensor containing previous frames is constructed and to execute its incremental SVD operation , unfolding is done by following the algorithm as explained in Figure 3.12 is applied. Then, a test frame whose foreground is to be detected is projected over the subspace created above in order to match the moving regions in the consecutive frames with the learned tensor based eigen space background model. The matching criterion used here is the reconstruction error. The foreground and background are separated as follows

$$(u, v) \in \begin{cases} \text{background} & \text{if } \exp\left(-\frac{RM_{uv}^2}{2\sigma^2}\right) > T_{gray} \\ \text{foreground} & \text{otherwise} \end{cases} \tag{3.1}$$

where, *RM* is the reconstruction error, and $T_{gray}$ is the threshold. The whole process to achieve background subtraction is explained in the block diagram as shown in the Figure 3.13.

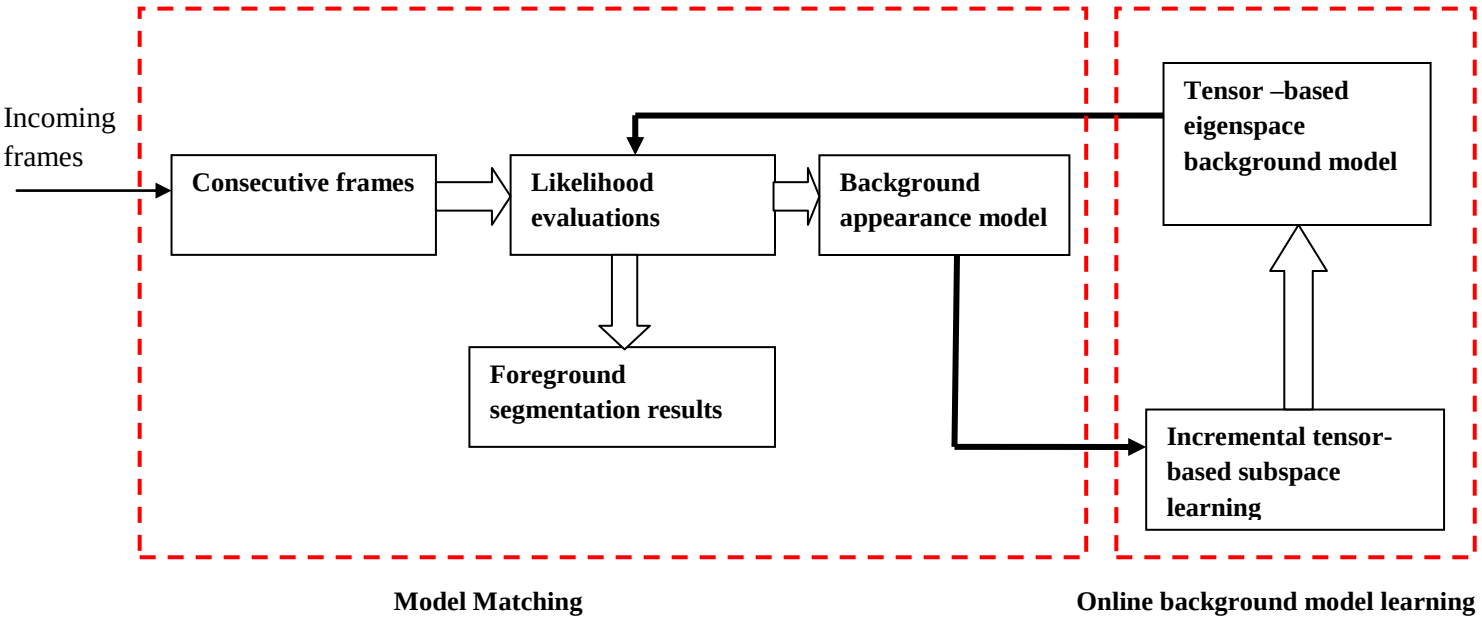


**Figure 3.13 Architecture of the foreground segmentation framework [6]**

This same method can also be applied for the colored image sequences by applying the same method on the three different channels, three different channels chosen are RGS in [6] and the technique is named as IRTSA-CBM. Thus, for the colored videos an RGS model of the images is computed and the learning

process of IRTSA-CBM is similar to that of IRTSA-GBM with only difference that IRTSA-CBM has the three tensor based eigen space models corresponding to the three tensor based eigen space models, while only one component is present in IRTSA-GBM .