Figure-ground segmentation - pixel-based

Ahmed Elgammal

Abstract Background subtraction is a widely used concept to detect moving objects in videos taken from a static camera. In the last two decades several algorithms have been developed for background subtraction and were used in various important applications such as visual surveillance, sports video analysis, motion capture, *etc.*Various statistical approaches have been proposed to model scene background. In this chapter we review the concept and the practice in background subtraction. We discuss several basic statistical background subtraction models, including parametric Gaussian models and nonparametric models. We discuss the issue of shadow suppression, which is essential for human motion analysis applications. We also discuss approaches and tradeoffs for background maintenance. We also point out many of the recent developments in background subtraction paradigm

1 Introduction

In many human motion analysis applications stationary cameras or pan-tilt-zoom (PTZ) cameras are used to monitor activities at outdoor or indoor sites. This is typical in visual surveillance systems as well as vision-based motion capture systems. Since the cameras are stationary, the detection of moving objects can be achieved by comparing each new frame with a representation of the scene background. This process is called *background subtraction* and the scene representation is called the *background model*. The scene here is assumed to be stationary or quasi stationary.

Typically, the background subtraction process forms the first stage in automated visual surveillance systems and motion capture applications. Results from background subtraction are used for further processing, such as tracking targets and understanding events. One main advantage of pixel-based detection using background subtraction is that the outcome is an accurate segmentation of the foreground regions

Ahmed Elgammal
Rutgers University, New Brunswick, NJ e-mail: elgammal@cs.rutgers.edu

from the scene background. For human subjects, the process gives accurate silhouettes of the human body which can be further used for tracking, fitting body limbs, pose and posture estimation, *etc.*. This is in contrast to classifier-based object-based detectors which mainly decides whether a bounding box or a region in the image contains the object of interest or not, *e.g.*pedestrian detectors. Such object-based detector will be discussed in Chapter **??**.

The concept of background modeling is rooted in photography since the 19th century where it was shown that film can be exposed for a period of time to capture the scene background without moving objects [14]. The use of background subtraction to detect moving objects is deeply rooted in image analysis and emanated from the concept of change detection, a process in which two images of the same scene taken at different time instances are compared, for example in Landsat Imagery, *e.g.* [8, 26].

The concept of background subtraction has been widely used since the early human motion analysis systems such as Pfinder [58], W4 [19], *etc.*. Efficient and more sophisticated background subtraction algorithms that can address challenging situations have been developed since then. The success of these algorithms lead to the growth of many commercial applications, for example sports monitoring and automated visual surveillance industry. Unlike earlier background subtraction algorithms while the cameras and the scenes are assumed to be stationary, many approaches have been proposed to overcome these limitations, for example dealing with quasi-stationary scenes and moving cameras. We will discuss such approaches later in this chapter.

The organization of this chapter is as follows. Section 2 discusses some of the challenges in building a background model for detection. Section 3 discusses some of the basic and widely used background modeling techniques. Section 4 discusses how to deal with color information to avoid detecting shadows. Section 5 discusses the tradeoffs and challenges in updating background models. Section 6 discusses some background models that can deal with moving cameras. Finally in Section 7 we discuss further issues and point out to further readings in the subject.

2 Challenges in Scene Modeling

In any indoor or outdoor scene there are changes that occur over time to the scene background. It is important for any background model to be able to tolerate these changes, either by being invariant to them or by adapting to them. These changes can be local, affecting only parts of the background, or global affecting the entire background. The study of these changes is essential to understand the motivations behind different background subtraction techniques. Toyama *et al.* [54] identified a list of ten challenges that a background model has to overcome, and denoted them by: *Moved objects, Time of day, Light switch, Waving trees, Camouflage, Bootstrapping, Foreground aperture, Sleeping person, Walking person, Shadows*. Elgammal

et al. [11] classifies the possible changes in a scene background according to their source:

Illumination changes:

- Gradual change in illumination as might occur in outdoor scenes due to the change in the relative location of the sun during the day.
- Sudden change in illumination as might occur in an indoor environment by switching the lights on or off, or in an outdoor environment, *e.g.*a change between cloudy and sunny conditions.
- Shadows cast on the background by objects in the background itself (*e.g.*, buildings and trees) or by moving foreground objects.

Motion changes:

- Global image motion due to small camera displacements. Despite the assumption that cameras are stationary, small camera displacements are common in outdoor situations due to wind load or other sources of motion which causes global motion in the images.
- Motion in parts of the background. For example, tree branches moving with the wind, or rippling water.

Structural Changes:

These are changes introduced to the background, including any change in the geometry or the appearance of the background of the scene introduced by targets. Such changes typically occur when something relatively permanent is introduced into the scene background. For example, if somebody moves (introduces) something from (to) the background, or if a car is parked in the scene or moves out of the scene, or if a person stays stationary in the scene for an extended period, *etc.*. Toyama *et al.* [54] denoted these situations by "Moved Objects", "sleeping person" and "walking person" scenarios.

A central issue in building a representation for the scene background is what features to use for this representation or, in other words, what to model in the background. In the literature a variety of features have been used for background modeling including pixel based features (pixel intensity, edges, disparity) and region based features (*e.g.*, image block). The choice of the features affects how the background model will tolerate the changes in the scene and the granularity of the detected foreground objects.

Another fundamental issue in building a background representation is the choice of the statistical model that explains the observation at a given pixel or region in the scene. The choice of the proper model depends on the type of changes expected in the scene background. Such a choice highly affects the accuracy of the detection. Section 3 discusses some of the statistical models that are widely used in background modeling context. Beyond choosing the features and the statistical model, maintaining the background representation is another challenging issue that we will discuss in Section 5.

3 Statistical Scene Modeling

In this section we will discuss some of the existing and widely used statistical background modeling approaches. For each model we will discuss how the model is initialized and how it is maintained. For simplicity of the discussion we will use pixel intensity as the observation. Instead, color or any other features can be used. At the pixel level, the process of background subtraction can be formulated as followed: Given the intensity observed at a pixel at time t, denoted by x_t, we need to classify that pixel to either the background or foreground classes. This is a two class classification problem. However, since the intensity of a foreground pixel can arbitrary take any value, unless some further information about the foreground is available, we can just assume that the foreground distribution is uniform. Therefore, the problem reduces to a one class classification problem, *i.e.*, modeling the distribution of the background class, which can be achieved if a history of background observations are available at that pixel. If the history observation is not purely coming from the background, *i.e.*, foreground objects are present in the scene, the problem become more challenging.

3.1 Parametric Background Models

Pixel intensity is the most commonly used feature in background modeling. In a completely static scene, a simple noise model that can be used is an independent stationary additive Gaussian noise model [13]. According to that model, the noise distribution at a given pixel is a zero mean Gaussian distribution $N(0,\sigma^2)$, it follows that the observed intensity at that pixel is a random variable with a Gaussian distribution $N(\mu,\sigma^2)$. This Gaussian distribution model for the intensity value of a pixel is the underlying model for many background subtraction techniques and widely know as a single Gaussian background model. For the case of color images, a multivariate Gaussian is used. Typically, the color channels are assumed to be independent which reduces a multivariate Gaussian to a product of single Gaussians, one for each color channel. More discussion about dealing with color will be presented in Section 4

Estimating the parameters for this model, *i.e.*, learning the background model, reduces to estimating the sample mean and variance from history pixel observations. The background subtraction process in this case is a classifier that decides whether a new observation at that pixel comes form the learned background distribution. Assuming the foreground distribution is uniform, this amounts to putting a threshold on the tail of the Gaussian, *i.e.*, the classification rule reduces to marking a pixel as foreground if

$$\|x_t - \hat{\mu}\| > k\hat{\sigma}$$

where $\hat{\mu}$ and $\hat{\sigma}$ are the estimated mean and standard deviation and k is a threshold. The parameter σ can be even assumed to be the same for all pixels. So, literally,

this simple model reduces to subtracting a background image B from the each new frame I_t and checking the difference against a threshold. In such case the background image B is the mean of the history background frames.

This basic single Gaussian model can made adaptive to slow changes in the scene (for example, gradual illumination changes) by recursively updating the mean with each new frame to maintain a background image

$$B_t = \frac{t-1}{t} B_{t-1} + \frac{1}{t} I_t,$$

where $t \geq 1$. Obviously this update mechanism does not forget the history and, therefore, the effect of new images on the model tends to zero. This is not suitable when the goal is to adapt the model to illumination changes. Instead the the mean and variance can be computed over a sliding window of time. However, a more practical and efficient solution is to recursively update the model via temporal blending, also known as exponential forgetting, *i.e.*

$$B_t = \alpha I_t + (1-\alpha) B_{t-1}. \tag{1}$$

Here, B_t denotes the background image computed up to frame t. The parameter α controls the speed of forgetting old background information. This update equation is a low-pass filter with a gain factor α that effectively separates the slow temporal process (background) from the fast process (moving objects). Notice that the computed background image is no longer the sample mean over the history but captures the central tendency over time [16]. This basic adaptive model is used in systems such as the Pfinder [58]. In [30, 29, 33] variations of this recursive update was used after masking out the foreground regions.

Typically, in outdoor environments with moving trees and bushes, the scene background is not completely static. For example, one pixel can be the image of the sky in one frame, a tree leaf in another frame, a tree branch in a third frame and some mixture subsequently. In each situation the pixel will have a different intensity (color), so a single Gaussian assumption for the probability density function of the pixel intensity will not hold. Instead, a generalization based on a Mixture of Gaussians (MoG) has been used in [14, 57, 56] to model such variations. This model was first introduced in [14], where a mixture of three Gaussian distributions was used to model the pixel value for traffic surveillance applications. The pixel intensity was modeled as a weighted mixture of three Gaussian distributions corresponding to road, shadow and vehicle distribution. Fitting a mixture of Gaussian (MoG) model can be achieved using the Expectation Maximization (EM) algorithm [6]. However this is impractical for a realtime background subtraction application. An incremental EM algorithm [40] was used to learn and update the parameters of the model.

Stauffer and Grimson [57, 56] proposed a generalization to the previous approach. The intensity of a pixel is modeled by a mixture of K Gaussian distributions (K is a small number from 3 to 5). The mixture is weighted by the frequency with which each of the Gaussians explains the background. The probability that a certain pixel has intensity x_t at time t is estimated as

$$Pr(x_t) = \sum_{i=1}^{K} w_{i,t} G(x_t, \mu_{i,t}, \Sigma_{i,t}), \tag{2}$$

where $w_{i,t}$, $\mu_{i,t}$, and $\Sigma_{i,t} = \sigma_{i,t}\mathbf{I}$ are the weight, mean, and covariance for the i-th Gaussian mixture component at time t respectively.

The parameters of the distributions are updated recursively using online K-means approximation. The mixture is weighted by the frequency with which each of the Gaussians explains the background, *i.e.*, a new pixel value is checked against the existing K Gaussians and when a match is found the weight for that distribution is updated as follows

$$w_{i,t} = (1-\alpha) w_{i,t-1} + \alpha M(i,t),$$

where $M(i,t)$ is an indicator variable which is 1 if the i-th component is matched, 0 otherwise. The parameter of the matched distributions are updated as follows

$$\mu_t = (1-\rho)\mu_{t-1} + \rho x_t,$$

$$\sigma_t^2 = (1-\rho)\sigma_{t-1}^2 + \rho (x_t - \mu_t)^T (x_t - \mu_t).$$

The parameters α and ρ are two learning rates. The K distributions are ordered based on w_j/σ_j^2 and the first B distributions are used as a model of the background of the scene where B is estimated as

$$B = \arg\min_b \left(\sum_{j=1}^{b} w_j > T \right). \tag{3}$$

The threshold T is the fraction of the total weight given to the background model. Background subtraction is performed by marking any pixel that is more that 2.5 standard deviations away from any of the B distributions as a foreground pixel.

The MoG background model was shown to perform very well in indoor and outdoor situations. Many variations has been suggested to the Stauffer and Grimson's model [56], *e.g.* [38, 27, 20]. The model also was used with in different feature spaces and/or with a subspace representations. Gao *et al.* [16] studied the statistical error characteristic of MoG background models.

3.2 Non-parametric Background Models

In outdoor scenes, typically there are wide range of variations, which can be very fast. Outdoor scenes usually contains dynamic areas such as waving trees and bushes, rippling water, ocean waves. Such fast variations are part of the scene background. Modeling such dynamics areas requires a more flexible representation of the background probability distribution at each pixel. This motivates the use of non-parametric density estimator for background modeling [12].

A particular nonparametric technique that estimates the underlying density and is quite general is the kernel density estimation (KDE) technique [47, 7]. Given a sample $S = \{x_i\}_{i=1..N}$ from a distribution with density function $p(x)$, an estimate $\hat{p}(x)$ of the density at x can be calculated using

$$\hat{p}(x) = \frac{1}{N}\sum_{i=1}^{N} K_\sigma(x - x_i), \tag{4}$$

where K_σ is a kernel function (sometimes called a "window" function) with a bandwidth (scale) σ such that $K_\sigma(t) = \frac{1}{\sigma}K(\frac{t}{\sigma})$. The kernel function K should satisfy $K(t) \geq 0$ and $\int K(t)dt = 1$. Kernel density estimators asymptotically converge to any density function with sufficient samples [47, 7]. In fact, all other nonparametric density estimation methods, e.g., histograms, can be shown to be asymptotically kernel methods [47]. This property makes these techniques quite general and applicable to many vision problems where the underlying density is not known [3, 9]. We can avoid having to store the complete data set by weighting a subset of the samples as

$$\hat{p}(x) = \sum_{x_i \in B} \alpha_i K_\sigma(x - x_i),$$

where α_i are weighting coefficients that sum up to one and B is a sample subset. A good discussion of KDE techniques can be found in [47].

Elgammal *et al.* [12] introduced a background modeling approach based on kernel density estimation. Let $x_1, x_2, ..., x_N$ be a sample of intensity values for a pixel. Given this sample, we can obtain an estimate of the probability density function of the pixel intensity at any intensity value using kernel density estimation using Eq. 4. This estimate can be generalized to use color features or other high dimensional features by using kernel products as

$$Pr(x_t) = \frac{1}{N}\sum_{i=1}^{N}\prod_{j=1}^{d} K_{\sigma_j}(x_{t_j} - x_{i_j}), \tag{5}$$

where x_t is a d dimensional color feature at time t and K_{σ_j} is a kernel function with bandwidth σ_j in the jth color space dimension.

A variety of kernel functions with different properties have been used in the literature of nonparametric estimation. Typically kernel functions are symmetric and unimodal functions that fall off to zero rapidly away from the center, i.e., the kernel function should have finite local support and points beyond certain window will have no contribution. The Gaussian function is typically used as a kernel for its continuity, differentiability and locality properties although it violates the finite support criterion [9]. Note that choosing the Gaussian as a kernel function is different from fitting the distribution to a Gaussian model (normal distribution). Here, the Gaussian is only used as a function to weight the data points. Unlike parametric fitting of a mixture of Gaussians, kernel density estimation is a more general approach that does not assume any specific shape for the density function.

Using this probability estimate, the pixel is considered a foreground pixel if $Pr(x_t) < th$, where the threshold th is a global threshold over all the image that can be adjusted to achieve a desired percentage of false positives. Practically, the probability estimation in Eq. 5 can be calculated in a very fast way using precalculated lookup tables for the kernel function values given the intensity value difference, $(x_t - x_i)$, and the kernel function bandwidth. Moreover, a partial evaluation of the sum in equation 5 is usually sufficient to surpass the threshold at most image pixels, since most of the image is typically from the background. This allows a realtime implementation of the approach.

(a) (b)

Fig. 1 Example of probability estimation using a nonparametric model (a) original image. (b) Estimated probability image.

Since kernel density estimation is a general approach, the estimate of Eq. 4 can converge to any pixel intensity density function. Here the estimate is based on the most recent N samples used in the computation. Therefore, adaptation of the model can be achieved simply by adding new samples and ignoring older samples [12], *i.e.*, using a sliding window over time. Figure 1-b shows the estimated background probability where brighter pixels represent lower background probability pixels.

This nonparametric technique for background subtraction was introduced in [12] and has been tested for a wide variety of challenging background subtraction problems in a variety of setups and was found to be robust and adaptive. We refer the reader to [12] for details about the approach such as details about model adaptation and false detection suppression. Figures 2 shows two detection results for targets in a wooded area where tree branches move heavily and the target is highly occluded. Figure 3-top shows the detection results using an omni-directional camera. The targets are camouflaged and walking through the woods. Figure 3-bottom shows the detection result for a rainy day where the background model adapts to account for different rain and lighting conditions.

One major issue that needs to be addressed when using kernel density estimation technique is the choice of suitable kernel bandwidth (scale). Theoretically, as the number of samples reaches infinity, the choice of the bandwidth is insignificant

and the estimate will approach the actual density. Practically, since only a finite number of samples are used and the computation must be performed in real time, the choice of suitable bandwidth is essential. a too small bandwidth will lead to a ragged density estimate, while a too wide bandwidth will lead to an over-smoothed density estimate [7, 9]. Since the expected variations in pixel intensity over time are different from one location to another in the image, a different kernel bandwidth is used for each pixel. Also, a different kernel bandwidth is used for each color channel. In [12] a procedure was proposed for estimating the kernel bandwidth for each pixel as a function of the median of absolute differences between consecutive frames. In [39] an adaptive approach for estimation of kernel bandwidth was proposed. Parag *et al.* [41] proposed an approach using boosting to evaluate different kernel bandwidth choices for bandwidth selection.

KDE-Background Practice and Other Nonparametric models:

One of the drawbacks of the KDE background model is the requirement to store a large number of history samples for each pixel. In KDE literature many approaches was proposed to avoid storing a large number of samples. Within the context of background modeling, Piccardi and Jan [43] proposed an efficient mean shift approach to estimate the modes of a pixel's history PDF then a few number of Gaussians was used to model the PDF. Mean shift is a nonparametric an iterative mode seeking procedure [15, 2, 4]. With the same goal of reducing memory requirement, Han *et al.* [18] proposed a sequential kernel density estimation approach where variable bandwidth mean shift was used to detect the density modes. Unlike mixture of Gaussian methods where the number of Gaussian is fixed, technique such as [43, 18] can adaptively estimate a variable number of modes to represent the density, therefore keeping the flexibility of a nonparametric model while achieving the efficiency of a parametric model.

Efficient implementation of KDE can be achieved through building look-up tables for the kernel function values, which facilitates realtime performance. Fast Gauss Transform has been proposed for efficient computation of KDE [10], however, the Fast Gauss Transform is only justifiable with a large number of samples required for the density estimation as well as the need for estimation at many pixels in batches. For example, Fast Gauss implementation was effectively used in a layered background representation [42].

Many variations have been suggested to the basic nonparametric KDE background model. In practice nonparametric KDE has been used at the pixel level as well as at the region level or in a domain-range representation to model a scene background. For example, in [42] a layered representation was used to model the scene background where the distribution of each layer is modeled using KDE. Such layered representation facilitates detecting the foreground under static or dynamic background and in the presence of nominal camera motion. In [49] KDE was used in a joint domain-range representation of image pixel (r,g,b,x,y), which exploits the spatial correlation between neighboring pixels. Parag *et al.* [41] proposed an approach for feature selection for the KDE framework where boosting based ensemble learning was used to combine different features. The approach also can be

used to evaluate different kernels bandwidth choices for bandwidth selection. Recently, Sheikh *et al.* [48] used a KDE approach in a joint domain-range representation within a foreground/background segmentation framework from freely moving camera as will be discussed in Section 6.

In [35] a biologically inspired nonparametric background subtraction approach was proposed where a self-organizing artificial neural network model was used to model the pixel process. Each pixel is modeled with a sample arranged in a shared 2D grid of nodes where each node is represented with a weight vector with the same dimensionality as the input observation. An incoming pixel observation is mapped to the node whose weights are most similar to the input where a threshold function is used to decide background/foreground. The weights of each node are updated at each new frame using a recursive filter similar to Eq. 1. A interesting feature of that approach is that the shared 2D grid of nodes allows the spatial relationships between pixels to be taken into account at both the detection and update phases.

Fig. 2 Example background subtraction detection results: Left: original frames, Right: detection results.

4 Moving Shadow Suppression

A background subtraction process on gray scale images, or on color images without carefully selecting the color space, is bound to detect the shadows of moving objects along with the objects themselves. While shadows of static objects can typically be adapted in the background process, shadows casted by moving object, *i.e.*, dynamic shadows, constitute a sever challenge for foreground segmentation. Since the goal of

Fig. 3 Top: Detection of camouflaged targets from an omni-directional camera. Bottom: Detection result for a rainy day.

background subtraction is to obtain accurate segmentation of moving foreground regions for further processing, it is highly desirable to detect such foreground regions without casted shadow attached to them. This is particularly important for human motion analysis since shadows attached to silhouettes would cause problems in fitting body limbs and estimating body poses, consider the example shown in Figure 4. Therefore, extensive researches have addressed the detection/supression of moving (dynamic) shadows.

Avoiding the detection of shadows or suppressing the detected shadows can be achieved in color sequences by understanding how shadows affect color images. This is also useful to achieve a background model that is invariant to illumination changes. Cast shadows has a dark part (umbra) where a light source is totally occluded, and a soft transitional part (penumbra) where light is partially occluded [51]. In visual surveillance scenarios, the penumbra shadows are common since diffused and indirect light is common in indoor and outdoor scenes. Penumbra shadows can be characterized by low value of intensity while preserving the chromaticity of the background, *i.e.*achromatic shadows. Most research on detecting shadows have focused on achromatic shadows [22, 12, 5].

Let us consider the RGB color space, which is a typical output of a color camera. The brightness of a pixel is a linear combination or the RGB channels, here denoted by I

$$I = w_r R + w_g G + w_b B.$$

When an object cast a shadow on a pixel, less light reaches that pixel and the pixel seems darker. Therefore, a shadow casted on a pixel can be characterized by a change of in brightness of that pixel such that

$$\tilde{I} = \alpha I$$

where $\tilde{I}$ is the pixel's new brightness. Similar effect happens under certain changes in illumination, e.g., turning on/off the lights. Here $\alpha < 1$ for the case of shadow, which means the pixel is darker under shadow, while $\alpha > 1$ for the case of highlights, the pixel seems brighter. A change in the brightness of a pixel will affect all the three color channels R, G, and B. Therefore any background model based on the RGB space, and of course gray scale imagery, is bound to detect moving shadows as foreground regions.

So, which color spaces are invariant or less sensitive to shadows and highlights? For simplicity, let us assume that the effect of the change in a pixel brightness is the same in the three channels. Therefore, the observed colors are αR, αG, αB. Any chromaticity measure of a pixel where the effect of the α factor is cancelled, is in fact invariant to shadows and highlights. For example, in [12] chromaticity coordinates based on normalized RGB were used for modeling the background. Given three color variables, R, G and B, the chromaticity coordinates are define as [34]

$$r = \frac{R}{R+G+B}, g = \frac{G}{R+G+B}, b = \frac{B}{R+G+B} \tag{6}$$

Obviously only two coordinates are enough to represent the chromaticity since $r + g + b = 1$. The above equation describes a central projection to the plane $R+G+B = 1$[1]. It can be easily seen that the chromaticity variables r, g, b are invariant to shadows and highlights (according to our assumption) since the α factor does not have an effect on them. Figure 4 shows the results of detection using both (R,G,B) space and (r,g) space. The figure shows that using the chromaticity coordinates allows detection of the target without detecting its shadow.

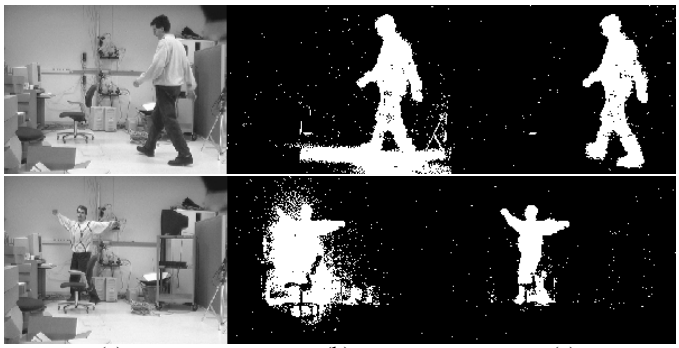
Some other color spaces also have chromaticity variable that are invariant to shadows and highlights in the same way. For example, the reader can verify that the Hue and Saturation variables in the HSV color space are invariant to the α factor and thus insensitive to shadows and highlights, while the Value variable, which represents the brightness is variant to them. Therefore, the HSV color space has been used in some background subtraction algorithms that suppress shadows, *e.g.* [5]. Similarly, HSL, CIE xy spaces have the same property. On the other hand color spaces such as YUV,YIQ, YCbCr are not invariant to shadows and highlights since they are just linear transformations from the RGB space

Although using chromaticity coordinates helps in the suppression of shadows, they have the disadvantage of losing lightness information. Lightness is related to the differences in whiteness, blackness and grayness between different objects [17]. For example, consider the case where the target wears a white shirt and walks against a gray background. In this case there is no color information. Since both white and gray have the same chromaticity coordinates, the target will not be detected

[1] This is analogous to the transformation used to obtain CIE xy chromaticity space from CIE XYZ color space. The CIE XYZ color space is a linear transformation to the RGB space [1]. The chromaticity space defined by the variable r,g is therefore analogous to the CIE xy chromaticity space.

using only chromaticity variables. In fact in the r,g space all the gray line (R=G=B) projects to the point (1/3,1/3) in the space, similarly for CIE xy. Therefore, there is no escape of using a brightness variable! In [12] a third "lightness" variable $s = R+G+B$ was used besides r,g. While the chromaticity variable r,g are not expected to change under shadow, s is expected to change within limits which corresponds to the expected shadows and highlights in the scene.

Most approaches for shadow suppression relies on the above reasoning of separating the chromaticity distortion from brightness distortion where each of these distortions are treated differently, *e.g.* [22, 12, 5, 31, 24] In [22] both brightness and color distortions are defined using a chromatic cylinder model. By projecting an observed pixel color to the vector defined by that pixel's background value in the RGB color space (chromaticity line), the color distortion is defined as the orthogonal distance, while the projection defines the brightness distortion. Here a single Gaussian background model is assumed. These two measures were used to classify an observation to either background, foreground, shadows or highlights. Notice that the orthogonal distance between an observed pixel's RGB color and a chromaticity line is affected by brightness of that pixel, while the distance measured in the r-g space (or xy space) corresponds to the angles between the observed color vector and the chromaticity line, *i.e.*, the r-g space used in [12] is a projection of a chromatic cone. In [24] a chromatic and brightness distortion model is used similar to [22, 31], however using a chromatic cone instead of a chromatic cylinder distortion model.



(a) (b) (c)

Fig. 4 (a) Original frames, (b) Detection using (R,G,B) color space, (c) detection using chromaticity coordinates (r,g) and the lightness variable s.

Another class of algorithms for shadow suppression are approaches that depend on image gradient to model the scene background. The idea is that texture information in the background will be consistent under shadow, hence using the image gradient as a feature will be invariant to cast shadows, except at the shadow boundary. These approaches utilize a background edge or gradient model besides the chromaticity model to detect shadows, *e.g.* [25, 37, 61, 24]. In [24] a multistage approach was proposed to detect chromatic shadows. In the first stage potential shadow region

are detected by fusing color (using the invariant chromaticity cone model described above) and gradient information. In the second stage pixels in these regions are classified using different cues including spatial and temporal analysis of chrominance, brightness, and texture distortion; and a measure of diffused sky lighting denoted by "bluish effect". The approach can successfully detect chromatic shadows.

5 Tradeoffs in Background Maintenance

As discussed in Section 1 there are different changes that can occur in a scene background, which can be classified to: Illumination changes, Motion Changes, Structural Changes. The goal of background maintenance is to be able to cope with these changes and keep an updated version of the scene background model. In parametric background models, recursive update in the form of Eq. 1 (or some variant of it) is typically used for background maintenance, *e.g.* [30, 33, 56]. In nonparametric models, the sample of each pixel history is updated continuously to achieve adaptability [12, 39]. These recursive updates along with careful choice of the color space are typically enough to deal with both the illumination changes and motion changes previously described.

The most challenging case is where changes are introduced to the background (objects moved in or from the background) denoted here by "Structural Changes". For example, if a vehicle came and parked in the scene. A background process should detect such a car but should also adapt it to the background model in order to be able to detect other targets that might pass in front of it. Similarly if a vehicle that was already part of the scene moved out, a false detection 'hole' will appear in the scene where that vehicle was parked. There are many examples similar to thesescenarios. Toyama *et al.* [54] denoted these situations "sleeping person" and "walking person" scenarios.

Here we point out two interwound tradeoffs that associate with maintaing any background model

Background update rate: The speed or the frequency in which a background model gets updated highly influence the performance of the process. In most parametric models, the learning rate α in Eq. 1 controls the speed in which the model adapts to changes. In nonparametric models, the frequency in which new samples are added to the model has the same effect. Fast model update makes the model able to rapidly adapt to scene changes such as fast illumination changes, which leads to high sensitivity in foreground/background classification. However, the model can also adapt to targets in the scene if the update is done blindly in all pixels or errors occurs in masking out foreground regions. Slow update is safer to avoid integrating any transient changes to the model. However, the classifier will lose its sensitivity in case of fast scene changes.

Selective vs. Blind update: Given a new pixel observation, there are two alternative mechanisms to update a background model: 1) Selective Update: update the model only if the pixel is classified as a background sample. 2) Blind Update: just update the model regardless of the classification outcome. Selective update is commonly used by masking out foreground-classified pixels from the update since updating the model with foreground information would lead to increased false negative, *e.g.*, holes in the detected targets. The problem with selective update is that any incorrect detection decision will result in persistent incorrect detection later, which is a deadlock situations, as denoted by Karmann *et al.* [30]. For example, if a tree branch is displaced and stayed fixed in the new location for a long time, it would be continually detected. This is what leads to the 'Sleeping/Walking person' problems as denoted in [54].

Blind update does not suffer from this deadlock situations since it does not involve any update decisions; it allows intensity values that do not belong to the background to be added to the model. This might lead to more false negatives as targets erroneously become part of the model. This effect can be reduced if the update rate is slow.

The interwound effects of these two tradeoffs is shown in table 1. Most background models chose a selective update approach and try to avoid the effects of detection errors by using a slow update rate. However, this is bound to deadlocks. In [12] the use of a combination of two models was proposed: a short-term model (selective and fast) and a long-term model (blind and slow). This combination tries to achieve high sensitivity and, in the same time, avoids deadlocks.

Table 1 Tradeoffs in Background Maintenance

	Fast Update	Slow Update
Selective Update	Highest sensitivity Adapts to fast illumination changes bound to Deadlocks	Less sensitivity bound to Deadlocks
Blind Update	Adapts to targets (more False Negatives) No deadlocks	Slow adaptation No deadlocks

Several approaches have been proposed for dealing with specific scenarios with structural changes. The main problem is that dealing with such changes requires a higher level of reasoning about what are the objects causing such structural changes (vehicle, person, animal) and what should be done with them, which mostly depends on the application. Such high level of reasoning is typically beyond the design goal of the background process, which is mainly a low level process that knows only about pixels' appearance.

The idea of using multiple background models was further developed by Kim *et al.* in [32] to address scene structure changes in an elegant way. In that work, a lay-

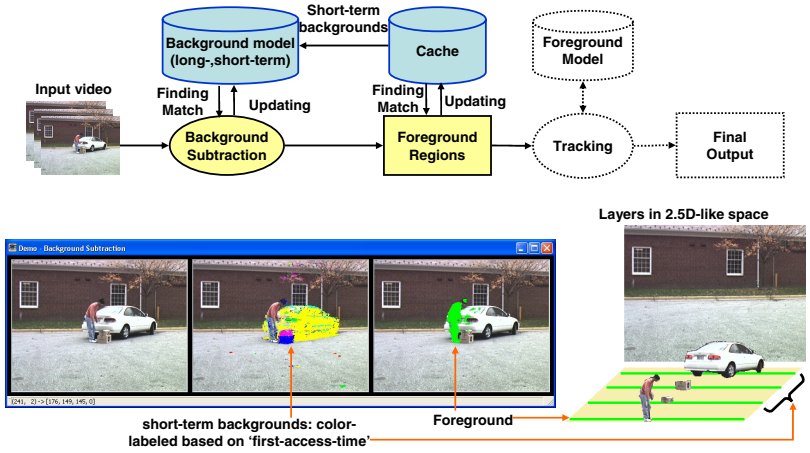


Fig. 5 An overview of Kim *et al.* approach [32] with short-term background layers: the foreground and the short-term backgrounds can be interpreted in a different temporal order.

ered background model was used where a long term background model is used besides several multiple short term background models that capture temporary changes in the background. An object that comes to the scene and stops is represented by a short term background (layer). Therefore, if a second object passes in front of the stopped object, it will also be detected and represented as a layer as well. Figure 5 shows an overview of the approach and detection results.

6 Background Subtraction from a Moving Camera

A fundamental limitation for background subtraction techniques is the assumption of stationary camera. Several approaches have been suggested to alleviate this constraint and develop background subtraction techniques that can work with moving camera under some motion constraints. Rather than a pixel-level representation, a region-based representation of the scene background can help tolerate some degree of camera motion, *e.g.* [42]. In particular, the case of pan-tilt-zoom (PTZ) camera has been addressed because of its importance in surveillance applications. If the camera motion is a rotation with no translation (or close to zero baseline), camera motion can be modeled by a Homography and Image Mosaicing approaches can be used to built a background model. There have been several approaches for building a background model from a panning camera based on building an image mosaic and the use of a MoG model, *e.g.* [46, 38, 44]. Alternatively, in [55] a representation of the scene background as a finite set of images on a virtual polyhedron is used to construct images of the scene background at any arbitrary pan-tilt-zoom setting.

Recently there have been some interests in Background Subtraction/ foreground-background separation from freely moving cameras, *e.g.* [21, 28, 48]. There is a huge literature on motion segmentation [60], which exploits motion discontinuity, however, these approaches do not necessarily aim at modeling scene background and segmenting the foreground layers. Fundamentally motion segmentation by itself is not enough to separate the foreground from the background in case both of them constitutes a rigid or close to rigid motion, *e.g.*, a car parked in the street or a person standing will have the same 3D motion w.r.t. to the camera as the rest of the scene. Similarly depth discontinuity by itself is not enough since objects of interest can be at a distance from the camera with no significant depth difference than the background.

Most notably, Sheikh *et al.* [48] used affine factorization to develop a framework for moving camera background subtraction. In this approach, trajectories of sparse image features are segmented using affine factorization [53]. A sparse representation of the background is maintained by estimating trajectory basis that span the background subspace. KDE was then used to model the appearance of the background and foreground from the sparse features. A Markov Random Field was used to achieve the final labeling.

7 Further Reading

The statistical models for background subtraction that are described in this chapters are basis for many other algorithms in the literature. In [54], linear prediction using the Wiener filter is used to predict pixel intensity given a recent history of values. The prediction coefficients are recomputed each frame from the sample covariance to achieve adaptivity. Linear prediction using the Kalman Filter was also used in [30, 29, 33].

Another approach to model a wide range of variations in the pixel intensity is to represent these variations as discrete states corresponding to modes of the environment, e.g., lights on/off, cloudy/sunny. Hidden Markov Model (HMM) (HMM) have been used for this purpose in [45, 52]. In [45], a three state HMM has been used to model the intensity of a pixel for traffic monitoring application where the three states correspond to the background, shadow, and foreground. The use of HMMs imposes a temporal continuity constraint on the pixel intensity, *i.e.*, if the pixel is detected as a part of the foreground then it is expected to remain part of the foreground for a period of time before switching back to be part of the background. In [52], the topology of the HMM representing global image intensity is learned while learning the background. At each global intensity state the pixel intensity is modeled using a single Gaussian. It was shown that the model is able to learn simple scenarios like switching the lights on-off .

Intensity has been the most commonly used feature for modeling the background. Alternatively, edge features have also been used to model the background. The use of edge features to model the background is motivated by the desire to have a rep-

resentation of the scene background that is invariant to illumination changes, as discussed in Section 4. In [59] foreground edges are detected by comparing the edges in each new frame with an edge map of the background which is called the background "primal sketch". The major drawback of using edge features to model the background is that it would only be possible to detect edges of foreground objects instead of the dense connected regions that result from pixel intensity based approaches. Fusion of intensity and edge information was used in [25, 37, 61, 24]. Among many other feature studied, Optical Flow was used in [39] to help capture background dynamics. A general framework for feature selection based on boosting for background modeling was proposed in [41].

Besides pixel-based approaches, block-based approaches have also been used for modeling the background. Block matching has been extensively used for change detection between consecutive frames. In [23] each image block is fit to a second order bivariate polynomial and the remaining variations are assumed to be noise. A statistical likelihood test is then used to detect blocks with significant change. In [36] each block was represented with its median template over the background learning period and its block standard deviation. Subsequently, at each new frame, each block is correlated with its corresponding template and blocks with too much deviation relative to the measured standard deviation are considered to be foreground. The major drawback with block-based approaches is that the detection unit is a whole image block and therefore they are only suitable for coarse detection.

Background subtraction techniques can successfully deal with quasi moving background, *e.g.*scenes with dynamic textures. The nonparametric model using Kernel Density Estimation (KDE), described in Section 3.2, has very good performance in scenes with dynamic backgrounds, such as outdoor scenes with trees in the background. Several approaches were developed to address such dynamic scenes. In [50] an Auto Regressive Moving Average Model (ARMA) (ARMA) model was proposed for modeling dynamic textures. ARMA is a first order linear prediction model. In [62] an ARMA model was used for background modeling of scenes with dynamic texture where a robust Kalman filter was used to update the model. In [39] a combination of optical flow and appearance features was used within an adaptive kernel density estimation framework to deal with dynamic scenes.

References

1. Wilhelm Burger and Mark Burge. *Digital Image Processing, an Algorithmic Introduction Using Java*. Springer, 2008.
2. Yizong Cheng. Mean shift, mode seeking, and clustering. *IEEE Transaction on Pattern Analysis and Machine Intelligence*, 17(8):790–799, Aug 1995.
3. Dorin Comaniciu. *Nonparametric Robust Methods For Computer Vision*. PhD thesis, Rutgers, The State University of New Jersey, January 2000.
4. Dorin Comaniciu and Peter Meer. Mean shift analysis and applications. In *IEEE 7th International Conference on Computer Vision*, volume 2, pages 1197–1203, Sep 1999.

5. Rita Cucchiara, Costantino Grana, Massimo Piccardi, and Andrea Prati. Detecting moving objects, ghosts, and shadows in video streams. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25:1337–1342, 2003.
6. A. Dempster, N. Laird, and D. Rubin. Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society*, 39:1–38, 1977.
7. Richard O. Duda, David G. Stork, and Peter E. Hart. *Pattern Classification*. Wiley, John & Sons,, 2000.
8. H.J. Eghbali. K-s test for detecting changes from landsat imagery data. *SMC*, 9(1):17–23, 1979.
9. Ahmed Elgammal. *Efficient Kernel Density Estimation for Realtime Computer Vision*. PhD thesis, University of Maryland, 2002.
10. Ahmed Elgammal, Ramani Duraiswami, and Larry S. Davis. Efficient non-parametric adaptive color modeling using fast gauss transform. In *Proc. of IEEE Conference on Computer Vision and Pattern Recognition*, Dec 2001.
11. Ahmed Elgammal, Ramani Duraiswami, David Harwood, and Larry S. Davis. Background and foreground modeling using non-parametric kernel density estimation for visual surveillance. *Proceedings of the IEEE*, July 2002.
12. Ahmed Elgammal, David Harwood, and Larry S. Davis. Nonparametric background model for background subtraction. In *Proc. of 6th European Conference of Computer Vision*, 2000.
13. David A. Forsyth and Jean Ponce. *Computer Vision a Modern Approach*. Prentice Hall, 2002.
14. Nir Friedman and Stuart Russell. Image segmentation in video sequences: A probabilistic approach. In *Uncertainty in Artificial Intelligence*, 1997.
15. K. Fukunaga and L.D. Hostetler. The estimation of the gradient of a density function, with application in pattern recognition. *IEEE Transaction on Information Theory*, 21:32–40, 1975.
16. Xiang Gao and T.E. Boult. Error analysis of background adaption. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2000.
17. Ernest L. Hall. *Computer Image Processing and Recognition*. Academic Press, 1979.
18. Bohyung Han, Durin Comaniciu, and Larry Davis. Sequential kernel density approximation through mode propagation: Applications to background modeling. In *In Proc. ACCV 2004*, 2004.
19. Ismail Haritaoglu, David Harwood, and Larry S. Davis. W4:who? when? where? what? a real time system for detecting and tracking people. In *International Conference on Face and Gesture Recognition*, 1998.
20. Michael Harville. A framework for high-level feedback to adaptive, per-pixel, mixture-of-gaussian background models. In *ECCV*, pages 543–560, 2002.
21. Eric Hayman and Jan olof Eklundh. Statistical background subtraction for a mobile observer. In *In Proceedings ICCV*, pages 67–74, 2003.
22. Thanarat Horprasert, David Harwood, and Larry S. Davis. A statistical approach for real-time robust background subtraction and shadow detection. In *IEEE Frame-Rate Applications Workshop*, 1999.
23. Y. Z. Hsu, H. H. Nagel, and G. Rekers. New likelihood test methods for change detection in image sequences. *Computer Vision and Image Processing*, 26:73–106, 1984.
24. I. Huerta, M. Holte, T. Moeslund, and J. Gonzalez. Detection and removal of chromatic moving shadows in surveillance scenarios. pages 1499–1506, 2009.
25. Sumer Jabri, Zoran Duric, Harry Wechsler, and Azriel Rosenfeld. Detection and location of people in video images using adaptive fusion of color and edge information. In *International Conference of Pattern Recognition*, 2000.
26. R.C. Jain and H.H. Nagel. On the analysis of accumulative difference pictures from image sequences of real world scenes. *PAMI*, 1(2):206–213, April 1979.
27. Omar Javed, Khurram Shafique, and Mubarak Shah. A hierarchical approach to robust background subtraction using color and gradient information. In *in IEEE Workshop on Motion and Video Computing*, pages 22–27, 2002.
28. Y.X. Jin, L.M. Tao, H. Di, N.I. Rao, and G.Y. Xu. Background modeling from a free-moving camera by multi-layer homography algorithm. In *ICIP*, pages 1572–1575, 2008.

29. Klaus-Peter Karmann, Achim V. Brandt, and Rainer Gerl. Moving object segmentation based on adabtive reference images. In *Signal Processing V: Theories and Application*. Elsevier Science Publishers B.V., 1990.
30. Klaus-Peter Karmann and Achim von Brandt. Moving object recognition using and adaptive background memory. In *Time-Varying Image Processing and Moving Object Recognition*. Elsevier Science Publishers B.V., 1990.
31. Kyungnam Kim, Thanarat H. Chalidabhongse, David Harwood, and Larry Davis. Background modeling and subtraction by codebook construction. In *In International Conference on Image Processing*, pages 3061–3064, 2004.
32. Kyungnam Kim, David Harwood, and Larry S. Davis. Background updating for visual surveillance. In *In Proceedings of the International Symposium on Visual Computing*, pages 1–337, 2005.
33. D. Koller, J. Weber, T.Huang, J.Malik, G. Ogasawara, B.Rao, and S.Russell. Towards robust automatic traffic scene analyis in real-time. In *International Conference of Pattern Recognition*, 1994.
34. Martin D. Levine. *Vision in Man and Machine*. McGraw-Hill Book Company, 1985.
35. L. Maddalena and A. Petrosino. A self-organizing approach to background subtraction for visual surveillance applications. 17(7):1168–1177, July 2008.
36. T. Matsuyama, Takashi Ohya, and Hitoshi Habe. Background subtraction for nonstationary scenes. In *4th Asian Conference on Computer Vision*, 2000.
37. Stephen J. Mckenna, Sumer Jabri, Zoran Duric, Harry Wechsler, and Azriel Rosenfeld. Tracking groups of people. *Computer Vision and Image Understanding*, 80:42–56, 2000.
38. Anurag Mittal and Dan Huttenlocher. Scene modeling for wide area surveillance and image synthesis. In *CVPR*, 2000.
39. Anurag Mittal and Nikos Paragios. Motion-based background subtraction using adaptive kernel density estimation. In *CVPR*, pages 302–309, 2004.
40. Radford M. Neal and Geoffrey E. Hinton. A new view of the em algorithm that justifies incremental and other variants. In *Learning in Graphical Models*, pages 355–368. Kluwer Academic Publishers, 1993.
41. Toufiq Parag, Ahmed Elgammal, and Anurag Mittal. A framework for feature selection for background subtraction. In *Proc. IEEE Computer Society Conference on Computer Vision and Pattern Recognition, CVPR06*, June 2006.
42. Kedar Patwardhan, Guillermo Sapiro, and Vassilios Morellas. Robust foreground detection in video using pixel layers. *IEEE Trans. Pattern Anal. Mach. Intell.*, 30:746–751, April 2008.
43. M. Piccardi and T. Jan. Mean-shift background image modelling. In *ICIP*, pages V: 3399–3402, 2004.
44. Ying Ren, Chin-Seng Chua, and Yeong-Khing Ho. Statistical background modeling for non-stationary camera. *Pattern Recogn. Lett.*, 24(1-3):183–196, 2003.
45. Jens Rittscher, J. Kato, S. Joga, and Andrews Blake. A probabilistic background model for tracking. In *6th European Conference on Computer Vision*, 2000.
46. Simon Rowe and Andrew Blake. Statistical mosaics for tracking. *Image and Vision Computing*, 14(8):549–564, 1996.
47. David W. Scott. *Mulivariate Density Estimation*. Wiley-Interscience, 1992.
48. Y. Sheikh, O. Javed, and T. Kanade. Background subtraction for freely moving cameras. In *ICCV*, pages 1219–1225, 2009.
49. Yaser Sheikh and Mubarak Shah. Bayesian modeling of dynamic scenes for object detection. *PAMI*, 27:1778–1792, 2005.
50. S. Soatto, G. Doretto, and Y. Wu. Dynamic textures. In *Proceedings of the International Conference on Computer Vision*, volume 2, pages 439–446, 2001.
51. Jrgen Stauder, Roland Mech, and Jrn Ostermann. Detection of moving cast shadows for object segmentation. *IEEE Transactions on Multimedia*, 1:65–76, 1999.
52. B. Stenger, V. Ramesh, N. Paragios, F. Coetzee, and J. Bouhman. Topology free hidden markov models: Application to background modeling. In *IEEE International Conference on Computer Vision*, 2001.

53. Carlo Tomasi. Shape and motion from image streams under orthography: a factorization method. *International Journal of Computer Vision*, 9:137–154, 1992.
54. Kentaro Toyama, John Krumm, Barry Brumitt, and Brian Meyers. Wallflower: Principles and practice of background maintenance. In *IEEE International Conference on Computer Vision*, 1999.
55. T. Wada and T. Matsuyama. Appearance sphere: Background model for pan-tilt-zoom camera. In *13th International Conference on Pattern Recognition*, 1996.
56. W.E.L.Grimson and C.Stauffer. Adaptive background mixture models for real-time tracking. In *IEEE Conference on Computer Vision and Pattern Recognition*, 1999.
57. W.E.L.Grimson, C.Stauffer, and R.Romano. Using adaptive tracking to classify and monitor activities in a site. In *IEEE Conference on Computer Vision and Pattern Recognition*, 1998.
58. Christopher Richard Wern, Ali Azarbayejani, Trevor Darrell, and Alex Paul Pentland. Pfinder: Real-time tracking of human body. *IEEE Transaction on Pattern Analysis and Machine Intelligence*, 1997.
59. Yee-Hong Yang and Martin D. Levine. The background primal sketch: An approach for tracking moving objects. *Machine Vision and Applications*, 5:17–34, 1992.
60. Luca Zappella, Xavier Lladó, and Joaquim Salvi. Motion segmentation: a review. In *Proceeding of the 2008 conference on Artificial Intelligence Research and Development*, pages 398–407, Amsterdam, The Netherlands, The Netherlands, 2008. IOS Press.
61. Wei Zhang, Xiang Zhong Fang, and Xiaokang Yang. Moving cast shadows detection based on ratio edge. In *International Conference on Pattern Recognition (ICPR)*, 2006.
62. Jing Zhong and Stan Sclaroff. Segmenting foreground objects from a dynamic textured background via a robust kalman filter. In *ICCV '03: Proceedings of the Ninth IEEE International Conference on Computer Vision*, page 44, Washington, DC, USA, 2003. IEEE Computer Society.

Glossary

Auto Regressive Moving Average Model (ARMA) Given a time series of data the ARMA model is a tool for understanding and, perhaps, predicting future values in this series. The model consists of two parts, an autoregressive (AR) part and a moving average (MA) part.. 18

Expectation Maximization In statistics, an expectation-maximization (EM) algorithm is a method for finding maximum likelihood or maximum a posteriori (MAP) estimates of parameters in statistical models, where the model depends on unobserved latent variables. 5

Hidden Markov Model (HMM) is a statistical Markov model in which the system being modeled is assumed to be a Markov process with unobserved (hidden) states. An HMM can be considered as the simplest dynamic Bayesian network.. 17

Homography A homography is an invertible transformation from the real projective plane to the projective plane that maps straight lines to straight lines.. 16

Image Mosaicing stitching several overlapping images on a surface, *e.g.*a plane to construct a combined image.. 16

K-means A statistical clustering method which aims to partition observations into k clusters in which each observation belongs to the cluster with the nearest mean.. 6

Kalman Filter In statistics, the Kalman filter is a mathematical method named after Rudolf E. Kalman. Its purpose is to use measurements observed over time, containing noise (random variations) and other inaccuracies, and produce values that tend to be closer to the true values of the measurements and their associated calculated values.. 17

Markov Random Field A Markov random field, Markov network or undirected graphical model is a graphical model in which a set of random variables have a Markov property described by an undirected graph. . 17

Mixture of Gaussians A statistical mixture model where a distribution is approximated with a combination of Gaussian distributions. 5

Optical Flow Optical flow or optic flow is the pattern of apparent motion of objects, surfaces, and edges in a visual scene caused by the relative motion between an observer (an eye or a camera) and the scene. 18