# Efficient Representation of Distributions for Background Subtraction*

Yedid Hoshen Chetan Arora Yair Poleg Shmuel Peleg
The Hebrew University of Jerusalem
Jerusalem, Israel

## Abstract

*Multi dimensional probability distributions are used in many surveillance tasks such as modeling color distribution of background pixels for Background Subtraction. Accurate representation of such distributions, e.g. in a histogram, requires much memory that may not be available when a histogram is computed for each pixel. Parametric representations such as Gaussian Mixture Models (GMM) are very efficient in memory but may not be accurate enough when the distribution is not from the assumed model.*

*We propose a memory efficient representation for distributions. Histograms cells usually have equal width, and count the hits in each cell (Equi-width histograms). In most cases a 1D distribution can be represented more efficiently when cell sizes change so that each cell will have same number of hits (Equi-depth histograms). We propose to describe compactly multi-dimensional distributions (e.g. color) using an equi-depth histograms. Online computation of such histograms is described, and examples are given for background subtraction.*

## 1. Introduction

Histograms are present everywhere in computer vision, with uses ranging from background modeling in video surveillance [8, 12], feature representations [11], and natural image modeling [9]. Histograms are very popular since they make very few assumptions on the probability density function (PDF) which they attempt to model. The only assumption made is on the range of measurements, which together with the requirements on the quantization error determine the number of bins in the histogram. Histograms can therefore model very general probability densities. Another popular approach to model PDF is Kernel Density Estimation (KDE) [14, 3], a technique also known as Parzen windows. This method stores a small number of data values in memory. The probability of any given value is the sum of kernel distances from each of the samples. Both equi-width histograms and KDE will converge to the correct probability, but an accurate estimate might require many histogram bins or many stored data points. This problem becomes especially unmanageable in higher dimensions, in which the curse of dimensionality comes into play.

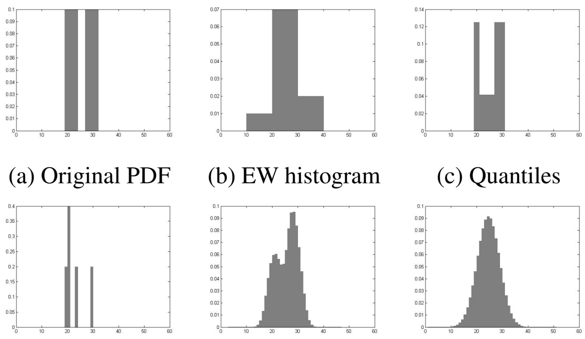
Much research on histogram methods has been carried out by the database community, where several new histogram types were suggested [6] with different performance guarantees, construction costs, and lookup costs. The approach most relevant to our needs concerns stream methods (Online methods in computer algorithms terminology) that update the histogram variables at run-time without storing all data points.

In order to reduce memory requirements, semi-parametric methods were proposed. An example of these methods is GMM [16]. This method assumes that the probability density can be represented by a certain number of clusters each represented by the Gaussian distribution. The number of clusters as well as the means and covariance matrices of each cluster are parameters to be estimated from the data. This scheme works well when the assumption of Gaussian clusters is correct, but might give poor results otherwise . Another way to reduce memory needs is by designing application specific semi-parametric models [7] that fit better the target distribution or the application objective; however it is not always obvious how to do so.

Most current approaches for background subtraction model the probability distribution of various features (such as color, texture[19], etc.) at each pixel in the background, and use this model to determine if a current values is likely to correspond to a background or a foreground object. As there are on the order of a million pixels in an image, and memory is limited, a memory efficient representation for histograms is essential. As we will show, covariance between the multi-dimensional features is important for accurate foreground detection and a multi-dimensional representation gives a more accurate segmentation. Current multi-dimensional models use either semi-parametric methods such as GMM [16] or non-probabilistic modeling methods such as a codebook [7]. We show an application of our method that provides the advantages of the truly non-parametric model of histogram methods without suffering from its unmanageable memory requirements.

*This research was supported by the Israel Science Foundation.

978-1-4799-0703-8/13/$31.00 ©2013 IEEE



(a) Original PDF (b) EW histogram (c) Quantiles

(d) KDE $\omega = 0.1$ (e) KDE $\omega = 2.0$ (f) GMM

Figure 1: Comparing different representations for a probability distribution. a) Original distribution. b) Equi-width Histogram with 5 bins. c) Equi-depth histogram with 5 quantiles. d) KDE, 5 Gaussian kernels, bandwidth = 0.1. e) KDE, 5 Gaussian kernels, bandwidth = 2.0. f) GMM using 2 Guassians.

The contributions of this paper address the following issues: (1) A memory efficient multi-dimensional histogram structure, and its online computation. (2) A generic memory efficient solution for improving background models. We hope that this work would showcase the utility of modern histogram techniques in surveillance applications.

### 1.1. Comparison of PDF Estimation Approaches

In normal histograms (i.e. equi-width histograms) all bins have an equal size, and the number of occurrences within each bin is counted. Equi-depth histograms, on the other hand, model the distribution by keeping the values of several quantiles $Q_{\frac{i}{N}}, i = (0..N)$. $Q_p$ is the p-th quantile of the distribution, the smallest sample which is larger than $(100 \cdot p)\%$ of the observed samples. For example the median is $Q_{0.5}$. The probability of each segment between two consecutive quantiles is $1/N$, and the probability density is lower as the segement becomes wider. A comparison of several approaches to the estimation of PDF are shown in Fig. 1. It is shown that for low entropy distributions an equi-depth histograms as in Fig. 1(c) can yield better probability estimates with less memory. However, the online calculation of an equi-depth histogram is not trivial without storing all data values. Another issue is that there is no consensus how to generalize quantiles to multiple dimensions.

Online quantile estimations are treated both in statistics and in the stream data literature (where "stream" is used for "online"). Various types of multi-dimensional histogram were proposed. Our method uses a multiple dimensional representation which is close to that of Cormode et.al. [2], and a statistical stream update method which has a very low memory requirement and fast lookup time.

## 2. Related work

**Non-parametric models**: A good survey on histogram techniques can be found in [6]. Several approaches have been presented for stream (online) quantile representation such as the Greenwald-Khanna [4] algorithm. Extensions of stream methods were proposed for multiple dimensions using approaches such as Sketching [17]. These methods are not appropriate for background distributions, and require more memory than commonly available. An approach for the multi-dimensional case similar to ours in presented in Cormode et. al. [2], but it too requires much memory. Several approaches were presented [1, 15] for online estimation of quantiles, with required memory equal to the number of quantiles. Our method leverages the statistical quantile estimator methods [1] to achieve a more efficient descriptor than Cormode et al [2].

**Background subtraction**: A popular approach for background subtraction computes a background probability model at each pixel. For every new frame, the model for each pixel is updated in an online manner. The probability density function (PDF) may depend on many possible features such as color, texture [19], gradient [5] and optical flow[13]. For non-parametric one-dimensional probability representation it is common to use either a histogram [8, 12] or KDE [3]. The problem with both methods is that they scale badly with dimension and thus cannot be used to model feature dependence. The Gaussian Mixture Models (GMM) is suitable for multi-dimensional distributions but suffers from the Gaussian assumption and is required to prespecify the number of modes in the distribution. This is a particularly challenging issue as choosing too many modes can be as bad as choosing too few, due to the sensitive online update rule. Several attempts have been made to find a good heuristic update rule, (for example [10]) however no rigorous update rule is known. Another way to avoid the curse of dimensionality is to rely on non-probabilistic models with heuristic similarity measures (such as the codebook in [7]), however these methods require specifying heuristics and non-intuitive parameters and are thus not naturally extended to include different features.

Another approach is to neglect dependence between variables altogether and thus avoid the curse of dimensionality. In [5] SVM is used to try to learn the dependency between variables as a means to avoid modeling the conditional probability. This approach fails to work when the dependency is not global but changes in different scene contexts.

Our approach is different from the above by its ability to scale up to multiple-dimensions whilst using a general probabilistic model and making very few assumptions on the data distribution. We do not have to know in advance the number of modes, nor the shape of the distribution.

# 3. QuantileGrid representation of distribution

## 3.1. The problem with histogram methods

Equi-width histograms represent a quantized representation of the distribution. Using a sufficient number of bins is essential for a good representation. As it is not easy to know a-priori the range at which interesting behavior will happen, histograms will require a large number of bins at a high memory cost. In multiple dimensions the required number of bins scales exponentially. This is known as the curse of dimensionality, as for even a fairly moderate dimension the required number of bins would explode. These histograms are therefore unsuitable for representation of multi-dimensional distributions for each pixel.

The relative advantage of equi-depth histograms for modeling probability distributions is the increased resolution in regions with highest concentration of samples, and less bins in regions with fewer samples. Another advantage of equi-depth histograms is that the scale of the interesting behavior does not have to be pre-specified but is dynamically determined by the data. The representation wastes no bins in regions that contain no samples.

There are several challenges with equi-depth histograms. Of primary concern to our work are finding online (stream) update rules, and extending the representation to multiple dimensions. An online update rule is essential for surveillance video applications. Offline quantile computation strategy would require storing all the data samples and sorting them. In surveillance settings we do not have the memory capacity to store all data, and an online solution is required. Another challenge is the multi-dimensional representation. Quantiles are well defined only in 1D, and a suitable definition is required in higher dimensions. An example for such definition is Skyline descriptors [2]. In our solution we break away from using the global quantiles as we shall show in the next subsection.

Our proposed solution, 'QuantileGrid', resolves both issues having extremely low memory requirements. We shall first address the online update rule.

## 3.2. An online quantile update rule

In the statistics literature [15] estimators are suggested whose memory size equals the number of quantiles. Let a PDF be described by $N+1$ quantiles $\{Q_{q_i}\}$ where $q_i = \frac{i}{N}$, $i = (0..N)$. By definition, given $L$ samples we expect $q_i \cdot L$ of them to be smaller than $Q_{q_i}$, and $(1-q_i) \cdot L$ to be larger.

The online update rule is as follows: When a new sample $S$ arrives, it is compared to each of the quantiles. If $S > Q_{q_i}$, $Q_{q_i}$ is increased by $q_i \cdot C$ ($C$ is a pre-determined constant), and if $S < Q_{q_i}$, $Q_{q_i}$ is decreased by $(1-q_i) \cdot C$. Equilibrium of this process occurs when $q$ of the samples are smaller than $Q_q$ and $(1-q)$ of the samples are larger than $Q_q$ [1]. A large constant $C$ contributes to faster convergence, while a small $C$ gives a more stable tracking of distributions. A possible constant $C$ for grey level histograms can start with $C_0 = 1$, decrease by a factor of $0.977$ at every additional sample, reaching the minimal value of $C_{min} = 0.01$ after 200 samples, and staying with this value for all following samples.

There are several possible approaches to determine the initial quantiles. One approach is to start by dividing the range (when known) into equal sized cells. Another approach is to take the first $N+1$ different samples, sort them, and use as the initial quantiles. This update rule for quantile computation has the minimal memory requirements for quantile tracking while still preserving a good estimate for the quantile position.

## 3.3. Multi-dimensional QuantileGrid

Various approaches have been suggested for multi-dimensional data representation, but most of them cannot be queried at every time step, can not be updated online, and use extensive memory. Our method is similar to that presented at [2], but needs less memory due to the use of quantile estimators. We estimate the joint probability using 1D quantiles, and we present two different representations that are useful under different cases. For clarity, the explanation is given only for 2D, and the extension to any dimension is similar.

**Joint probability matrix**

The joint probability matrix, whose 2D example is shown in Fig. 2.a, uses 1D quantiles computed as in the previous section for each variable using its marginal distribution. A 2 dimensional array is created where each axis has its own quantiles, giving cells having variable sizes (as in Fig. 2.a). The matrix value corresponding to each cell gives the number of samples that fall in this cell. The total memory cost of this array is $N^2 + 2(N+1)$.

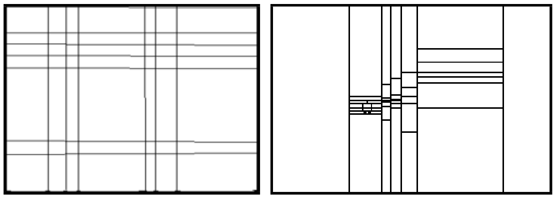
The online update process of this matrix, given a new sample, is as follows:

1. The counter of the cell corresponding to the new sample is incremented by 1.

2. The quantiles in all dimensions are updated according to the 1D update. This creates a new array of cells, which is slightly different than the previous array.

3. The counters of every cell in the new array are computed from the area of overlap with cells in the previous array. For example, if the new cell covers 50% of the area of a previous cell with count 30, and 30% of the area of a previous cell with count 10, the count of the new cell with be $0.5 \cdot 30 + 0.3 \cdot 10 = 18$.

4. The PDF at each cell is the count at the cell divided by the total counts and the area of the cell.



(a) Joint probability matrix (b) Adaptive Grid

Figure 2: Two approaches for 2D quantile grids.

The approximation of the joint probability matrix to the true joint probability improves as the 1D quantiles converge and becomes stable. The cost of the update step is linear in the dimension and number of quantiles $O(d * q)$.

**Adaptive grid**

The adaptive grid is presented to better take into account the joint probability between variables. A 2D example can be seen in Fig. 2.b. Assume that variables are sorted by order of increased marginal entropy. The data structure is initialized by examining some samples, and computing 1D quantiles for the first variable. Within each bin of the first variable, 1D quantiles are computed for the second variable.

Given a new sample, the update steps are as follows:

1. Update the quantiles of the first variable using the new sample.
2. Find the bin of the first variable in which the new sample falls.
3. Update the quantiles of the second variable within the respective bin of the first variable.

Since in the adaptive grid all bins have on average an equal number of hits, we do not need to store the counts in each cell, only the positions of the quantiles are needed. The PDF in each cell is inverse to its area. The cost of the update step is linear in the dimension and number of quantiles $O(d * q)$.

**Features of QuantileGrid**

Both representations have the same memory requirements. The joint matrix has the advantage of greater stability but lower expressiveness. The adaptive grid has the advantage of encoding dependence at an earlier stage at the price of longer convergence times.

Both representations work in real time, our unoptimized C++ implementation operates at 15-20 fps, with 5 quantiles in 2D.

The joint matrix representation is extended to higher dimensions by computing 1D quantiles for each dimension and creating a $d$-dimensional tensor for the counts. The adaptive grid representation is extended to higher dimensions by preforming the above procedure recursively, splitting every cell created using the first $d$ dimensions into quantiles using the $(d + 1)st$ dimension.

| | **32*32 Histogram** | **6*6 QuantileGrid** |
|---|---|---|
| $L_1$ **Difference** | 0.82 | 0.77 |
| $L_2$ **Difference** | 0.05 | 0.03 |

Table 1: Comparing accuracy of quantized histogram to QuantileGrid by computing the difference from ground truth. The difference is computed using the sum of elementwise L1 and L2 norms. A 2D QuantileGrid with 36 memory cells is more accurate than a 2D histogram with 1024 memory cells.

QuantileGrid is a non-parametric representation that makes only a few assumptions about the data. The main assumption is that the data has some structure. Also, there should be at least twice as many quantiles as there are peaks. Having too many quantiles will not hurt accuracy.

We also assume the distribution of colors at each pixel is changing slowly, and therefore we can base our matrix or grid update rules on previous quantile estimates, this is a valid assumption for most practical scenarios.

### 3.4. Improvement over equi-width histograms

We compare the distributional accuracy of QuantileGrid with that of Equi-Width histograms for a synthetic case. The true 2D distribution has 8 equal height, non zero values in two well-separated clusters. We estimated the distribution by representing its samples using 3 approaches. i) A full 255 × 255 equi-width histogram (considered as the ground truth). ii) A 32×32 equi-width histogram. iii) QuantileGrid with 5 quantiles per dimension (using 36 memory locations). We smoothed the ground-truth histogram with a Gaussian kernel ($\sigma = 1.5$) to compensate for slight shifts. We then compared the smoothed ground truth vs. the $32 \times 32$ histogram and the QuantileGrid using elementwise L1 and L2 norms. The results are displayed in Table. 1, and show that in this case a 2D QuantileGrid with 36 memory cells was more accurate than a 2D histogram with 1024 memory cells

We can see that although using just 3.6% of the amount of memory used by the 2D histogram, QuantileGrid still provided a better estimate of the distribution. This is due to its better spatial resolution at the critical places.

## 4. Application to background subtraction

Background Subtraction is the first step in many surveillance applications. Its objective is to separate the static background from the moving foreground in a video taken by a static camera. Most current approaches for background subtraction operate at the pixel level, processing one frame at a time. A model describing the typical background behavior is computed for each pixel. If this model gives to

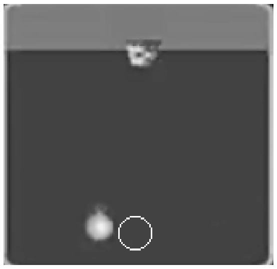


Figure 3: A single frame from the pendulum video. In the video a pendulum is swinging periodically, until a white hoop suddenly appears. The hoop is static and has similar intensity as the pendulum.

a new sample a low probability of belonging to the background, it is labeled as foreground.

Due to the aforementioned memory issues, the model must be specified with only a small number of values. This forces the model designer to make a choice between using non-parametric 1D models do not model the joint distribution, or use less stable and semi-parametric multi-dimensional methods such as GMM.

We offer a new solution that is both non-parametric, multi-dimensional and uses little memory. Strongly dependent features benefit the most from joint modeling. To illustrate this concept, let us suppose that the probability of background is determined by two binary factors, intensity and optical flow. As shown in Fig. 3, the background contains both a dark static background and a fast moving bright pendulum. The joint probability density of intensity and optical flow is showing high probabilities for either dark and static or bright and moving:

$$P(I, OF|Background = true) = \begin{pmatrix} 1.0 & 0.0 \\ 0.0 & 1.0 \end{pmatrix}.$$

The 1D marginal probabilities based on either intensity or optical flow are uniform: $P(I|Background = true) = (0.5, 0.5)$ and $P(OF|Background = true) = (0.5, 0.5)$.

In this case the 1D representation will not be able to separate foreground from background. A joint representation however, would be able to point out dark static pixels and bright moving pixels as background. There is very little probability that for other moving objects this dependency would be observed. A global SVM learning of the correlations such as suggested by [5] will therefore not be effective here.

The scenario described above is shown in Fig. 3. In the video a pendulum is swinging periodically, until a white hoop suddenly appears. The hoop is static and has similar intensity as the pendulum. Fig 4.a shows the minimum of the two 1D probabilities ($P(I|Foreground)$ and $P(OF|Foreground)$, and Fig 4.b shows the joint probability $P(I, OF|Foreground)$. While both cases show some compression artifacts as foreground, only the joint probability indicates the hoop as foreground. This is due to the greater expressiveness of the 2D representation.

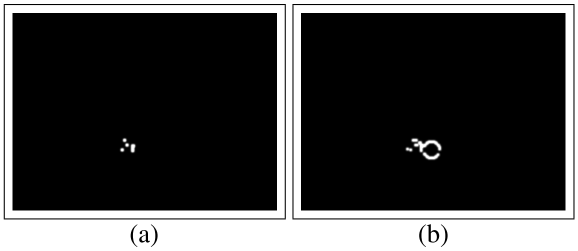
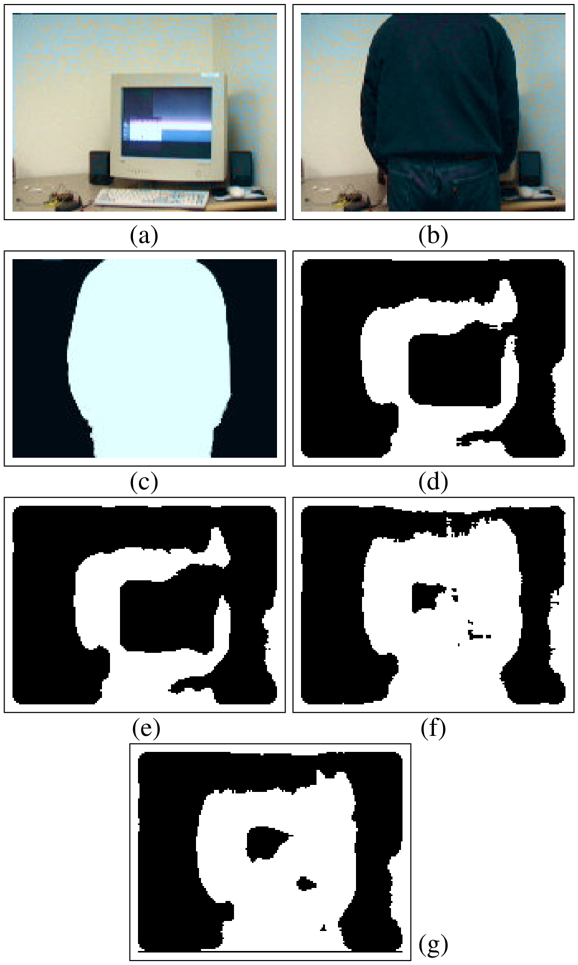


Figure 4: Background Probability given by various models. a) Minimum of two 1D models. b) 2D Joint Probability. The hoop is missing by the 1D models, but is detected by the 2D model.

Another example is shown in Fig. 5. The video is a modified version of the Camouflage sequence in the Wallflower dataset [18]. In the original sequence the clothes and screen are well separated and can be detected by intensity alone. We changed the color of the clothes to be more similar to the screen in the H and S color components to make the example more challenging. More accurately, the screen is flickering with rolling bars between two modes with H, S values of(140, 120) and (155,110) as shown in Fig. 5.a. A person, whose cloths have H, S values of (140,110), enters the scene and blocks the screen from the camera (Fig. 5.b).

We analyze this scene using the H and S color components of the HSV domain. For comparison, 1D quantile histograms are computed for both the H and S components, 2D GMM, and 2D QuantileGrid (in the joint matrix representation). Foreground detection is shown in Fig. 5 using (d) two 1D histograms ($min(P(H), P(S))$), (e) GMM, and (f-g) $P(H, S)$ using QuantileGrid (using 5 quantiles in each dimension, 36 memory cells in total). In all cases result were smoothed by a 9*9 median filter, as used in recent papers.

It can be observed that using two 1D projections of the probability failed in the screen region. This is because both the H and S color components are individually probable. GMM has not preformed well, due to the effect of strong HSV noise and flicker on its update scheme. QuantileGrid, on the other hand, detected the foreground object well, even in front of the screen. This is because the combination of the two components is jointly unlikely. As before, this is a local dependence and could not be learned globally. The computer screen could just as easily have exhibited the opposite dependence relation between the color channels. This demonstrates the importance of QuantileGrid in modeling pixel-level multi-dimensional distributions with very little memory.



(a) (b) (c) (d) (e) (f) (g)

Figure 5: Segmentation of the camouflage sequence using the 2D H,S color components. a) The background is a flickering computer screen. b) The foreground is a person standing in front of the screen. c) Ground Truth segmentation. d) Minimum of the two 1D probabilities (thresholded at 0.1). The screen region is not detected. e) GMM has not detected the screen region. f-g) The screen region is well detected by QuantileGrid Matrix (f) and Adaptive Grid (g) (thresholded at 0.1).

## 5. Concluding Remarks

In this work we proposed a new approach to probability density representation. A highly memory-efficient data structure, QuantileGrid, was presented for online estimation of multi-dimensional probability distributions in a truly non-parametric way. An application to background subtraction was presented, and showed the benefit of the method over existing methods: both 1D methods and GMM. QuantileGrid can be used also to collect high dimensional web-scale pixel-level natural image statistics, with applications to scene priors and recognition.

## References

[1] T. Bylander and B. Rosen. A perceptron-like online algorithm for tracking the median. In *Int. Conf. on Neural Networks*, volume 4, pages 2219–2224, 1997.

[2] G. Cormode, F. Korn, S. Muthukrishnan, and D. Srivastava. Summarizing two-dimensional data with skyline-based statistical descriptors. In *Proc. of the 20th int. conf. on Scientific and Statistical Database Management*, pages 42–60, 2008.

[3] A. Elgammal, D. Harwood, and L. Davis. Non-parametric model for background subtraction. In *ECCV*, pages 751–767, 2000.

[4] M. Greenwald and S. Khanna. Space-efficient online computation of quantile summaries. In *ACM SIGMOD*, pages 58–66, 2001.

[5] B. Han and L. Davis. Density-based multifeature background subtraction with support vector machine. *IEEE Transactions on PAMI*, 34(5):1017–1023, 2012.

[6] Y. Ioannidis. The history of histograms (abridged). In *Proc. of the 29th int. conf. on Very large data bases*, volume 29, pages 19–30, 2003.

[7] K. Kim, T. Chalidabhongse, D. Harwood, and L. Davis. Background modeling and subtraction by codebook construction. In *ICIP*, volume 5, pages 3061–3064, 2004.

[8] T. Ko, S. Soatto, and D. Estrin. Background subtraction on distributions. In *ECCV*, pages 276–289, 2008.

[9] S. Kuthirummal, A. Agarwala, D. B. Goldman, and S. K. Nayar. Priors for Large Photo Collections and What They Reveal about Cameras. In *ECCV*, pages 74–87, 2008.

[10] D. Lee. Effective gaussian mixture learning for video background subtraction. *IEEE Trans. PAMI*, 27(5):827–832, 2005.

[11] D. Lowe. Distinctive image features from scale-invariant keypoints. *IJCV*, 60(2):91–110, 2004.

[12] M. Mason and Z. Duric. Using histograms to detect and track objects in color video. In *30th AIPR*, pages 154–159, 2001.

[13] A. Mittal and N. Paragios. Motion-based background subtraction using adaptive kernel density estimation. In *CVPR*, volume 2, pages II–302, 2004.

[14] D. Scott. *Multivariate density estimation*. Wiley, 1992.

[15] R. Serfling. Quantile functions for multivariate analysis: approaches and applications. *Statistica Neerlandica*, 56(2):214–232, 2002.

[16] C. Stauffer and W. Grimson. Adaptive background mixture models for real-time tracking. In *CVPR*, volume 2, pages 246–252, 1999.

[17] S. Suri, C. Tóth, and Y. Zhou. Range counting over multidimensional data streams. *Discrete & Computational Geometry*, 36(4):633–655, 2006.

[18] K. Toyama, J. Krumm, B. Brumitt, and B. Meyers. Wallflower: Principles and practice of background maintenance. In *ICCV*, volume 1, pages 255–261, 1999.

[19] J. Zhong and S. Sclaroff. Segmenting foreground objects from a dynamic textured background via a robust kalman filter. In *ICCV*, pages 44–50, 2003.