Article

A Novel and Practical Scheme for Resolving the Quality of Samples in Background Modeling

Guian Zhang 1,2, Zhiyong Yuan 2,*, Qianqian Tong 2 and Qiong Wang 1,*

1 Guangdong Provincial Key Laboratory of Computer Vision and Virtual Reality Technology, Shenzhen Institutes of Advanced Technology, Chinese Academy of Sciences, Shenzhen 518055, China; zga@whu.edu.cn
2 School of Computer Science, Wuhan University, Wuhan 430072, China; qqtwhucs@gmail.com
* Correspondence: yuanzywhu@gmail.com (Z.Y.); wangqiong@siat.ac.cn (Q.W.)

Received: 31 January 2019; Accepted: 14 March 2019; Published: 18 March 2019

Abstract: In view of its important application value, background modeling is studied so widely that many techniques have emerged, which mainly concentrate on the selections of the basic model, the granularity of processing, the components in a framework, etc. However, the quality of samples (QoS) for training has long been ignored. There are two aspects regarding this issue, which are how many samples are suitable and which samples are reliable. To tackle the “how many” problem, in this paper, we propose a convergent method, coined Bi-Variance (BV), to decide an appropriate endpoint in the training sequence. In this way, samples in the range from the first frame to the endpoint can be used for model establishment, rather than using all the samples. With respect to the “which” problem, we construct a pixel histogram for each pixel and subtract one from each bin (called number of intensity values (NoIV-1)), which can efficiently get rid of outliers. Furthermore, our work is plug-and-play in nature, so that it could be applied to diverse sample-based background subtraction methods. In experiments, we integrate our scheme into several state-of-the-art methods, and the results show that the performance of these methods in three indicators, recall, precision, and F-measure, improved from 4.95% to 16.47%, from 5.39% to 26.54%, and from 12.46% to 20.46%, respectively.

Keywords: bi-variance; convergent frame; denoising; pixel histogram; background modeling

1. Introduction

Background modeling is the basis and also the first step for many vision-based tasks, like foreground detection, objects of interest segmentation, high-level operations [1–4] (such as target recognition and tracking, pose and gesture analysis, action understanding), etc. In the past decades, a gamut of approaches, ranging from simple methods of applying differencing of adjacent frames to the complex methods of mixing or combining multi-cues, have been proposed. Obviously, not only the accuracy of the background model in detection or other applications has been improved greatly, but the model itself has evolved by assimilating new techniques for pursuing higher efficiency and robustness.

Statistical theory has proven appropriate for modeling uncertainties of natural scenes [5], and in fact, it yielded the best results in CDnet2014 [6]. Generally, most of those methods based on statistics obey a suggestion from Stauffer and Grimson [7] that a recent sequence of observations is more important for parameters’ estimate, which somewhat coincides with the Markov property [8]. These observations are used for obtaining optimal parameters of the model in the training stage, whether online or offline, and then, these parameters are applied for initialization to guide the model to proceed elegantly. Subsequently, many researchers have determined this number $N$ heuristically or by choosing $N$ at point $N_i$, where the performance reaches the beginning of the plateau, from a range $(N_0, N_m)$.
For example, Elgammal et al. [9] simply took a sample of size 100 to train and represent the background in their KDE model. The number of samples adopted analytically by Barnich and Droogenbroeck’s ViBe model [10] and the SuBSENSE model of St-Charles et al. [11] was 20 and 35, respectively. We found that in CDnet 2014, SuBSENSE performed better than KDE and ViBe, while there were fewer samples needed in SuBSENSE than KDE and a few more than ViBe.

Therefore, a naturally-asked question is how many samples are needed after all and which samples are the best. All these questions point to an essential problem, which is the quality of samples (QoS) used for training or learning. To the best of our knowledge, QoS has not yet been focused on, and there is no principled way to settle it. In this paper, we tackle the two problems of “how many” samples and “which” samples of QoS.

To obtain an appropriate number in a principled way (that is the “how many” problem), we apply the tool variance, which is an important digital characteristic of random variables in statistics, on pixel values accumulatively (that is, each time from the first frame to the current frame for each pixel point). Thus, a series of variances is obtained, and the phenomenon of tending to converge at some frame emerges. However, for different pixel points, these variance sequences have different variance values where they converge, which can be seen from their distributions clearly, and this means that we cannot use a unified criterion to judge when they converge. We propose a new idea of applying the variance upon a window while sliding it on a previous variance sequence, which can tune the different values of corresponding converging points to a small positive value (such as $5^{-2}$), and so gets its name, bi-variance. The proof and more details will be given in Section 3.1.

The “which” problem mainly concerns the effect of noises. We know that noises are always present in videos and, thus, disturb the modeling process of normal intensity values. Methods in the literature either explicitly model the noise by the assumption of some distribution (for example, Gaussian with mean zero) or implicitly regard it as part of normal intensity values. However, noises are always there. To get rid of the noise, we construct the pixel histogram by counting the number of intensity values (NoIV-1) observed for each pixel point, and then at some time instant, we subtract one from those with a number greater than zero. This method obeys an underlying fact that the number of noises is less than that of normal intensity values in each pixel point over a period of time.

Summarily, the main contributions in this paper are: (1) a novel and effective convergent method, called Bi-Variance (BV), is proposed and proven in statistical and mathematical ways; and (2) a simple denoising method is introduced by subtracting one from the number of intensity values (NoIV-1); (3) our scheme can be easily integrated into other approaches thanks to its advantages of being plug-and-play; and (4) extensive tests are carried out to verify our proposals, and it turns out that the performance indicators of recall, precision, and F-measure are improved to different extents for several contemporary state-of-the-art models.

In the following, the content is organized as follows: the related work will be introduced firstly in Section 2; then, in Section 3, we will demonstrate our methods in detail and describe the algorithm in pseudocode; Section 4 is devoted to verifying the effectiveness of our scheme by applying it to several contemporary state-of-the-art models; finally, we will conclude our work and discuss how to make it perfect in our future work in Section 5.

2. Related Work

Background modeling of a real scene is a basic work for vision tasks. From about half a century ago to now, there have been numerous methods proposed for modeling the background [5,9–18] (a more detailed introduction about this technique and its applications was given in recent surveys [19–23]). However, most of them focus on how to choose the basic model (for example, Gaussian mixture model [7], kernel- or filter-based model [9,24], sample-based model [25–29], codebook model [30], etc.), the granularity of processing (for example, pixel-wise [13], blob-wise [31], region based [32,33], or spatiotemporal [5]), the components in a framework [10,11] (for example, one or more initializations, error propagation, and updating, apart from the essential subtraction process), and so on. In this
paper, we do not come up with a new theory or method for background modeling, but propose a plug-and-play method to promote the performance of those approaches based on statistics.

Generally, a recent sample of a pixel’s intensity values is needed, denoted as \( p_1, p_2, \cdots, p_N \), where \( N \) is the number of values. Then, the model will be constructed by using this sample, and the incident pixel \( p_t (t > N) \) will be judged as to which side it belongs, background or foreground. The question is how to decide the value of \( N \). Toyama et al. [34] set a sequence with at least 200 frames for training the background model. Elgammal et al. [9,35] decided to use 100 samples to represent their background. Sheikh and Shah [5] straightforwardly ignored the discussion about this parameter and set it to 200 in their implementation. Barnich and Droogenbroeck [10] studied the influence of \( N \) by testing it from 2–50 and then set it to 20 because the curve of \( N \)-performance became flat as \( N \) became greater than 20. After their seminal work, recently, researchers have started to pay some attention to the tradeoff between the performance and computational costs by choosing an appropriate value for \( N \).

Toyama et al. [34] set a sequence with at least 200 frames for training the background model. Elgammal et al. [9,35] decided to use 100 samples to represent their background. Sheikh and Shah [5] straightforwardly ignored the discussion about this parameter and set it to 200 in their implementation. Barnich and Droogenbroeck [10] studied the influence of \( N \) by testing it from 2–50 and then set it to 20 because the curve of \( N \)-performance became flat as \( N \) became greater than 20. After their seminal work, recently, researchers have started to pay some attention to the tradeoff between the performance and computational costs by choosing an appropriate value for \( N \).

Generally, a recent sample of a pixel’s intensity values is needed, denoted as \( p_1, p_2, \cdots, p_N \), where \( N \) is the number of values. Then, the model will be constructed by using this sample, and the incident pixel \( p_t (t > N) \) will be judged as to which side it belongs, background or foreground. The question is how to decide the value of \( N \). Toyama et al. [34] set a sequence with at least 200 frames for training the background model. Elgammal et al. [9,35] decided to use 100 samples to represent their background. Sheikh and Shah [5] straightforwardly ignored the discussion about this parameter and set it to 200 in their implementation. Barnich and Droogenbroeck [10] studied the influence of \( N \) by testing it from 2–50 and then set it to 20 because the curve of \( N \)-performance became flat as \( N \) became greater than 20. After their seminal work, recently, researchers have started to pay some attention to the tradeoff between the performance and computational costs by choosing an appropriate value for \( N \).

Toyama et al. [34] set a sequence with at least 200 frames for training the background model. Elgammal et al. [9,35] decided to use 100 samples to represent their background. Sheikh and Shah [5] straightforwardly ignored the discussion about this parameter and set it to 200 in their implementation. Barnich and Droogenbroeck [10] studied the influence of \( N \) by testing it from 2–50 and then set it to 20 because the curve of \( N \)-performance became flat as \( N \) became greater than 20. After their seminal work, recently, researchers have started to pay some attention to the tradeoff between the performance and computational costs by choosing an appropriate value for \( N \).

In order to be consistently robust against various challenges, like illumination change, camouflage, etc., models should resolve well the influence by noises. In the literature, we can conclude that the noise is always treated in one of two ways, either explicitly or implicitly. The explicit way [11,24,31,35,38–43] means that the noise is modeled by using some measures, for instance a Gaussian distribution assumption. Tavakkoli et al. [44] claimed that PFinder [31] models each pixel with a single 3D Gaussian and models the noise, then detects pixels whose probabilities are smaller than a threshold chosen heuristically. Soatto et al. [38] assumed that a Gaussian white noise is involved in their description of dynamic texture, which is a generative model for the dynamic background [45]. Jing Zhong and Stan Sclaroff [24], in their ARMA, assumed that the noise was distributed as \( N(0, Q) \) where \( Q \) is its variance. Furthermore, a band-pass filter was used previously to account for the background changes over time. Civelli et al. [41] proposed a so-called mixed state statistical framework. In this framework, they thought that background intensity values were perturbed by the Gaussian noise. However, unlike the aforementioned ways, they used a local average of intensity differences to reduce the influence of the noise. For computational consideration, Lópex-Rubio and López-Rubio [39] utilized a diagonal matrix to model the noise, which was introduced by the pixel’s quantization and compression. Nikos and Visvanathan [46] utilized the spatial information to wipe out the disturbance of the noise. Thereafter, Li et al. [47] superadded the temporal information (that is, the spatio-temporal features) to further improve the ability against noises. The implicit way [5,10,37,48] always views the noise as part of the sources in modeling the background and sometimes adds a post-processing step, for example, to fill the holes, remove the blinking pixels, etc. Sheikh and Shah [5] applied a threshold \( k \) in their classifier to balance the trade-off between the sensitivity to the change and the robustness to the noise. Barnich and Droogenbroeck [10] claimed that their model can automatically adapt to the noise because it was constructed from noisy pixel values. Seunglong Noh and Moongu Jeon [48] and St-Charles and Bilodeau [37] simply applied pre- and post-processing to smooth or reduce the noise, respectively. However, Hofmann et al. [13] worried about the side effect of that, whether pre or post, which will also smooth blobs and lead to less sharp boundaries. Apart from these two types, Elgammal et al. [9,35,40] denoted that image noises can be modeled by a zero mean normal distribution if the scene is completely static, so only local or small global displacements, except for random noises, were addressed in their method.

No matter which case it is, the noise is always there. In this paper, we try to eliminate the noise to provide high-quality samples to train the model, and thus, it can be robust from the beginning. Section 3.2 will demonstrate our idea of removing the noise away in detail.
In the next section, the methodology, the main part of our paper, will be elaborated.

3. Methodology

In this section, firstly, we will give the detailed explanations (including motivations, proofs, effects, and analyses) of our methods: bi-variance and NoIV-1, in Sections 3.1 and 3.2, respectively. Then, according to the descriptions presented in the former two parts, in Section 3.3, the corresponding algorithm is demonstrated in pseudocode.

3.1. Bi-Variance

As we know, variance is used to measure how far a set of (random) numbers is spread out from the average value. In the case of background modeling, we denote a recent set of intensity values of each pixel as \( p_1, p_2, \ldots, p_n, \ldots, p_N \), which are viewed as random numbers. Most of pixels' intensity values behave with some underlying laws (see Figure 1), which motivates us to try to use the variance to model those values.

![Figure 1](image)

**Figure 1.** Points \( p^1(204, 82) \) in (a), \( p^2(94, 195) \) in (b), \( p^3(217, 332) \) in (c), and \( p^4(15, 52) \) in (d) are selected randomly from the scene bus station in category shadow. Those four subfigures display the variation trend of the intensity values of the four points over a period of time. The X-axis' name, frame number, indicates the order of each frame in a video in the temporal direction.

It can be seen evidently that values of the pixel intensity fluctuate around some horizontal line and become stable asymptotically. To get the underlying law, we try to calculate variances of those cumulative sequences (like “\( p_1 \), “\( p_1, p_2 \), “\( p_1, p_2, p_3 \), etc.), which is defined as follows,

\[
V_{CS_n} = \text{var}([p_n]), n \in [1, N]
\]  

where \( \cdot \) represents a sequence and var indicates the math operator “variance”. Thus, we can intuitively view the tendency of intensity values changing over time (see Figure 2).
Figure 2. VCS tendencies of points mentioned in Figure 1 are showed in (a–d), separately. When the tendency reaches a stable status, the Y-axis value is also called the convergent value (cv).

As we expect, although they fluctuate drastically in the early stage, plateaus are always formed at some time instant. For instance, point p\(^1\) became stable approximately at Frame 65, and the VCS\(_{65}\) was two; point p\(^3\) almost became stable at 10 with the VCS\(_{10}\) being 0.9. So far, we can conclude, experimentally, that the VCS of each pixel intensity values was almost convergent and also call values like VCS\(_{65}\) convergent values (cv). However, the cvs (2 for point p\(^1\), 1.9 for point p\(^2\), 0.9 for point p\(^3\), and 3.6 for point p\(^4\)) were different, leading to a difficulty in judging when the convergent status (that is, the stable status) of each pixel intensity value was reached, which is related to resolving the “how many” sub-problem (the first problem of QoS). Therefore, we need a unified criterion to determine whether the sequence has achieved a convergent status.

Here, we propose a method, coined bi-variance, which views the VCS as a random variable, and thus, its value is the sequence (VCS\(_n\)). Then, we set a window (with a size L, as 30 in our work) above the (VCS\(_n\)) and also calculate the variance in each window by sliding along the (VCS\(_n\)). In this way, we can obtain a unified criterion, zero, in theory (nevertheless, generally a small non-negative real value, like 5e-2 in our experiment). In the following, we will give proofs from two aspects, statistics and mathematics.

In statistics, assume that \(X\) is a random variable; thus, its expectation and variance are \(E(X)\) and \(D(X)\), respectively, and \(D(X)\) can be calculated as follows,

\[
D(X) = E(X^2) - [E(X)]^2. \tag{2}
\]

Then, we define the bi-variance of \(X\) as \(B(X)\), and:

\[
B(X) = D(D(X)). \tag{3}
\]
By replacing (2) into (3) and expanding it,
\[
B(X) = E[D(X)^2] - E[D(X)]^2 \\
= E[(E(X^2) - E(X)^2)^2] - \{E(E(X^2) - E(X)^2)^2\} \\
= E(X^2)^2 - 2E(X^2)E(X)^2 + E(X)^4 - E(X)^2 \cdot 2E(X^2)E(X)^2 - E(X)^4 = 0,
\]
we can see that no matter what the variance of \( X \) is, its bi-variance will tend to be zero.

From another perspective, according to (4),
\[
B(X) = E[D(X)^2] - E[D(X)]^2 \\
= \frac{1}{n} \sum_{i=1}^{n} D(X_i)^2 - \overline{D(X)}^2 \\
= \frac{1}{n} \sum_{i=1}^{n} [D(X_i) - \overline{D(X)}]^2
\]
where \( \overline{D(X)} \) is the mean of sequence \( (D(X_n)) \). From the last row of (5), \( D(X_i) - \overline{D(X)} \) reflects the rate of \( D(X_i) \) deviating from the mean \( \overline{D(X)} \), and from Figure 2, we know that the tendency of \( D(X_i) \) goes toward being stable, which means the deviation becomes increasingly weaker, and so does \( D(X_i) - \overline{D(X)} \) (i.e., \( D(X_i) - \overline{D(X)} \) approaching zero).

In mathematics, a series \( S_n \) [51] can be defined by a sequence \( (a_n) \) in its summation form,
\[
S_n = a_1 + a_2 + a_3 + \cdots = \sum_{i=1}^{n} a_i
\]
and a straightforward way [52] for judging whether \( S_n \) is convergent is if the term \( a_i \) reduces to zero with the \( i \) growing, meaning that from some position, like \( i' \), when \( i > i' \), \( a_i \to 0 \), and thus, the sum of this sequence, \( S_n \), will tend to be a fixed value (that is, \( S_n = a_1 + a_2 + a_3 + \cdots + a_{i'} \)). Therefore, in (5), \( [D(X_i) - \overline{D(X)}]^2 \) can be viewed as a term of a sequence \( ([D(X_i) - \overline{D(X)}]^2) \), and thus, this sequence converges to zero (since \( \lim_{n \to \infty} \frac{S_n}{n} = 0 \) with \( S_n \) fixed).

By using the bi-variance on VCS in a window, we can get a new series of values. We call those values as “bi-variance”, which can converge in a unified way. Figure 3 illustrates the bi-variances of those VCS in Figure 2.

![Figure 3. Cont.](image-url)
Figure 3. The bi-variance method is applied to the VCS of the four points displayed in Figure 2. Points $p_1(204, 82)$ in (a), $p_2(94, 195)$ in (b), $p_3(217, 332)$ in (c), and $p_4(15, 52)$ in (d).

As mentioned before, we utilized a small positive value, $5^{-2}$, as the convergent criterion. Thus, those four points will converge approximately at Frames 65, 75, 10, and 55, respectively.

3.2. NoIV-1

We know that the way of constructing the histogram of an image (e.g. Figure 4b) is to count the number of each intensity value (from 0–255) by scanning the whole image (Figure 4a). Different from the classic image histogram, a pixel histogram is established by counting the number of observed intensity values in the temporal axis (Figure 4c,d).

Figure 4. (a,c) display one image and an image sequence of the training set of the scene bus station in category shadow; (b,d) exhibit their histograms: image histogram and pixel histogram. Not like the image histogram, the range of pixel histogram always concentrates on a small interval, for instance from 45–64 in pixel (15, 52).
Further, by careful examination of Figure 4d, we can notice that the number of some intensity values (like 46, 63, 64) was very small, which is very important information from which our method derives.

Although we have no idea about when and where the noise will appear, we know that the number of normal intensity values is more than that of noises (normally, noises scatter in different places with the number of them being few during a short period). We simply cut each column by one in the pixel histogram (see Figure 5), leading to the disappearance of noises.

**Figure 5.** The number of squares in each column represents the number of intensity values in this column; for instance, the number of intensity values $I_5$ is eight. Squares in Level 1, colored in gray, will be deleted according to number of intensity values (NoIV-1).

However, the effect of normal intensity values (for the case in Figure 5, $I_1$, $I_8$, and $I_n$ are viewed as noises, and others are normal) depends on their distribution. Therefore, analyses of it go as follows.

Assume that intensity values of one certain pixel $p$ during a time period are $I_1, I_2, \ldots, I_n$. The number of each intensity value is denoted by $N_{I_1}, N_{I_2}, \ldots, N_{I_n}$. Thus, the total number of intensity values, $N_T$, is:

$$N_T = N_{I_1} + N_{I_2} + \cdots + N_{I_n}. \quad (7)$$

Then, the contributions of each intensity value are:

$$P_{I_1} = \frac{N_{I_1}}{N_T}, \ P_{I_2} = \frac{N_{I_2}}{N_T}, \ldots, P_{I_n} = \frac{N_{I_n}}{N_T}. \quad (8)$$

After using NoIV-1, the number of intensity values is:

$$N'_T = (N_{I_1} - 1) + (N_{I_2} - 1) + \cdots + (N_{I_n} - 1) = N_T - N_{L_1} \quad (9)$$

where $N_{L_1}$ indicates the number of the first-level pixel $p_{L_1}$ in the pixel histogram (for example, in Figure 5, the number of squares colored in gray in the bottom). According to (9), we will rewrite (8) as:

$$P'_{I_1} = \frac{N_{I_1} - 1}{N'_T}, \ P'_{I_2} = \frac{N_{I_2} - 1}{N'_T}, \ldots, P'_{I_n} = \frac{N_{I_n} - 1}{N'_T}. \quad (10)$$

On the one hand, for those intensity values with the number being only one, their contributions will be reduced to zero, like $I_1$, $I_8$, and $I_n$ in Figure 5; on the other hand, for the others, we analyze the ratio by comparing their contributions presented in (8) and (10) as follows,

$$\text{ratio} = \frac{P'_{I_i}}{P_{I_i}} = \frac{N_{I_i} - 1}{N'_T} \cdot \frac{N_T - N_{L_1}}{N_{I_i}} = \frac{N_T - N_{L_1}}{N_T - N_{L_1}} = \frac{N_T - N_{L_1}}{N_{I_i}} \quad (11)$$

where $i \in p - p_{L_1}$. Let us have a look at the first item, $\frac{N_T - N_{L_1}}{N_T - N_{L_1}}$, in (11), in which the numerator is a constant and the denominator, in fact, can reflect the shape of the distribution. If $N_{L_1}$ is large, especially exceeding half of $N_T$, the shape always tends to be flat, otherwise, tall and narrow. For the other item, $\frac{N_{I_i} - 1}{N_{I_i}}$, if $N_{I_i} = 1$ (that is the noise), the effect of NoIV-1 is a fatal blow, or greater...
than one, \( \lim_{N_i \to \infty} \frac{N_i - 1}{N_i} \to 1 \), meaning that the harm of those intensity values with a large number becomes weaker. Therefore, the contributions of the remaining intensity values mainly lie on the shape of their whole distribution, which is very important in determining an incident pixel belonging to the foreground or background in background models. As for the example in Figure 5, by applying NoIV-1, the contributions of noises, \( I_1, I_8, \) and \( I_{10} \), become zero. Apart from those with only one remaining weakened intensity (that is \( P_{I_{4}}, P_{I_{5}^r}, \) and \( P_{I_{10}^r} \) equal to \( \frac{15}{19} \)), the contribution of \( P_{I_{4}^r} \), \( P_{I_{5}^r} \), and \( P_{I_{10}^r} \) is enhanced a little to be \( \frac{60}{57} \), while \( P_{I_{5}^r} \) and \( P_{I_{10}^r} \) increase to \( \frac{105}{76} \) and \( \frac{45}{38} \), respectively.

In the real scene, by taking point \( p^4 \) for example, we can see that some peaks in Figure 6a (denoted by red circle) have disappeared compared to Figure 6b. Note that the point \( p^4 \) owns a unimodal distribution, which features a bump with contributions in two ends fewer than that of in the middle. That is why the points that have disappeared are located at the top or bottom in its distribution.

The effect of NoIV-1 on the VCS of the point \( p^4 \) can be seen in Figure 7. In Figure 7a, we can see that the VCS curve of point \( p^4 \) had big changes from about 4.2, dropping down to 2.1 sharply, then rising to 4 with fluctuation and tended to be stable at about Frame 100. By using NoIV-1, the VCS curve (denoted by \( p^4 \) in Figure 7b) experienced fewer big changes and went into the convergent status at about Frame 50 quickly. The same effect was reflected by the bi-variance (see c and d, respectively).

However, an extreme case is that if the scene is very complex, like trees in a storm, the shape of its histogram will be very flat, and there would be many intensities in the bottom of the histogram. That means that all the intensities of the quantity being one in the bottom of the histogram will be viewed as noises and thus will be deleted according to the rule of NoIV-1. In this situation, one effective way is to maintain the reference samples long term.

### 3.3. Algorithm

According to the descriptions of our methods above, we will give the corresponding algorithm in pseudocode in Algorithm 1. Since the case of using bi-variance partnering with NoIV-1 performs better than that of using bi-variance alone (see Section 4), in our implementation, we placed NoIV-1 as the first step, then applied bi-variance on those purified results. Note that as we have not optimized our code, especially in NoIV-1, the whole runtime in the training stage of each host method was delayed from several milliseconds to tens of milliseconds (if the scene was challenging, leading to the sequence VCS converging slowly).
Algorithm 1 The implementation of bi-variance and NoIV-1.

**Input:** an intensity value sequence \((p_n)\) of a pixel \(P\), which is gathered from a scene video or image sequence during a period of time (for example, a set of frames, from the first image of any scene, with the amount specified in each scene’s configuration file in our experiment)

**Output:** the new sequence \((p'_n)\) without noises and the convergent frame \(c_f\)

1. Establish the pixelHistogram from \((p_n)\)
2. Remove the bottom level of the pixelHistogram
3. Reconstruct the new sequence \((p'_n)\)
4. \(vcs\_std = []\) // to save the VCS
5. \(seqLen = (p'_n).length\)
6. for \(i = 1 : seqLen\) do
7. \(subSequence = (p'_n)[1..i]\)
8. \(vcs\_std[i] = std(subSequence)\)
9. end for
10. \(window = L\) // the length of the sliding window
11. \(cnt = 0\) // to record consecutive times of the bi-variance below \(\epsilon\), which is introduced in the following
12. \(cnt\_thres = THRESHOLD\) // the threshold of \(cnt\)
13. \(cf = seqLen\) // initialized for the non-convergent case
14. \(wstd = 0\) // the variance of \(vcs\_std\) in the window, i.e. the bi-variance
15. for \(j = window : seqLen\) do
16. \(wstd = std(vcs\_std(j - window + 1 : j))\)
   // \(\epsilon\), a small positive real value, like 5e-2
17. if \(wstd < \epsilon\) then
18. \(cnt = cnt + 1\)
19. if \(cnt < cnt\_thres\) then
20. \(cf = j - cnt\_thres\)
21. break
22. else
23. \(cnt = 0\)
24. end if
25. end if
26. end for
27. return \((p'_n), cf\)

---

![Figure 7. Cont.](image-url)
Figure 7. (a,b) depict the VCS curves of $p^4$ before and after using NoIV-1, respectively. (c,d) are the bi-variances of $p^4$ before and after using NoIV-1, respectively. Red rectangles emphasize the obvious changes.

4. Experiment

To validate the effectiveness of our methods, we tested our method on a PC with a 3.3-GHz Intel Xeon E3-1230 V2 380 CPU and 24 GB memory, and the environment was Microsoft Visual Studio 2010. We performed extensive tests on a well-known dataset (CDnet 2014) [6], which is adopted widely by researchers in studying background modeling and other works related to computer vision. CDnet 2014 involves eleven categories and 4–6 scenes in each one (53 scenes in total). There were three performance indicators, recall, precision, and F-measure, which are often used in the literature, and so, we also used them in this work. Their definitions are:

\[
\text{Recall} = \frac{TP}{TP + FN} \\
\text{Precision} = \frac{TP}{TP + FP} \\
F - \text{Measure} = 2 \cdot \frac{\text{Recall} \cdot \text{Precision}}{\text{Recall} + \text{Precision}} = \frac{2TP}{2TP + FN + FP}
\]

where $TP$, $FP$, $TN$, and $FN$ indicate the number of “true positives”, “false positives”, “true negatives”, and “false negatives”, respectively.

Four famous methods (KDE [9], ViBe [10], PBAS [13], SuBSENSE [11]) in the recent two decades were chosen to take the role of a carrier (as claimed before, our methods can be viewed as a plug-in).

Then, we set three scenarios for evaluating the performance of our methods:

1. The basic one is that the original codes of these methods were executed without any modification. Note that because the starting points of the test of each method were different (such as KDE and SuBSENSE testing after their training frames, while ViBe from the first frame and PBAS with the first one blank), we conventionally obeyed the rule of setting the training number, which is in each scene’s configuration file of CDnet 2014, as the starting points of the test.
2. The second test utilized the bi-variance to get the “right” number of frames for training based on the basic one.
3. The third scenario was a combination of the basic one with our methods, bi-variance and NoIV-1.

4.1. Quantitative Results and Analysis

The quantitative results of four host methods are listed in Tables 1–4, respectively. The last row with the title ratio indicates the performance enhancement by comparing the best results of each method after using bi-variance and NoIV-1 against the basic one.
### Table 1. Results of KDE executed on three scenarios.

| Scenario                      | Times | Recall | Precision | F-Measure |
|-------------------------------|-------|--------|-----------|-----------|
| basic                         | -     | 0.677320 | 0.460483  | 0.489317  |
| Bi-Variance only              | -     | 0.702139 | 0.457460  | 0.475546  |
| Bi-Variance and NoIV-1        | 1     | 0.734267 | 0.500651  | 0.525397  |
|                               | 2     | 0.744162 | 0.514061  | 0.536388  |
|                               | 4     | 0.790847 | 0.582704  | 0.586779  |
|                               | 7     | 0.749623 | 0.521745  | 0.542624  |
| ratio                        | -     | 10.86% | 26.54%    | 19.92%    |

### Table 2. Results of ViBe executed on three scenarios.

| Scenario                      | Times | Recall | Precision | F-Measure |
|-------------------------------|-------|--------|-----------|-----------|
| basic                         | -     | 0.536554 | 0.518769  | 0.442704  |
| Bi-Variance only              | -     | 0.536887 | 0.602184  | 0.499898  |
| Bi-Variance and NoIV-1        | 1     | 0.540093 | 0.602429  | 0.501373  |
|                               | 2     | 0.550113 | 0.614631  | 0.510812  |
|                               | 4     | 0.581746 | 0.633618  | 0.533265  |
|                               | 7     | 0.563545 | 0.621112  | 0.518979  |
| ratio                        | -     | 8.42%  | 22.14%    | 20.46%    |

### Table 3. Results of PBAS executed on three scenarios.

| Scenario                      | Times | Recall | Precision | F-Measure |
|-------------------------------|-------|--------|-----------|-----------|
| basic                         | -     | 0.703802 | 0.538357  | 0.556289  |
| Bi-Variance only              | -     | 0.712157 | 0.577949  | 0.579673  |
| Bi-Variance and NoIV-1        | 1     | 0.718419 | 0.614902  | 0.607733  |
|                               | 2     | 0.724719 | 0.624375  | 0.614255  |
|                               | 4     | 0.738673 | 0.635568  | 0.625628  |
|                               | 7     | 0.732899 | 0.630746  | 0.620714  |
| ratio                        | -     | 4.95%  | 18.01%    | 12.46%    |

### Table 4. Results of SuBSENSE executed on three scenarios.

| Scenario                      | Times | Recall | Precision | F-Measure |
|-------------------------------|-------|--------|-----------|-----------|
| basic                         | -     | 0.706819 | 0.742927  | 0.681875  |
| Bi-Variance only              | -     | 0.709695 | 0.762506  | 0.693169  |
| Bi-Variance and NoIV-1        | 1     | 0.717996 | 0.774682  | 0.699669  |
|                               | 2     | 0.730749 | 0.777897  | 0.706701  |
|                               | 4     | 0.823209 | 0.783015  | 0.767367  |
|                               | 7     | 0.738729 | 0.780936  | 0.712979  |
| ratio                        | -     | 16.47% | 5.39%     | 12.54%    |

Meanwhile, to provide intuitive knowledge, we depict all of these data in Figure 8.
In Tables 1–4, column “times” was only used for the third scenario, in which the method NoIV-1 was executed one time, two times, four times, and seven times, respectively. Firstly, we can notice that the performance of all host methods in the three indicators was enhanced by using bi-variance only or that together with NoIV-1. Secondly, the effects of applying their combination, bi-variance and NoIV-1, were better than those of only utilizing bi-variance. Thirdly, we found that when the method NoIV-1 was executed a different number of times, the final results were different. However, when we executed it four times, each host method in the combined model reached its best performance, and if we executed it three more times, the performance went down.

These phenomena reveal two facts: (1) the noise is a key factor in affecting the precision of background models; (2) the number of noises is a small number. By viewing Figure 9 below, there is a tendency of rising first and then descending for each method’s performance indicator. Recall Figure 5 and the analyses in Section 3.2: executing method “NoIV-1” more times was increasingly harmful to the normal intensities and would destroy (rather than restore) the distribution of data. Therefore, the tendencies in Figure 9 are reasonable, and the tendency was basically predictable if we executed the method “NoIV-1” more than seven times.

In Table 1, for method KDE, the values of the indicators precision and F-measure of bi-variance only are smaller than that of basic (visually in Figure 8, part of bi-variance only’s line below that of basic),
because the training number of frames of bi-variance only was more than that of basic (100 mentioned in Section 2), leading to more noises being introduced.

In Tables 2 and 3, the ratios of performance enhancement in recall are less than those in precision and F-measure, which shows that our methods effectively reduced the number of FP more than that of FN by referring to their definitions in (12).

Interestingly, in Figure 8, we can also find that SuBSENSE had the best performance in all three scenarios against the others, which is a more recent method and gained a very high ranking in CDnet 2014 [6]. ViBe’s performance obtained a relatively big promotion when our methods were applied to its training stage, which actually reflected that the influence of noises cannot be ignored and that an explicit processing will be better (for example, SuBSENSE adopts an explicit way, while ViBe’s denoising mechanism is by constructing the model from noisy data straightforwardly).

4.2. Qualitative Results and Analysis

The qualitative results (see Figure 10 below) will display each method’s cons and pros visually, especially in detail.

In Figure 10, for the results of the method KDE in the third row compared with those of KDE combined with our methods in the forth row, the obvious change is that the noise was depressed greatly (especially in the last three scenes).

For methods ViBe and PBAS, the promotion was relatively higher than that in the other two host methods, which can also be noticed in their quantitative results in Figure 8. For example, the method ViBe cannot reduce the disturbance of white lines in the badminton court of the scene badminton and the white pattern on the T-shirt of the scene office. However, those problems were alleviated after using our methods. Method PBAS was susceptible to illumination change, the dynamic background, and the thermal environment. Those can be identified distinctly in scenes office, overpass, and park in the seventh and eighth row.

As for the method SuBSENSE, one of its drawback is overfitting, which can be viewed in the last two rows; for instance, the one missing leg in scenes badminton and office, the separated body in scenes sofa and park, and the connected legs in the scene overpass.

4.3. Convergent Frames and Analysis

Table 5 gives the convergent frames (that is, the desirable number of training frames) of all scenes in each category in CDnet 2014 [6].

By comparing those convergent frames of different scenes, there are several highlights presented in Table 5:

1. the convergent frame of every scene was totally different from the others;
2. there was a big gap between the minimal convergent frame, 32, and the maximal convergent frame, 380, of the whole scenes;
3. the challenging scenes, like badWeather, dynamicBackground, PTZ, turbulence, etc., evolved longer to obtain convergence.

Therefore, we assert that the arrangement of assigning all scenes with the same \( N \) is not appropriate, and the exact value of \( N \) should be determined by the data themselves. Additionally, the runtime of our method was from about 20 ms to less than 2 s, which specifically depended on the complexity, the resolution, and the convergent frame of each scene. Since we directly plugged our method into the source code of each host method, it exactly led to a delay to some extent. We are also considering this issue and intend to optimize our approach for speedup.
Figure 10. Scenes from column (a–f) are from categories camerJitter, intermittentObjectMotion, baseline, dynamicBackground, shadow, and thermal, respectively.
Table 5. Convergent frames of all scenes in each category in CDnet 2014.

| Category                  | Scene               | Convergent Frame |
|---------------------------|---------------------|------------------|
| badWeather                | blizzard            | 38               |
|                           | skating             | 125              |
|                           | snowFall            | 66               |
|                           | wetSnow             | 139              |
| baseline                  | highway             | 64               |
|                           | office              | 60               |
|                           | pedestrians         | 58               |
|                           | PETS2006            | 51               |
| cameraJitter              | badminton           | 116              |
|                           | boulevard            | 92               |
|                           | sidewalk             | 88               |
|                           | traffic              | 96               |
| dynamic Background        | boats               | 237              |
|                           | canoe               | 32               |
|                           | fall                | 87               |
|                           | fountain01          | 67               |
|                           | fountain02          | 124              |
|                           | overpass            | 141              |
| intermittent ObjectMotion | abandonedBox        | 154              |
|                           | parking             | 76               |
|                           | sofa                | 62               |
|                           | streetLight         | 38               |
|                           | tramstop            | 172              |
|                           | winterDriveway      | 132              |
| lowFramerate              | port_0_17fps        | 233              |
|                           | tramCrossroad_1fps  | 89               |
|                           | tunnelExit_0_35fps  | 238              |
|                           | turnpike_0_5fps     | 118              |
| nightVideos               | bridgeEntry         | 218              |
|                           | busyBoulevard       | 163              |
|                           | fluidHighway        | 43               |
|                           | streetCornerAtNight | 152              |
|                           | tramStation         | 116              |
|                           | winterStreet        | 214              |
| PTZ                       | continuousPan       | 60               |
|                           | intermittentPan     | 195              |
|                           | twoPositionPTZCam   | 79               |
|                           | zoomInZoomOut       | 59               |
| shadow                    | backdoor            | 48               |
|                           | bungalows           | 37               |
|                           | busStation          | 166              |
|                           | copyMachine         | 85               |
|                           | cubicle             | 128              |
|                           | peopleInShade       | 80               |
| thermal                   | corridor            | 44               |
|                           | diningRoom          | 47               |
|                           | lakeSide            | 42               |
|                           | library             | 40               |
|                           | park                | 104              |
| turbulence                | turbulence0         | 373              |
|                           | turbulence1         | 380              |
|                           | turbulence2         | 149              |
|                           | turbulence3         | 338              |
5. Conclusions

In this paper, we focus on the quality of training samples (QoS) for background modeling, and we have proposed a plug-and-play method, bi-variance, to cope with the “how many” problem in QoS and coupled with another denoising method, called NoIV-1, to sweep away those outliers, which refers to the “which” problem of QoS. Four seminal works, KDE [9], ViBe [10], PBAS [13], and SuBSENSE [11], have been introduced to evaluate our methods. In the experiment, we can see that the performance of each method (in three indicators, recall, precision, and F-measure) has been enhanced to different degrees.

By viewing the ratio-rows from Tables 1–4, the ratio of performance promotion was limited, generally speaking. However, how to greatly improve the performance is concerned with the quality of background model itself. Consequently, in our future work, we need to propose a high-quality background model based on the current study.

Author Contributions: Funding acquisition, Q.W.; methodology, G.Z.; software, Q.T.; supervision, Z.Y.; validation, Q.T.; writing, original draft, G.Z.; writing, review and editing, Z.Y. and Q.W.

Funding: This work was supported by the Shenzhen Science and Technology Program under Grants JSGG2017041121341 and JCYJ20170413162617606.

Acknowledgments: The authors gratefully acknowledge the financial support above. Useful suggestions given by Weixu Zhu of Universite Libre de Bruxelles and Mianlun Zheng of University of Southern California are also acknowledged. Special thanks to the reviewers and Editors of Sensors.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Uddin, M.A.; Joolee, J.B.; Alam, A.; Lee, Y.K. Human action recognition using adaptive local motion descriptor in spark. IEEE Access 2017, 5, 21157–21167. [CrossRef]
2. Lin, L.; Zhu, M. Efficient Tracking of Moving Target Based on an Improved Fast Differential Evolution Algorithm. IEEE Access 2018, 6, 6820–6828. [CrossRef]
3. Singh, J.P.; Jain, S.; Arora, S.; Singh, U.P. Vision-Based Gait Recognition: A Survey. IEEE Access 2018. [CrossRef]
4. Chou, K.P.; Prasad, M.; Wu, D.; Sharma, N.; Li, D.L.; Lin, Y.F.; Blumenstein, M.; Lin, W.C.; Lin, C.T. Robust Feature-Based Automated Multi-View Human Action Recognition System. IEEE Access 2018, 6, 15283–15296. [CrossRef]
5. Sheikh, Y.; Shah, M. Bayesian modeling of dynamic scenes for object detection. IEEE Trans. Pattern Anal. Mach. Intell. 2005, 27, 1778–1792. [CrossRef]
6. Wang, Y.; Jodoin, P.M.; Porikli, F.; Konrad, J.; Benezeth, Y.; Ishwar, P. CDnet 2014: An expanded change detection benchmark dataset. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, Columbus, OH, USA, 23–28 June 2014; pp. 387–394.
7. Stauffer, C.; Grimson, W.E.L. Adaptive background mixture models for real-time tracking. In Proceedings of the Computer Vision and Pattern Recognition, Fort Collins, CO, USA, 23–25 June 1999; Volume 2, pp. 246–252.
8. Li, S.Z. Markov random field models in computer vision. In European Conference on Computer Vision; Springer: Berlin/Heidelberg, Germany, 1994, pp. 361–370.
9. Elgammal, A.; Harwood, D.; Davis, L. Non-parametric model for background subtraction. In European Conference on Computer Vision; Springer: Berlin/Heidelberg, Germany, 2000, pp. 751–767.
10. Barnich, O.; Van Droogenbroeck, M. ViBe: A universal background subtraction algorithm for video sequences. IEEE Trans. Image Process. 2011, 20, 1709–1724. [CrossRef]
11. St-Charles, P.L.; Bilodeau, G.A.; Bergervin, R. Subsense: A universal change detection method with local adaptive sensitivity. IEEE Trans. Image Process. 2015, 24, 359–373. [CrossRef]
12. Yang, L.; Li, J.; Luo, Y.; Zhao, Y.; Cheng, H.; Li, J. Deep Background Modeling Using Fully Convolutional Network. IEEE Trans. Intell. Transp. Syst. 2018, 19, 254–262. [CrossRef]
13. Hofmann, M.; Tiefenbacher, P.; Rigoll, G. Background segmentation with feedback: The pixel-based adaptive segmenter. In Proceedings of the 2012 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), Providence, RI, USA, 16–21 June 2012; pp. 38–43.

14. Zeng, D.; Zhu, M. Background Subtraction Using Multiscale Fully Convolutional Network. *IEEE Access* 2018, 6, 16010–16021. [CrossRef]

15. Varga, D.; Havasi, L.; Szirányi, T. Pedestrian detection in surveillance videos based on CS-LBP feature. In Proceedings of the 2015 International Conference on Models and Technologies for Intelligent Transportation Systems (MT-ITS), Budapest, Hungary, 3–5 June 2015; pp. 413–417.

16. Chiranjeevi, P.; Sengupta, S. Robust detection of moving objects in video sequences through rough set theory framework. *Image Vis. Comput.* 2012, 30, 829–842. [CrossRef]

17. Wang, S.; Wang, Y.; Chen, Y.; Pan, P.; Sun, Z.; He, G. Robust PCA Using Matrix Factorization for Background/Foreground Separation. *IEEE Access* 2018, 6, 18945–18953. [CrossRef]

18. Fang, W.; Zhang, T.; Zhao, C.; Soomro, D.B.; Taj, R.; Hu, H. Background Subtraction Based on Random Superpixels Under Multiple Scales for Video Analytics. *IEEE Access* 2018, 6, 33376–33386. [CrossRef]

19. Kim, W.; Jung, C. Illumination-invariant background subtraction: Comparative review, models, and prospects. *IEEE Access* 2017, 5, 8369–8384. [CrossRef]

20. Cuevas, C.; Martinez, R.; Garcia, N. Detection of stationary foreground objects: A survey. *Comput. Vis. Image Underst.* 2016, 152, 41–57. [CrossRef]

21. Maddalena, L.; Petrosino, A. Background subtraction for moving object detection in rgbd data: A survey. *J. Imaging* 2018, 4, 71. [CrossRef]

22. Bouwmans, T. Traditional and recent approaches in background modeling for foreground detection: An overview. *Comput. Sci. Rev.* 2014, 11, 31–66. [CrossRef]

23. Bouwmans, T.; Javed, S.; Sultana, M.; Jung, S.K. Deep Neural Network Concepts for Background Subtraction: A Systematic Review and Comparative Evaluation. *arXiv* 2018, arXiv:1811.05255

24. Zhong, J.; Sclaroff, S. Segmenting foreground objects from a dynamic textured background via a robust kalman filter. In Proceedings of the Ninth IEEE International Conference on Computer Vision 2003, Nice, France, 13–16 October 2003; pp. 44–50.

25. Wang, H.; Suter, D. Background subtraction based on a robust consensus method. In Proceedings of the 18th International Conference on Pattern recognition (ICPR’06), Hong Kong, China, 20–24 August 2006; Volume 1, pp. 223–226.

26. Han, G.; Cai, X.; Wang, J. Object detection based on combination of visible and thermal videos using a joint sample consensus background model. *J. Softw.* 2013, 8, 987–995. [CrossRef]

27. St-Charles, P.L.; Bilodeau, G.A.; Bergevin, R. Universal background subtraction using word consensus models. *IEEE Trans. Image Process.* 2016, 25, 4768–4781. [CrossRef]

28. Luo, H.; Li, B.; Zhou, Z. Improved background subtraction based on word consensus models. In Proceedings of the 2017 International Symposium on Intelligent Signal Processing and Communication Systems (ISPACS), Xiamen, China, 6–9 November 2017; pp. 689–694.

29. Javed, S.; Oh, S.; Jung, S. IPBAS: Improved pixel based adaptive background segmenter for background subtraction. In Proceedings of the Conference on Human Computer Interaction, Lisbon, Portugal, 15–17 July 2014.

30. Kim, K.; Chalidabhongse, T.H.; Harwood, D.; Davis, L. Real-time foreground–background segmentation using codebook model. *Real-Time Imaging* 2005, 11, 172–185. [CrossRef]

31. Wern, C.; Azarbayejani, A.; Darrell, T.; Pentland, A.P. Pfinder: Real-time tracking of human body. *IEEE Trans. Pattern Anal. Mach. Intell.* 1997, 19, 780–785.

32. Girshick, R.; Donahue, J.; Darrell, T.; Malik, J. Region-based convolutional networks for accurate object detection and segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.* 2016, 38, 142–158. [CrossRef]

33. Varadarajan, S.; Miller, P.; Zhou, H. Region-based Mixture of Gaussians models for foreground detection. *Pattern Recogn.* 2015, 48, 3488–3503. [CrossRef]

34. Toyama, K.; Krumm, J.; Brumitt, B.; Meyers, B. Wallflower: Principles and practice of background maintenance. In Proceedings of the Seventh IEEE International Conference on Computer Vision, Kerkyra, Greece, 20–27 September 1999; Volume 1, pp. 235–261.

35. Elgammal, A.; Duraiswami, R.; Harwood, D.; Davis, L.S. Background and foreground modeling using nonparametric kernel density estimation for visual surveillance. *Proc. IEEE* 2002, 90, 1151–1163. [CrossRef]
36. Bloisi, D.; Iocchi, L. Independent multimodal background subtraction. *Comput. Image* 2012, 39–44. [CrossRef]
37. St-Charles, P.L.; Bilodeau, G.A. Improving background subtraction using local binary similarity patterns. In Proceedings of the 2014 IEEE Winter Conference on Applications of Computer Vision (WACV), Steamboat Springs, CO, USA, 24–26 March 2014; pp. 509–515.
38. Soatto, S.; Doretto, G.; Wu, Y.N. Dynamic textures. In Proceedings of the Eighth IEEE International Conference on Computer Vision, Vancouver, BC, Canada, 7–14 July 2001; Volume 2, pp. 439–446.
39. López-Rubio, F.J.; López-Rubio, E. Features for stochastic approximation based foreground detection. *Comput. Vis. Image Underst.* 2015, 133, 30–50. [CrossRef]
40. Elgammal, A. Figure-ground segmentation–pixel-based. In *Visual Analysis of Humans*; Springer: Berlin/Heidelberg, Germany, 2011; pp. 31–51.
41. Crivelli, T.; Bouthemy, P.; Cernuschi-Frias, B.; Yao, J.F. Simultaneous motion detection and background reconstruction with a conditional mixed-state Markov random field. *Int. J. Comput. Vis.* 2011, 94, 295–316. [CrossRef]
42. Haines, T.S.; Xiang, T. Background subtraction with dirichlet processes. In *European Conference on Computer Vision*; Springer: Berlin/Heidelberg, Germany, 2012; pp. 99–113.
43. Makantasis, K.; Nikitakis, A.; Doulamis, A.D.; Doulamis, N.D.; Papaefstathiou, I. Data-driven background subtraction algorithm for in-camera acceleration in thermal imagery. *IEEE Trans. Circuits Syst. Video Technol.* 2018, 28, 2090–2104. [CrossRef]
44. Tavakkoli, A.; Nicolescu, M.; Bebis, G.; Nicolescu, M. Non-parametric statistical background modeling for efficient foreground region detection. *Mach. Vis. Appl.* 2009, 20, 395–409. [CrossRef]
45. Mumtaz, A.; Zhang, W.; Chan, A.B. Joint motion segmentation and background estimation in dynamic scenes. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Columbus, OH, USA, 23–28 June 2014; pp. 368–375.
46. Paragios, N.; Ramesh, V. A MRF-Based Approach for Real-Time Subway Monitoring. In Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2001), Kauai, HI, USA, 8–14 December 2001; pp. 1034–1040. doi:10.1109/CVPR.2001.990644. [CrossRef]
47. Li, L.; Huang, W.; Gu, I.Y.H.; Tian, Q. Statistical modeling of complex backgrounds for foreground object detection. *IEEE Trans. Image Process.* 2004, 13, 1459–1472. [CrossRef]
48. Noh, S.; Jeon, M. A new framework for background subtraction using multiple cues. In *Asian Conference on Computer Vision*; Springer: Berlin/Heidelberg, Germany, 2012; pp. 493–506.
49. Fisher, R.A. XV. The correlation between relatives on the supposition of Mendelian inheritance. *Earth Environ. Sci. Trans. R. Soc. Edinb.* 1919, 52, 399–433. [CrossRef]
50. Gaughan, E. *Introduction to Analysis*; American Mathematical Soc.: Providence, RI, USA, 1998; Volume 1.
51. Thompson, S.; Gardner, M. *Calculus Made Easy*; St. Martin’s Press: New York, NY, USA, 1998; Volume 1.
52. Spivak, M. *Calculus. World Student Series*; Addison Wesley: Boston, MA, USA, 1967.