An Improved Real-time Method for Counting People in Crowded Scenes Based on a Statistical Approach

Shirine Riachi, Walid Karam and Hanna Greige
University of Balamand, Deir Al Balamand, Al Kurah, Lebanon

Keywords: Crowd Counting, Indirect Approach, Feature Regression, SURF Features, PETS Dataset.

Abstract: In this paper, we present a real-time method for counting people in crowded conditions using an indirect/statistical approach. Our method is based on an algorithm by Albiol *et al.* that won the PETS 2009 contest on people counting. We employ a scale-invariant interest point detector from the state of the art coined SURF (Speeded-Up Robust Features), and we exploit motion information to retain only interest points belonging to moving people. Direct proportionality is then assumed between the number of remaining SURF points and the number of people. Our technique was first tested on three video sequences from the PETS dataset. Results showed an improvement over Albiol's in all the three cases. It was then tested on our set of video sequences taken under various conditions. Despite the complexity of the scenes, results were very reasonable with a mean relative error ranging from 9.36% to 17.06% and a mean absolute error ranging from 1.13 to 3.33. Testing this method on a new dataset proved its speed and accuracy under many shooting scenarios, especially in crowded conditions where the averaging process reduces the variations in the number of detected SURF points per person.

1 INTRODUCTION

Real-time estimation of the number of people in a given area could be crucial for crowd management and safety purposes, especially in places witnessing mass gatherings of people (stadiums, theatres, holy sites, subway stations, etc.). For instance, such information might be very useful to prepare evacuation plans in cases of potential threats. Moreover, an accurate estimation could lead numerous economic advantages like managing human resources, improving service quality and analyzing customers' behavior.

Early methods for people counting involve the use of turnstiles, thermal sensors, tally counters and light beams which represent the disadvantage of being inaccurate when more than one person is passing through the monitored gate at the same time. Besides, their use is limited to entrance gates and doors and cannot be extended to wider regions. Hence, it was necessary to rely on image processing techniques to provide a real-time automatic count of passing people through a certain region by analyzing a series of images captured with a video camera.

Common methods for crowd counting make use of detection algorithms to spot faces, heads, human silhouettes, or other parts of the human body in order to estimate the count. These methods proved to be accurate in low-density crowds, but their performance was drastically reduced in more crowded scenes involving a high-degree of occlusions. Recent methods aim at extracting some easily detectable scene features such as foreground pixels, edges, interest points, etc. A certain relation is then established between these features and the number of people. Some of these techniques could perform in real-time and with good accuracy especially in crowded scenes.

To maintain the real-time feature while achieving better accuracy, we base our work on (Albiol et al., 2009) winner of the PETS contest on people counting. We first detect interest points using the SURF algorithm (Bay et al., 2008). SURF is used for its stability and scale- and rotation-invariance (Bauer et al., 2007). Subsequently, static interest points are eliminated using ARPS (Adaptive Rood Pattern Search) block-matching technique (Nie and Ma, 2002). Finally, a linear relation is assumed between the remaining SURF points and the number of people.

There are mainly two contributions of this paper. First, we improve on a feature-based real-time

Riachi S., Karam W. and Greige H..
An Improved Real-time Method for Counting People in Crowded Scenes Based on a Statistical Approach.
DOI: 10.5220/0005108002030212
In *Proceedings of the 11th International Conference on Informatics in Control, Automation and Robotics* (ICINCO-2014), pages 203-212
ISBN: 978-989-758-040-6

people counting method from the state of the art by using a more stable, repeatable and scale-invariant interest point detector (SURF). Second, we test this method on a new, more challenging dataset of videos taken under various conditions, and involving different crowd densities and perspective effects in order to point out its strengths and weaknesses.

2 RELATED WORK

Recent research took advantage of the progress made in computer vision and image processing to devise robust crowd counting methods. These methods could be divided into two main categories: direct approach methods and indirect approach methods. The first, also called detection-based, aim at detecting people individually and counting them. Huang et al. (2011) perform ellipse detection for the whole human silhouette after extracting foreground blobs. The best-fit ellipse parameters are then used to determine the number of people in each blob. Zhang and Chen (2007) use a single Gaussian model for moving objects detection before people are segmented. To deal with the occlusions problem, they perform group tracking in order to keep record of the number of people in each group.

To overcome the complex task of people segmentation, some techniques suggest detecting only visible parts of the body such as faces (Zhao et al., 2009), heads (Merad et al., 2010; Subburaman et al., 2012), or Ω-shaped head-shoulders region (Li et al., 2008; Zeng and Ma, 2010). While these methods could be advantageous in some cases, their use is limited to specific camera viewpoints and sparse crowds.

Indirect approach methods, also called feature-based, started receiving more attention lately for their ability to deal with occlusions and perspective effects. These techniques exploit various image features such as foreground pixels and interest points, and they map them to the number of people in the image through a certain learning process.

An early real-time method developed by Davies et al. (1995) employs background removal techniques to obtain foreground pixels and edge pixels. Then a relation is established between the number of these pixels and the total number of people by combining these two measurements through a linear Kalman filter. A method proposed by Ma et al. (2004) performs geometric correction (GC) to bring all the objects at different distances to the same scale before establishing a linear relation between the scaled number of foreground pixels and the number of people. Similarly, a method proposed by Li et al. (2011) consists of extracting the foreground using an adaptive model for background subtraction. Two steps are then performed to remove the shadow treated falsely as foreground, first by analyzing the texture, and then the HSV values. Perspective effects are also accounted for by computing a normalization map where pixels are weighed according to the depth of the objects. Some low-level features such as total area, perimeter, edge pixels, etc. are extracted from each crowd blob and fed into a trainable regressor to estimate the output.

Marana et al. (1997) suggested that the texture of a crowd image is strongly affected by the crowd density. They assumed that a low-density crowd image presents coarse texture patterns while a high-density one presents fine texture patterns. Hence, texture features were extracted using two different methods: statistical and spectral. The statistical approach relies on Grey Level Dependence Matrix (GLDM) which estimates the probability of a pair of grey levels occurring in the image. The spectral approach uses frequency information by analyzing the Fourier spectrum. Recent texture-based methods include the one proposed by Chan et al. (2008) where the number of features extracted was increased to 28, and the Gaussian Process regression was employed to estimate the count. Also, Wen et al. (2011) use a set of well-established 2-D Gabor filters to extract global texture features, and the Least Squares Support Vector Machine (LS-SVM) to regress the output.

The winner of the PETS2009 contest on people counting was an algorithm presented by Albiol et al. (2009). The authors detect corner points using the Harris corner detector. Motion vectors computed with respect to the previous frame are subsequently associated to the detected corners through a multi-resolution block-matching technique, and corners with null motion vectors are filtered out. A constant number of points per person is assumed, so a linear relation is established between the remaining corner points and the number of people. Finally, the output is smoothed out by averaging over time to reduce the oscillations.

Conceptually similar methods trying to account for perspective and density effects were lately introduced. Conte et al. (2010) use the SURF algorithm for interest point detection. Moving interest points are then divided into groups of people according to their distance from the camera. Also, the density of each group of points is computed. These measures with the number of points for each cluster are fed into a trainable regressor (ε-SVR) to

obtain the results. Fradi and Dugelay (2012) use the SIFT detector instead and perform perspective normalization at a pixel level. A density-based clustering algorithm is also employed to divide moving interest points into clusters according to their density, and Gaussian Process regression is trained to estimate the count. These methods achieved a high accuracy rate, but they represent the disadvantage of being computationally expensive, and they require a lot of training.

3 PROPOSED METHOD

To overcome the complexity of segmentation and detection algorithms, and to keep on the crucial real-time characteristic of this system, we propose an indirect approach method that is strongly inspired by (Albiol et al., 2009).

The flowchart of our system is represented in Figure 1, and its components are further explained throughout this section.

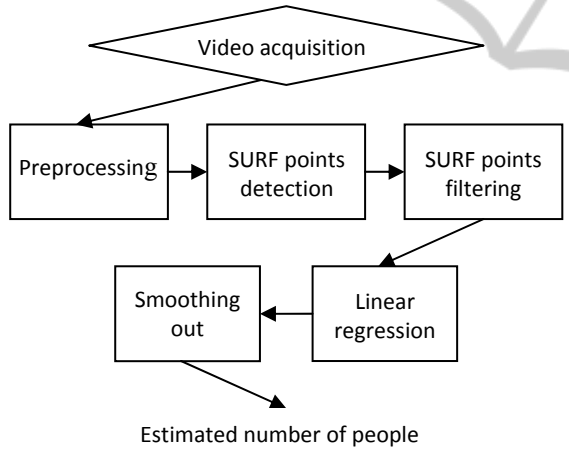

Figure 1: Architecture of the proposed technique.

3.1 SURF Points Detection

In order to understand how interest points could relate to the number of people in an image, we start by defining them. An interest point is a point that has a well-defined position in the image space, a clear and well-founded mathematical definition and can be robustly detected. Examples of such points are corners (intersection of two edges), line endings, T-shapes, blobs, a point on a curve where the curvature is locally maximal, etc. The most widely used interest point detectors are: Harris corner detector, Scale Invariant Feature Transform (SIFT) and Speeded-Up Robust Features (SURF).

As per (Conte et al., 2010), interest point detection is performed using the SURF algorithm (Bay et al., 2008) instead of the Harris corner detector. The choice of the first is based on (Bauer et al., 2007) where various implementations of Harris, SIFT and SURF methods were tested and evaluated for invariance against rotation, scale change, image noise, change in illumination and change in viewpoint. The authors concluded eventually that SURF is superior to the other two methods in terms of performance with respect to computational cost.

The nature of the interest points detected is highly dependant on the algorithm employed. For instance, SURF is based on the Hessian matrix; therefore it detects blob-like features. Moreover, it uses a basic Hessian matrix approximation and relies on integral images. This approximation by box type convolution filters and the use of its determinant for interest point localization and scale selection renders SURF very fast and scale-invariant. The mathematical background of the SURF algorithm, described hereafter, highlights its efficiency when used in people counting systems.

3.1.1 Integral Images

The use of box type convolution filters becomes computationally expensive when the filters' sizes increase. Therefore, integral images are associated to the original images in order to allow for fast computation of box type convolutions.

The entry of an integral image $I_\Sigma(\mathrm{z})$ at a location $\mathrm{z} = (x,y)$ represents the sum of all pixels in the input image I within a rectangular region formed by the origin and z.

$$I_\Sigma(\mathrm{z}) = \sum_{i=0}^{i\le x}\sum_{j=0}^{j\le y} I(i,j) \tag{1}$$

Using integral images, only three operations are needed to calculate the sum of intensities inside any rectangular area. This fact allows the use of big size filters without increasing the computational time.

3.1.2 Hessian Matrix

In their winning algorithm, Albiol et al. (2009) use the Harris corner detector. Assuming that corner points aren't a main characteristic of the human shape, we suggest detecting blob-like features instead. This is ensured by the Hessian matrix-based SURF algorithm.

The Hessian matrix $\mathrm{H}(\mathrm{z},\sigma)$ in a point $\mathrm{z} = (x,y)$ and at scale σ is given by:

$$\mathrm{H}(\mathrm{z},\sigma) = \begin{bmatrix} L_{xx}(\mathrm{z},\sigma) & L_{xy}(\mathrm{z},\sigma) \\ L_{xy}(\mathrm{z},\sigma) & L_{yy}(\mathrm{z},\sigma) \end{bmatrix} \tag{2}$$

Where $L_{xx}(z,\sigma)$ is the convolution of the Gaussian second order derivative $\frac{\partial^2}{\partial x^2} g(\sigma)$ with the image in point z, and similarly for $L_{xy}(z,\sigma)$ and $L_{yy}(z,\sigma)$.

The discrete nature of the images requires the Gaussian derivatives to be discretized and cropped. Bay et al. (2008) took a step further by also approximating Gaussians with box type convolution filters (see Figure 2).

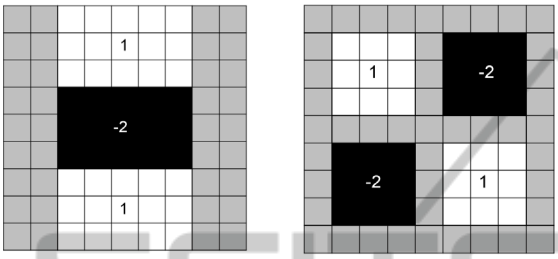

Figure 2: The approximated second order Gaussian derivatives for scale σ=1.2 in y- (Dyy) and xy- direction (Dxy), respectively.

With such filters used on integral images, it is enough to find the sum of the intensities in each rectangle/square (which requires only three integer operations), multiply by the corresponding weight, and add the results for the whole filter instead of convolving it with the original image

The blob response at a location z = (*x*,*y*) and scale σ is then given by the determinant of the approxi-mated Hessian:

$$\det(H\text{approx}) = DxxDyy - (0.9Dxy)^2 \quad (3)$$

Dxx, Dyy and *Dxy* are the approximated Gaussians. The value 0.9 is used to ensure energy is conserved between Gaussians and approximated Gaussians. Given that $|x|_F$ is the Frobenius norm, 0.9 is obtained as follows:

$$\frac{|L_{xy}(1.2)|_F |D_{yy}(9)|_F}{|L_{yy}(1.2)|_F |D_{xy}(9)|_F} \approx 0.9 \quad (4)$$

Further normalisation is performed by maintaining a constant Frobenius norm for all filters, and response maps are stored over space and scale allowing the detection of local maxima.

3.1.3 Scale Space

Creating a scale-invariant detector requires the interest points being found at different scales. This is usually achieved by building a scale space pyramid, where the base represents the lowest scale. The scale is then increased as we reach towards the top of the pyramid. Normally, to fill the pyramid upwards, the image is repeatedly smoothed by a Gaussian and sub-sampled. This sub-sampling might cause aliasing problems. Therefore, Bay et al. offer an alternative for this method where no sub-sampling is needed. Instead, they suggest filtering the original image by increasingly larger filters until reaching the top of the pyramid (see Figure 3). This isn't computationally expensive because, as mentioned previously, the use of box type filters on integral images is independent of the filter's size.

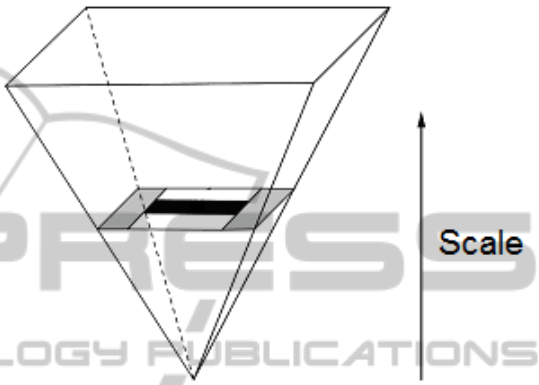

Figure 3: Bay scale pyramid formed by iteratively filtering with increasing size filters until the filter's size becomes larger than the original image.

The scale space is also divided into octaves; each octave covers a scale change of factor at least 2. Therefore, an octave includes a constant number of scale levels obtained by filtering the image repeatedly with bigger filters until the initial filter's size is more than doubled. Generally, three to four octaves are enough to cover all scales needed, and the octaves overlap to ensure full coverage of each scale.

The filters used for the first octave increase symmetrically by 6 pixels for each new scale level. They are of sizes 9, 15, 21 and 27. For each new octave, the increment in the size of the filter is doubled. Subsequently, the filter sizes used for the second octave are 15, 27, 39 and 51 with an increase of 12 pixels. The remaining octaves are obtained similarly.

3.1.4 Localisation

After filtering with the approximated Gaussian filters in the three directions (x-, y- and xy-) to construct the scale pyramid, the determinant of the approximated Hessian matrix is evaluated spatially and over scales. Subsequently, a fast non-maximum suppression method is employed to detect local maxima in each 3×3×3 neighbourhood.

The positions of these maxima are then interpo-lated in space and scale by fitting a 3D quadratic. For this purpose, the Taylor expansion of the Hessian determinant is computed:

$$H(\mathrm{x}) = H + (\frac{\partial H}{\partial \mathrm{x}})^{\mathrm{T}} \mathrm{x} + \frac{1}{2} \mathrm{x}^{T} \frac{\partial^2 H}{\partial \mathrm{x}^2} \mathrm{x} \quad (5)$$

Where $\mathrm{x} = (x,y,s)^{\mathrm{T}}$ is the coordinate in image space and scale, and $H(\mathrm{x})$ is the Hessian determinant at location x. Derivatives are approximated using finite differences formulas, and the interest point location $\hat{\mathrm{x}}$ is the extremum of this 3D quadratic, given by:

$$\hat{\mathrm{x}} = -(\frac{\partial^2 H}{\partial \mathrm{x}^2})^{-1} \frac{\partial H}{\partial \mathrm{x}} \quad (6)$$

Interpolation in scale space is particularly important as the difference between the first layers of each octave is large. Finding a sub-scale location of interest points can partially solve this issue.

Finally, it's worth noting that SURF is not only a detector, but also a descriptor that enables matching interest points in different frames. It uses techniques such as Haar Wavelet responses and sign of the Laplacian (i.e. trace of the Hessian matrix) to build 64×64 descriptors of interest points neighbourhoods, allowing them to be matched with corresponding points in other images. As this feature wasn't used in our case, it won't be explained in detail here.

3.2 Motion Estimation

Interest points are detected at blob-like locations all over the image. But since only those belonging to people are needed, a method should be employed to remove irrelevant ones. Based on the assumption that all pedestrians in the image are moving, motion information could be used efficiently to extract static interest points and eliminate them. This assumption is true in most cases because even if a person is standing, slight movements of his arms, legs and head happen all the time.

To estimate motion vectors efficiently, a well-known block-matching technique from the state of the art was employed. Based on (Barjatya, 2004), the Adaptive Rood Pattern Search (ARPS) algorithm (Nie and Ma, 2002) outperformed all the others in terms of accuracy and speed, therefore we choose to use it in our system.

Like all other block-matching techniques, ARPS divides the image into macro blocks of a specific size (usually 16×16 pixels). A cost function is then intelligently minimized to match each block with its corresponding one in a previous frame. This change in position of corresponding macro blocks enables the computation of their motion vectors. Interest points are then assigned the motion vector of the macro blocks they fall into.

The most widely used cost functions include the Mean Absolute Difference (MAD), the Sum of Absolute Differences (SAD) and the Mean Squared Error (MSE).

$$MAD = \frac{1}{N^2} \sum_{i=1}^{N} \sum_{j=1}^{N} | C(i,j) - R(i,j) | \quad (7)$$

$$SAD = \sum_{i=1}^{N} \sum_{j=1}^{N} | C(i,j) - R(i,j) | \quad (8)$$

$$MSE = \frac{1}{N^2} \sum_{i=1}^{N} \sum_{j=1}^{N} [C(i,j) - R(i,j)]^2 \quad (9)$$

Where N is the side of the macro block, C(i,j) and R(i,j) are the pixels at location (i,j) that are being compared in the current and reference macro blocks, respectively.

The neighbourhood of the macro block searched for the best match is specified by a search parameter p (p is usually equal to 7 pixels). While the basic Full Search (FS) algorithm checks all possible locations in the search area defined by p, ARPS moves directly to the most promising area where the probability of finding the matching block is the highest. This reduces drastically the computational time.

Based on the assumption that motion in a frame is coherent, i.e. a macro block moves in the same direction with its neighbouring blocks, ARPS exploits spatial correlation for motion vector prediction. More precisely, the motion vector of the current block is predicted to be equal to that of its immediate left.

Furthermore, camera movements occur mostly in the horizontal and vertical directions, therefore these directions need to be also checked for a possible match. Hereby, the search points for the first step include, in addition to the position pointed by the predicted vector, the adaptive rood pattern represented by circles in Figure 4. It consists of five points equally spaced at step size S = Max(|X|,|Y|), where X and Y are respectively the x- and y-coordinates of the predicted motion vector (in this case S = Max(|2|,|-4|) = 4). We note here that an overlap could occur between the predicted motion vector and a vertex of the rood pattern. This reduces by one the number of search points. Also, the rood pattern is reduced to one point if the predicted motion vector is zero. Finally, a step size of 2 is used for the leftmost macro blocks.

Moving to the second step, the point with the least weight becomes the center of the refined

search. The search pattern used here is the Unit-size Rood Pattern (URP) represented by triangles in Figure 4. This pattern is iteratively repeated after repositioning the minimum weighted point found at its center until the minimal matching error is found to be at the center of the URP.

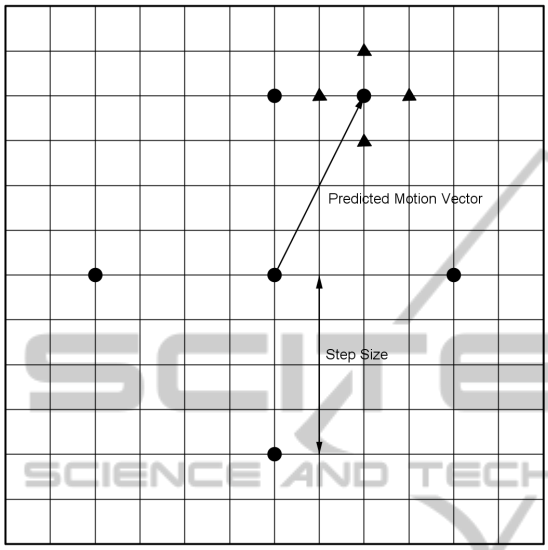

Figure 4: The points checked in the first step are represented by circles. The neighbourhood of the least weighted point is subsequently checked using the URP search pattern represented by triangles. This procedure is iteratively repeated until the minimum error is found at the center of the URP.

After interest points are associated with motion vectors, those whose motion vector's length is below a certain threshold are considered static and eliminated. The remaining points belonging to the foreground of moving people are retained to be used in the next step.

3.3 Linear Regression

Another assumption that each person contains an average number k of interest points is also made. Subsequently, the number of people in a frame could be obtained by dividing the total number of points N retained for that frame by k. To determine k, we have to train the system; that is, N is computed for a certain frame and the number of people P in that frame is annotated manually. k would then be equal to N/P. We note here that k is not constant for all shooting scenarios, it's rather highly dependant on the camera viewpoint and the scale perceived. Therefore, training has to be done whenever the viewpoint is changed.

3.4 Low-pass Filtering

In order to reduce the oscillations due to image noise, the initial estimate is smoothed out using a low-pass filter. In this case, we choose to average the output over a certain number of consecutive frames. This number depends on the frame rate at which the video was taken.

4 EXPERIMENTAL RESULTS

At a first stage, the proposed method is assessed using a public dataset. This enables the comparison of the results with those achieved by other methods, in this case with (Albiol et al., 2009). Nevertheless, this might not be sufficient as no single dataset can cover all possible scenarios.

At a second stage, four video sequences recorded indoor and outdoor under different viewing and weather conditions, also involving different crowd densities are tested.

4.1 Experiments on the PETS Dataset

PETS is a public dataset that has been widely used to assess crowd counting, density estimation and other algorithms. In our experimentations, three video sequences from the section S1 of the PETS2009 dataset are tested. The characteristics of these videos are shown in Table 1.

The number of people was estimated for each frame using the method described in Section 3, and the ground truth number (i.e. actual number) was annotated manually. The frame used for training is the one with the highest number of people, and the low-pass filtering is performed by averaging the output over 7 frames.

Table 1: Characteristics of the three PETS videos used.

			Number of people	
Video sequence	Length (frames)	Conditions	Min	Max
S1.L1.13-57 (View1)	221	Medium density crowd, overcast	5	34
S1.L1.13-57 (View 2)	221	Medium density crowd, overcast	8	46
S1.L1.13-59 (View 1)	241	Medium density crowd, overcast	3	26

Results are reported using the Mean Absolute

Error (MAE) and the Mean Relative Error (MRE) defined as follows:

$$MAE = \frac{1}{N}\sum_{i=1}^{N} |E(i) - T(i)| \quad (10)$$

$$MRE = \frac{1}{N}\sum_{i=1}^{N} \frac{|E(i) - T(i)|}{T(i)} \quad (11)$$

Where N is the number of frames in the video sequence, E(i) and T(i) are the estimated and the true number of persons in the *i*-th frame, respectively.

The results obtained are shown in Table 2 along with those of Albiol et al. as provided in (Conte et al., 2010). Also, graphs of the estimated and the ground truth number of people with respect to time are represented in Figure 5.

Table 2: Results on the PETS videos.

Video sequence	View	Albiol et al.		Our method	
		MAE	MRE	MAE	MRE
S1.L1.13-57	1	2.80	12.6%	2.28	11.5%
S1.L1.13-57	2	29.45	106.0%	12.06	36.3%
S1.L1.13-59	1	3.86	24.9%	1.81	12.7%

An overall improvement over Albiol's method is noticed for these three video sequences while simplicity and computational efficiency are maintained. Despite the improvement, results aren't very impressive for the View 2 of the S1.L1.13-57 sequence with an MAE of 12.06 and an MRE of 36.3%. This is due mainly to the wide-depth range characteristic of this video as people's trajectory is almost parallel to the optical axis of the camera. This fact increases significantly the perspective effects.

4.2 Experiments on Our Dataset

Assessing further the performance of the proposed method requires its testing on a set of more challenging videos. For this purpose, four video sequences were taken in three different locations in Lebanon. Their characteristics are represented in Table 3.

All of these sequences include pedestrians of different sizes (adults and kids). Some of them also include people that are standing or sitting (Za.B.0001), and occasionally some non human moving elements such as boats, water (Za.B.0001, U.O.B.0003 U.O.B.0004), or people with trolleys (B.C.C.0002). Serious occlusions and perspective effects are also observed in some cases (U.O.B.0003 and U.O.B.0004).

Frames are extracted from these videos and preprocessing is done to make them similar to the PETS videos in terms of frame rate, size and bit-depth.

Training of the system and smoothing of the output are performed similarly to the PETS experiments. The results are also reported in terms of MAE and MRE as shown in Table 4, and as comparative graphs in Figure 6. It is worth noting that the use of the relative error (MRE) is necessary. The same absolute error could be considered trivial if the scene is crowded, while it becomes significant as the level of crowdedness decreases.

Table 3: Characteristics of our set of videos.

Video sequence	Length (frames)	Conditions	Number of people	
			Min	Max
Za.B.0001	216	Medium density crowd, bright sunshine, shadows	21	32
B.C.C.0002	251	Low to medium density crowd, indoor	2	21
U.O.B.0003	251	Medium density crowd, bright sunshine	8	22
U.O.B.0004	221	Medium to high density crowd, bright sunshine	14	34

Table 4: Results on our dataset of videos.

Video sequence	**MAE**	**MRE**
Za.B.0001	2.53	9.36%
B.C.C.0002	1.13	13.04%
U.O.B.0003	2.81	17.06%
U.O.B.0004	3.33	13.48%

Despite the complexity of the scenes, the results achieved were very reasonable with an MAE ranging from 1.13 to 3.33 and an MRE ranging from 9.36% to 17.06%. Surprisingly, the most complex scene (Za.B.0001) produced the best MRE (9.36%) and the second best MAE (2.53). The least crowded scene with remarkable perspective effects produced the worst results (U.O.B.0003).

It was also noticed throughout this experiment that the performance of the proposed algorithm is best in crowded conditions. The use of a statistical approach and the large number of people allow these variations in the number of detected points per person to compensate each other. Moreover, they diminish remarkably the effect of the error occurred in motion estimation, causing some outliers in interest points filtering.

It was shown as well that the presence of a few still people or non-human moving objects in the scene could be tolerated into a certain extent, especially when the scene is crowded.

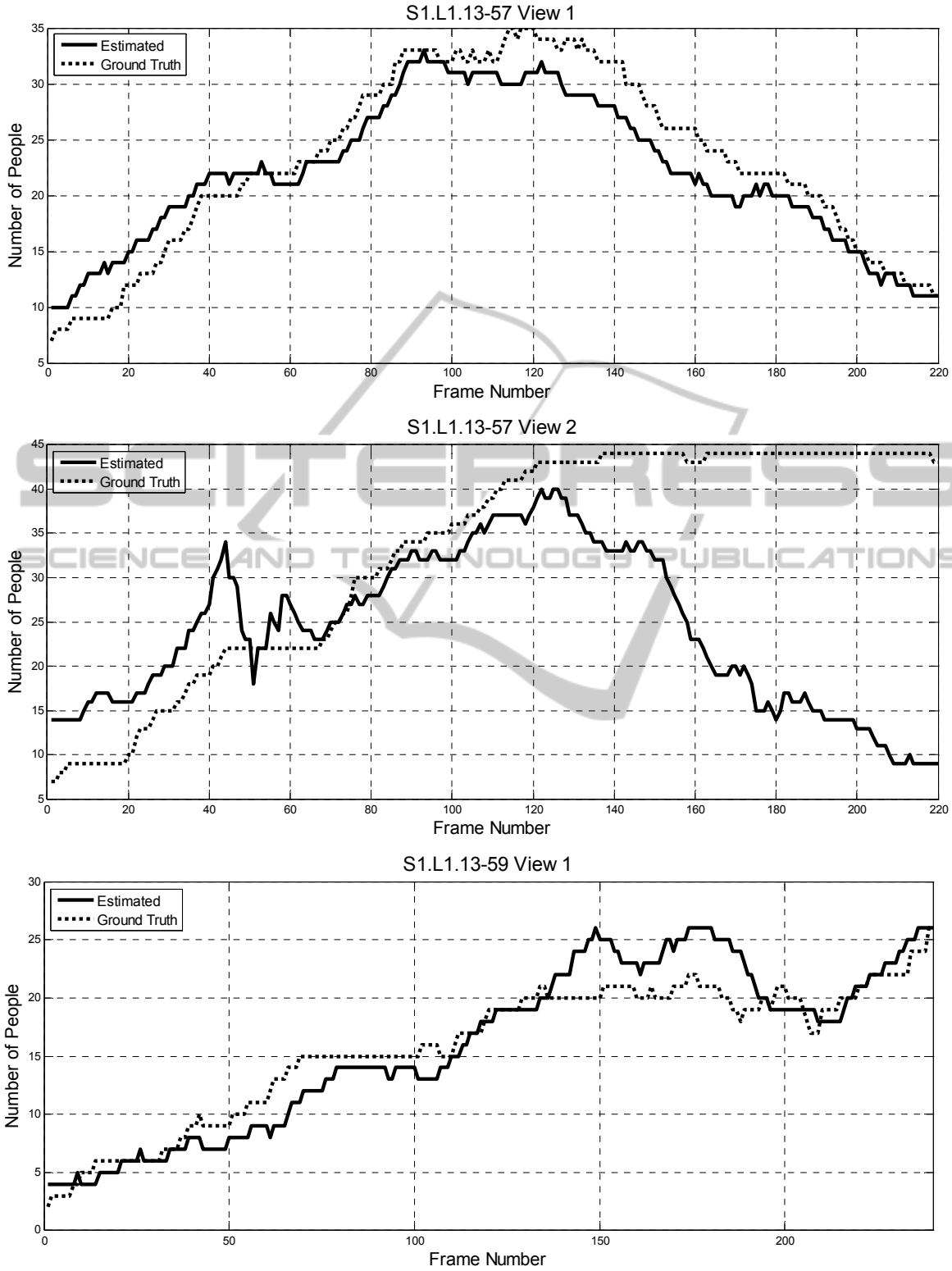

Figure 5: Estimated and ground truth number of people versus time for the PETS videos.

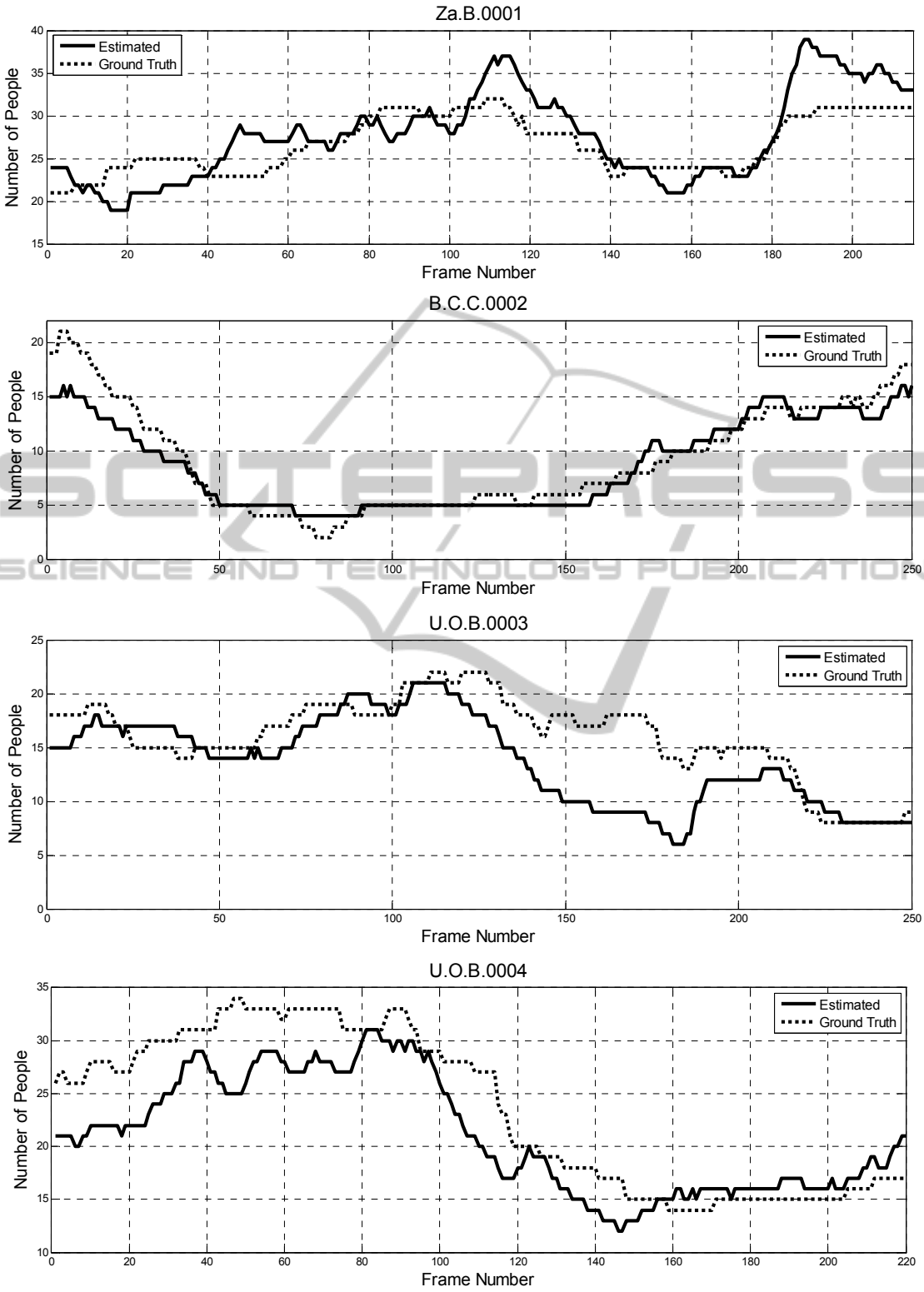

Figure 6: Estimated and ground truth number of people versus time for our set of videos.

5 CONCLUSIONS

In this paper, we have presented a real-time method for counting people in crowded scenes based on a statistical approach. We took advantage of the work achieved by Albiol et al. in this field, and we proved that with minor changes, significant improvement in accuracy could be accomplished. Furthermore, we maintained the robustness, simplicity, and computational efficiency of their algorithm.

The experiments undertaken on a new and more challenging dataset of video sequences confirmed the accuracy of the proposed technique in indoor and outdoor scenarios, and under different viewing and weather conditions. It also revealed its ability to handle partial occlusions and perspective effects up to a certain extent especially in crowded conditions.

REFERENCES

Albiol, A., Silla, M. J., Albiol, A., & Mossi, J. E. M. (2009). Video analysis using corner motion statistics. In *IEEE International Workshop on Performance Evaluation of Tracking and Surveillance* (pp. 31-38).

Barjatya, A. (2004). Block matching algorithms for motion estimation. *IEEE Transactions Evolution Computation*, *8*(3), 225-239.

Bauer, J., Sunderhauf, N., & Protzel, P. (2007). Comparing several implementations of two recently published feature detectors. In *Proceedings of the International Conference on Intelligent and Autonomous Systems* (Vol. 6, No. pt 1).

Bay, H., Ess, A., Tuytelaars, T., & Van Gool, L. (2008). Speeded-up robust features (SURF). *Computer vision and image understanding*, *110*(3), 346-359.

Chan, A. B., Liang, Z. S., & Vasconcelos, N. (2008). Privacy preserving crowd monitoring: Counting people without people models or tracking. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (pp. 1-7).

Conte, D., Foggia, P., Percannella, G., Tufano, F., & Vento, M. (2010). A method for counting people in crowded scenes. In *Seventh IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)* (pp. 225-232).

Davies, A. C., Yin, J. H., & Velastin, S. A. (1995). Crowd monitoring using image processing. *Electronics & Communication Engineering Journal*, *7*(1), 37-47.

Fradi, H., & Dugelay, J. (2012). People counting system in crowded scenes based on feature regression. In *Proceedings of the 20th European Signal Processing Conference (Eusipco)* (pp. 136-140).

Huang, C. L., Hsu, S. C., Tsao, I. C., Huang, B. S., Wang, H. W., & Lin, H. W. (2011). People counting using ellipse detection and forward/backward tracing. In *First Asian Conference on Pattern Recognition (ACPR)* (pp. 505-509).

Li, J., Huang, L., & Liu, C. (2011, August). Robust people counting in video surveillance: Dataset and system. In *8th IEEE International Conference on Advanced Video and Signal-Based Surveillance (AVSS)* (pp. 54-59).

Li, M., Zhang, Z., Huang, K., & Tan, T. (2008). Estimating the number of people in crowded scenes by mid based foreground segmentation and head-shoulder detection. In *19th International Conference on Pattern Recognition (ICPR)* (pp. 1-4).

Merad, D., Aziz, K. E., & Thome, N. (2010). Fast people counting using head detection from skeleton graph. In *Seventh IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)* (pp. 233-240).

Ma, R., Li, L., Huang, W., & Tian, Q. (2004). On pixel count based crowd density estimation for visual surveillance. In *IEEE Conference on Cybernetics and Intelligent Systems* (Vol. 1, pp. 170-173).

Marana, A. N., Velastin, S. A., Costa, L. D. F., & Lotufo, R. A. (1998). Automatic estimation of crowd density using texture. *Safety Science*, *28*(3), 165-175.

Nie, Y., & Ma, K. K. (2002). Adaptive rood pattern search for fast block-matching motion estimation. *IEEE Transactions on Image Processing, 11*(12), 1442-1449.

PETS dataset. (n.d.). Retrieved April 14, 2013 from http://www.cvg.rdg.ac.uk/PETS2013/a.html.

Rahmalan, H., Nixon, M. S., & Carter, J. N. (2006). On crowd density estimation for surveillance. In *The Institution of Engineering and Technology Conference on Crime and Security* (pp. 540-545).

Subburaman, V. B., Descamps, A., & Carincotte, C. (2012). Counting people in the crowd using a generic head detector. In *IEEE Ninth International Conference on Advanced Video and Signal-Based Surveillance (AVSS)* (pp. 470-475).

Wen, Q., Jia, C., Yu, Y., Chen, G., Yu, Z., & Zhou, C. (2011). People number estimation in the crowded scenes using texture analysis based on gabor filter. *Journal of Computational Information Systems*, *7*(11), 3754-3763.

Zeng, C., & Ma, H. (2010). Robust head-shoulder detection by pca-based multilevel hog-lbp detector for people counting. In *20th International Conference on Pattern Recognition (ICPR)* (pp. 2069-2072).

Zhang, E., & Chen, F. (2007). A fast and robust people counting method in video surveillance. In *International Conference on Computational Intelligence and Security* (pp. 339-343).

Zhao, X., Delleandrea, E., & Chen, L. (2009). A people counting system based on face detection and tracking in a video. In *Sixth IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)* (pp. 67-72).