Robust methods for background extraction in video

Martin Bruder[†], Gustavo A. Roitman[†] and Bruno Cernuschi-Frias[†‡]

†Facultad de Ingeniería, Universidad de Buenos Aires, Argentina
‡CONICET, Buenos Aires, Argentina

Abstract. In this paper[1] a framework is presented to automatically extract a sequence of images of the background of a scene from a shaky film. That is, the input video sequence may have local and global motion but the output video must contain exclusively the static background scene. Applying robust procedures to this end is one of the main goals of this work, since the aim is to get a procedure not only resistant to low scale noise but to occasional high scale noise. The median is used as an estimate of the background, the median absolute deviation (MAD) is used to establish a threshold to locate foreground and M-estimation for regression is used to stabilize the video sequence.

Keywords: background subtraction, robust estimation, video stabilization.

1 Introduction

Image sequences in the scope of this work come from a one-scene film that was not necessarily shot still - that is, they might contain camera jitter. At the same time, they have local motion produced by objects or people in the scene. This is why the framework is divided in three sections. First, the image sequence is stabilized according to a reference image to remove camera jitter produced on acquisition. An optical flow based technique [10, 2] with a multi-scale and robust approach has been used [13]. These methods achieve sub-pixel accuracy in contrast to feature-methods [18, 11, 5] which first extract features and then compute the image matching relations obtaining in some cases fast schemes [3]. Second, local motion is detected. This is achieved by estimating a model of the background with the temporal median, a method that does not update the background model on real time such as [17, 7, 1] but achieves low complexity computation [4]. Then, background and foreground are separated, the latter is removed to obtain a sequence with background information only. Next, the removed parts are filled in to accomplish a consistent sequence. Finally, the camera movement is returned to the edited sequence to show the original video with no foreground.
This paper is organized as follows: in section 2, robust estimation is introduced and some estimators that will be used are shown; in section 3, the stabilization

[1] This work was partially supported by Universidad de Buenos Aires and CONICET.

method based in optical flow [13] is given; in section 4, background estimation and video completion is explained and in section 5, global motion recovery is shown. In the last section the conclusions are drawn.

2 Robust estimation

In this section some robust procedures are explained and ways to analyze their behavior are shown. These methods will then be used in the next sections: to justify the minimization function chosen to estimate the parameters to align the images, to estimate the background of the scene and to select the threshold to differentiate between foreground and background.

Two basic tools to measure robustness are the "breakdown point" and the "influence function". Both are explained below.

- **Breakdown point**: In simple terms it is the smallest fraction of contamination that would cause an estimator to take arbitrarily large values [9]. Let,

$$X = \{x_1, x_2, \ldots, x_n\}$$

a set of n samples. Let T be an estimator and $T(X)$ its value at the samples. Consider X' as a set of corrupted samples obtained replacing ϵ original samples for any arbitrary values. Let

$$b(\epsilon, T, X) = \sup_{X'} ||T(X') - T(X)||$$

the maximum bias caused by a ϵ-order contamination. The supremum is taken over the set of all X' with ϵ replacements. If $b(\epsilon, T, X)$ is infinite, then ϵ outliers can have an arbitrary large effect on T. Then, the estimator "breaks down". The finite sample breakdown point for the estimator T on the sample X is defined as,

$$\epsilon_n^*(T, X) = min\left\{ \frac{\epsilon}{n} \,\middle|\, b(\epsilon, T, X) = \infty \right\}$$

There are estimators such as the sample mean with which only one observation can cause a breakdown, being $\epsilon_n^*(T, X) = 1/n$. On the other hand there are other estimators such as the constants which never break. However, useful estimators have a breakdown point of at best 0.5, for example the median.

- **Influence function**: introduced by Hampel in 1974 [8] as a function that describes the effect of an infinitesimal contamination at a point on an esti-

mator. It is defined as:

$$\begin{aligned} IF(x_0, T, F) &= \lim_{t \to 0^+} \frac{T\big((1-t)F + t\delta_{x_0}\big) - T(F)}{t} \\ &= \frac{\partial}{\partial t} T\big((1-t)F + t\delta_{x_0}\big)\Big|_{t \to 0^+} \\ &= \frac{\partial}{\partial t} T\big(F_t\big)\Big|_{t \to 0^+} \end{aligned}$$

where F is the distribution function, T the estimator, δ_{x_0} the mass point at x_0 (the distribution functions is $P(x = x_0) = 1$) and t the contamination fraction.

2.1 M-estimators

M-estimators are a generalization of maximum likelihood estimators and some of them show robustness properties.
An estimator $\hat{\theta}_n = \hat{\theta}(F_n)$ defined as a minimization problem such as

$$\hat{\theta}_n = \operatorname*{argmin}_{\theta} \sum_{i=1}^{n} \rho(x_i, \theta)$$

is an M-estimator with a score function ρ, an estimation parameter θ and the observations x_i. This is a generalization of the maximum likelihood estimator and it is simplified in this particular case when

$$\rho(x, \theta) = -\log f(x/\theta)$$

being f the probability density function of x. This can be seen in the likelihood function to be maximized in a maximum likelihood estimation:

$$L(\theta/x_1, \ldots, x_n) = \prod_{i=1}^{n} f(x_i/\theta)$$

which is equivalent to

$$l(\theta/x_1, \ldots, x_n) = \log \prod_{i=1}^{n} f(x_i/\theta) = \sum_{i=1}^{n} \log f(x_i/\theta)$$

The M-estimator has an equivalent form called its implicit equation

$$\sum_{i=1}^{n} \psi(x_i, \hat{\theta}_n) = 0$$

where $\psi(x, \theta) = (\partial/\partial\theta)\rho(x, \theta)$. It can be shown [9] that the shape of the influence function does not depend on the samples distribution function and it is,

$$IF(x_0, F) = \frac{\psi(x_0, \hat{\theta})}{-\int \psi'(x, \hat{\theta}) F(dx)}$$

Next, some useful examples of M-estimators are presented.

Median An M-estimator of location is given by

$$\hat{\theta}_n = \underset{\theta}{\operatorname{argmin}} \sum_{i=1}^{n} \rho(x_i - \theta) \text{ or } \sum_{i=1}^{n} \psi(x_i - \hat{\theta}_n) = 0$$

If the distribution of the samples has an exponential form

$$f(x) = \frac{1}{2} e^{-|x|}$$

the score function given by the maximum likelihood estimator is $\rho(x) = |x|$ except for a constant. This is equivalent to

$$\hat{\theta}_n = \underset{\theta}{\operatorname{argmin}} \sum_{i=1}^{n} |x_i - \theta|$$

The derivative of ρ exists for $x \neq 0$ and it's the sign function

$$sgn(x) = \begin{cases} -1 & x < 0 \\ 1 & x > 0 \end{cases}$$

It can be solved using the implicit equation

$$\sum_{i=1}^{n} \psi(x_i - \hat{\theta}_n) = \sum_{i=1}^{n} sgn(x_i - \hat{\theta}_n) = \#(x_i > \hat{\theta}_n) - \#(x_i < \hat{\theta}_n) = 0$$

with $\#(\cdot)$ the function that counts the event. This problem has a solution when $\#(x_i > \hat{\theta}_n) = \#(x_i < \hat{\theta}_n)$, which implies that $\hat{\theta}_n$ is the sample median.
This estimator is optimum in the maximum likelihood sense for the probability density function $f(x) = \frac{1}{2} e^{-|x|}$ but it is used in many non optimum situations because it has breakdown point of 0.5.

Median absolute deviation (MAD) It could be seen as a scale M-estimator. It is calculated as

$$\hat{\theta} = med\{|x_i - med\{x_i\}|\}$$

being $med\{\cdot\}$ the sample median. In [9] is shown that its breakdown point is 0.5. Normally the MAD is used to estimate the standard deviation. For the estimate to be consistent it has to be multiplied by a scale factor that depends on the distribution of the samples. One way to express MAD is as:

$$P(|X - \mu| \leq \hat{\theta}) = \frac{1}{2}$$

Which is equivalent to

$$P\left(\left|\frac{X - \mu}{\sigma}\right| \leq \frac{\hat{\theta}}{\sigma}\right) = P\left(|Z| \leq \frac{\hat{\theta}}{\sigma}\right) = \frac{1}{2}$$

Assuming that F is a normal distribution, it means that $F(\frac{\hat{\theta}}{\sigma}) - F(-\frac{\hat{\theta}}{\sigma}) = 1/2$ and $F(\frac{\hat{\theta}}{\sigma}) + F(-\frac{\hat{\theta}}{\sigma}) = 1$. With these equations one finds that

$$F\left(\frac{\hat{\theta}}{\sigma}\right) = \frac{3}{4} \Rightarrow \sigma = \frac{\hat{\theta}}{F^{-1}(3/4)} \sim \frac{\hat{\theta}}{0.6745}$$

While the estimator it is fairly simple to compute and it has a good breakdown point, its efficiency for the normal distribution is relatively low [16].

3 Sequence stabilization

The approach on the stabilization problem is focused exclusively on videos of one scene. It is assumed that motion was only produced by camera jitter so that the generated sequence could be as stable as possible to best recover the background.
The algorithm takes the last frame of the input sequence as the reference frame and estimates the parameters of a quadratic transformation for every frame of the sequence in relation to the last one. The quadratic model used assumes that a displacement in space $\mathbf{d}(x, y)$ from one image to another of a point (x, y) depends on twelve parameters. That is,

$$\mathbf{d}(x, y) = \begin{pmatrix} a_1 + a_2x + a_3y + a_4y^2 + a_5xy + a_6x^2 \\ b_1 + b_2x + b_3y + b_4y^2 + b_5xy + b_6x^2 \end{pmatrix}$$

Then, the parameters are used to align all the frames of the sequence with the last frame.
The algorithm used to estimate the parameters is developed in [13]. Odobez et al. define a minimization from

$$DFD = I(\mathbf{x}_i + B(\mathbf{x}_i)\mathbf{a}, t + 1) - I(\mathbf{x}_i, t) + \xi \tag{1}$$

an equation that is essentially based on the "constant brightness" hypothesis. And where $I(\mathbf{x}_i, t)$ is the intensity value of the image at position $\mathbf{x}$ in time t, ξ is a constant brightness factor that in this application is not used, $B(\mathbf{x})$ is a matrix that depends on the quadratic model used and $\mathbf{a}$ is a vector that contains the model parameters. These parameters are selected to minimize the residual product of linearizing this function in the vicinity of a previous estimation (the algorithm is based on an incremental scheme). The Tukey's "biweight" function is used (a robust score function) as the loss function to minimize the residuals. Thus, the influence of the outliers is reduced, penalizing in the same way the observations with an error above a certain threshold.

In the obtained sequences, black borders are found in areas where no image information was available as frame correction was performed. The results are presented in figure 1.

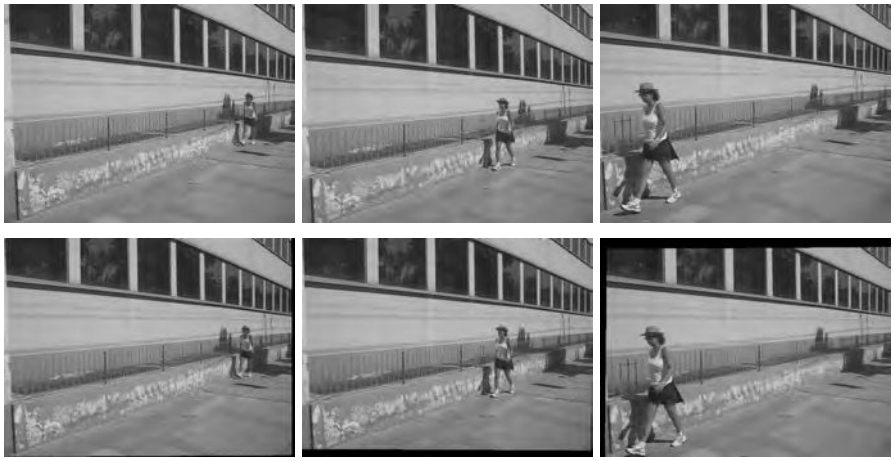

Fig. 1. Frames of the same sequence. Above, part of the original sequence and below, part of the aligned sequence using the quadratic model.

4 Background subtraction

In this section a technique for background estimation is developed and used for foreground detection. Next, two methods to fill removed objects with background are evaluated.

4.1 Background estimation

The input of the algorithm is a sequence of images aligned to a reference. Considering there is waste of information due to translation of the original images to the new reference, a white frame (at least the size of the maximum translation) is added to every image in the sequence to avoid this waste. These images are transformed with the parameters estimated for the original images to reduce the residual of the stabilization algorithm.
The temporal median is used for background estimation for each (x, y). This a low complexity and robust estimator.
Given this application, there are some points in the sequence that have a large quantity of outliers, in fact, a lot more than 50%, the breakdown point of the median. The intensity values near 0 show up due to the black border introduced at transformation by the translation of images. The points near 255 appear due to the white frame added. These points are outliers and should be removed prior to median calculation. It was observed that the points added in the frame are not exactly equal to the largest value of intensity as would be assumed - this is because the image was transformed by a warping algorithm that changed it increasing dispersion on estimated values.
Two thresholds are established to remove these outliers before the median is calculated. This could be calculated analyzing the dynamic range of the original

images and knowing the proportion of outliers to be removed, but in order to minimize the computational burden, these are manually fixed at relatively low costs. In the shown example this thresholds are $k_1 = 250$ y $k_2 = 2$. The reason for the selective threshold chosen for low intensity values is that black borders are introduced after the warping algorithm and not before.
The procedure is summed up in algorithm 1 and the result can be seen at figure 2. Note that the obtained image is greater than the originals.

Algorithm 1 Background estimation

1: Transform parameters are estimated for every image in relation to the reference.
2: A white frame is added considering the maximum translation in a transformed image of the sequence.
3: These new images are transformed using the parameters previously estimated.
4: Thresholds k_1 and k_2 are defined and values outside that range are extracted.
5: For every position (x, y) of $I(x, y, t)$ that has an accepted value, the median is calculated.

Fig. 2. Background estimation in 768 x 576 pixels.

It is worth pointing out that the estimation was not perfect especially near the borders where a "rainy" effect is seen. These is a consequence of boundary effects produced by the warping algorithm. These points are mainly outliers removed in step 4 of the algorithm but there are also grayish intensity values that belong

to the border of the transformed image and come from an interpolation between the real border of the image and the added white frame. If the contour of figure 2 is closely observed it can be seen that the last row of pixels of the border of the real image is "whitened". These are outliers that were not removed.

4.2 Video completion

Image segmentation and object removal are fields of special interest in image processing. Object removal leaves "holes" in a sequence where the information is unknown. In images, the process of estimating these unknowns is called "inpainting" and it is commonly solved using geometrical assumptions supported on boundary information. In video the term used is "completion". In this case one commonly counts with information from other images of the sequence to fill the missing parts, but when it is not available, inpainting techniques are used. In the following sections, a way to solve the video completion given the results of previous stages is detailed. The images at hand have a static background because after the alignment of the sequence all motion is considered foreground. This method does not cover the case in which objects are occluded during the whole sequence and nothing is assumed on the periodicity of motion [19].

Threshold selection To identify where the foreground and background is in the sequence, with a background estimation at hand, a threshold fixed at $k\sigma$ times the value of background is used. This means that for every point in the background image along with its corresponding point in the sequence, an evaluation of the intensity value is made. If the difference is high, the detector decides foreground, if it is below this certain threshold, background is assumed. In other terms, this equation is used:

$$|I_B(x,y) - I_t(x,y)| > K \tag{2}$$

where I_B is the estimated background, I_t is the t image of the sequence and K is the fraction of the estimator of standard deviation of intensity at position (x,y). For it to be robust and consistent, MAD was used (median absolute deviation) and it was assumed that the samples were drawn from a normal distribution. Then, [16]:

$$\hat{\sigma} = 1.4826 \cdot MAD \tag{3}$$

Two alternatives were considered for the threshold, $K = 2.5\hat{\sigma}$ and $K = 3\hat{\sigma}$. To evaluate and analyze their effectiveness, "recall" and precision were measured. As [15] points out, recall of a classification system is the fraction of relevant material given a classification, in this case it is measured as

$$\text{recall} = \frac{\text{p.f.w.c.}}{\text{r.p.f.f.}} \tag{4}$$

and precision is the fraction of relevant information classified, in this application it is

$$\text{precision} = \frac{\text{p.f.w.c.}}{\text{p.i.a.f.}} \tag{5}$$

with p.f.w.c. the "pixel of foreground well classified", p.i.a.f. the "pixels identified as foreground" and r.p.f.f. the "real pixels from foreground". Both recall and precision belong to the interval $[0, 1]$.
These measures were calculated according to their equations considering an area of interest that excludes black borders that show up in the images of the sequence. The pixel count took three images of the sequence. A pair of P-R values for each threshold were:

$$\text{recall} = 0.92 \quad \text{precision} = 0.55$$

for $K = 2.5\hat{\sigma}$, and:

$$\text{recall} = 0.90 \quad \text{precision} = 0.73$$

for $K = 3\hat{\sigma}$.
As it was expected, when the threshold is low, from the total foreground pixels, the actually identified ones are more than when the threshold is high. In this case the precision is lower because there are more background pixels that are actually identified as foreground. Although in this example the particular increase at recall is not as big as decrease at precision when $K = 2,5\hat{\sigma}$, in this particular application, this will be the threshold used given that the classification of less foreground pixels that actually exist would lead to a significant visual distortion (artifacts) to the final result.

Video completion Images were filled in two different ways. 1) Every pixel identified as foreground was replaced for the background estimation or; 2) it was replaced by the nearest neighbor identified as background.
In figure 3 results are shown. The results were visually similar. As a quantitative measure of evaluation the original sequence was compared with the filled sequence and the sum of square difference (SSD) was calculated for both methods. Results for the first technique where the background estimation was used was

$$SSD_1 = 12.8906$$

and for the nearest neighbor

$$SSD_2 = 12.7418$$

On the other hand, if small variations from background are taken as foreground in the classification stage it is possible that the nearest neighbor technique produces better results since it would adapt better to changes.

5 Global motion recovery

Once the sequence with filled background is obtained, global motion is restored. If the transformation used to align the images to the reference is bijective, one should be able to calculate the parameters of the inverse transformation and

Fig. 3. At the upper row, foreground is in black. At the lower rows pixels were replaced with background estimation first and with the nearest neighbor second.

apply this transformation to the modified images. In this application the transformation used is quadratic and has 12 parameters and it is not bijective so the parameters were estimated again with the stabilization technique previously used. Basically, parameters to align the reference with each image of the sequence are estimated (although the transformation of every image in the sequence of the stabilized video with the original images can also be used) and then, the transformation is applied to the modified images.
One could think that the original images are in a "moving sequence space" and the aligned images are in a "stabilized space". The images with the recovered background are in the stabilized space. The idea is to estimate the transformation parameters of the reference image with each image of the original sequence in the moving sequence space (It) and these are used to warp the modified images with the recovered background in the moving sequence space to return the global motion to them. A diagram of the procedure is shown in figure 4.

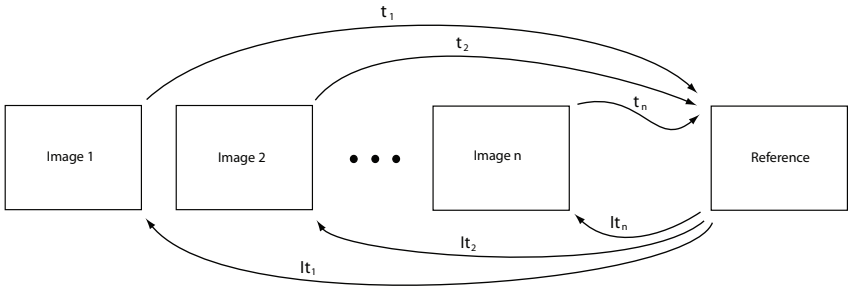

Fig. 4. For each image a transformation t is estimated to align it with the reference image and another one It is estimated to align it back to the original.

The main problem when global motion is returned to the images is the loss of information due to translation seen on black borders. This is solved during the first stages of the algorithm (stabilization, detection and completion) by using the images with a white frame added. Then, when motion is returned the images are resized cropping them to their original size.
In figure 5 images with the recovered background are seen on the moving sequence space at their original size. In the row below, images that belong to the original sequence are shown. The obvious difference between each corresponding image is the absence of foreground given that the sequences have their global motion restored. A second and more subtle difference is consequence of the transformation. The contrast of the processed images is lower. This is possibly a result of the warping algorithms that affect images as lowpass spatial filters. This is, however, only appreciable at close inspection.

Fig. 5. Images with global motion. At the upper row foreground is absent. At the lower row, the original sequence.

6 Conclusions

Throughout this work a framework to extract an image sequence of the background of a shaky film of a single scene is presented. The proposal is to stabilize the sequence initially and later extract the background information and replace the foreground with estimated values.
Automation was prioritized and robust methods were a main concern when procedures were chosen. The algorithm must be resistant to outliers commonly found in image processing problems.
Some characteristics of this algorithm are that the video must not necessarily be static. The background should be so. Nothing needs to be assumed about

the periodicity of the motion in the scene as some other algorithms do [19]. The classification method of foreground/background demands that background must be shown in more than 50% of the images for every pixel in order to prevent estimation errors. Other algorithms [14, 6, 12] use inpainting techniques to estimate occluded objects from background that could be well added to this procedure.
Calculation times are in the order of tens of seconds with a desktop computer. Image stabilization is what requires more capacity. The selected methods are effective on their results and that is why this optical flow method along with a quadratic model was chosen to align the sequence [13].
Future work could be to eliminate the restriction that the film must have only one scene. If the video has accepted global motion the procedure should include a reliable technique of "digital image stabilization (DIS)" and could also use a windowing scheme, applying the algorithm in small fractions of the whole sequence. They should be sufficiently small for the algorithm to be reliable and sufficiently large for it not be affected by the restriction of the fraction of images with uncovered background. Possibly, if different shots with changes in global motion and local structure are acquired, the algorithm could require to be semi-automatic and have an increase it the quantity of parameters to select.

References

[1] O. Barnich and M. Van Droogenbroeck. Vibe: A universal background subtraction algorithm for video sequences. *Image Processing, IEEE Transactions on*, pages 1–1, 2011.

[2] S. Battiato, G. Puglisi, and AR Bruna. A robust video stabilization system by adaptive motion vectors filtering. In *Multimedia and Expo, 2008 IEEE International Conference on*, pages 373–376. IEEE, 2008.

[3] H. Bay, A. Ess, T. Tuytelaars, and L. Van Gool. Speeded-up robust features (surf). *Computer Vision and Image Understanding*, 110(3):346–359, 2008.

[4] M. Blum, R.W. Floyd, V. Pratt, R.L. Rivest, and R.E. Tarjan. Time bounds for selection*. *Journal of Computer and System Sciences*, 7(4):448–461, 1973.

[5] A. Censi, A. Fusiello, and V. Roberto. Image stabilization by features tracking. In *Image Analysis and Processing, 1999. Proceedings. International Conference on*, pages 665–667. IEEE, 1999.

[6] A. Criminisi, P. Pérez, and K. Toyama. Region filling and object removal by exemplar-based image inpainting. *Image Processing, IEEE Transactions on*, 13(9):1200–1212, 2004.

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

[8] F.R. Hampel. The influence curve and its role in robust estimation. *Journal of the American Statistical Association*, pages 383–393, 1974.

[9] P.J. Huber and Ronchetti. *Robust statistics*, volume 1. Wiley, 1981.

[10] M. Irani and P. Anandan. About direct methods. *Vision Algorithms: Theory and Practice*, pages 267–277, 2000.

[11] D.G. Lowe. Distinctive image features from scale-invariant keypoints. *International journal of computer vision*, 60(2):91–110, 2004.

[12] Y. Matsushita, E. Ofek, W. Ge, X. Tang, and H.Y. Shum. Full-frame video stabilization with motion inpainting. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 1150–1163, 2006.

[13] J.M. Odobez and P. Bouthemy. Robust multiresolution estimation of parametric motion models. *Journal of visual communication and image representation*, 6(4):348–365, 1995.

[14] K.A. Patwardhan, G. Sapiro, and M. Bertalmio. Video inpainting of occluding and occluded objects. In *Image Processing, 2005. ICIP 2005. IEEE International Conference on*, volume 2, pages II–69. IEEE, 2005.

[15] C.J.V. Rijsbergen. *Information retrieval*. Butterworths, 1979.

[16] P.J. Rousseeuw and C. Croux. Alternatives to the median absolute deviation. *Journal of the American Statistical Association*, pages 1273–1283, 1993.

[17] C. Stauffer and W.E.L. Grimson. Adaptive background mixture models for real-time tracking. In *Computer Vision and Pattern Recognition, 1999. IEEE Computer Society Conference on.*, volume 2. IEEE, 1999.

[18] P. Torr and A. Zisserman. Feature based methods for structure and motion estimation. *Vision Algorithms: Theory and Practice*, pages 278–294, 2000.

[19] Y. Wexler, E. Shechtman, and M. Irani. Space-time video completion. In *Computer Vision and Pattern Recognition, 2004. CVPR 2004. Proceedings of the 2004 IEEE Computer Society Conference on*, volume 1, pages I–120. IEEE, 2004.