FLICKER REMOVAL AND SUPERPIXEL-BASED MOTION TRACKING FOR HIGH SPEED VIDEOS

Ali Kanj[1], *Hugues Talbot*[1] *and Raoul Rodriguez Luparello*[2]

[1]Université Paris-Est, Laboratoire d'Informatique Gaspard Monge
bd Blaise Pascal, Cité DESCARTES, BP 99, 93162 Noisy le Grand CEDEX, France
[2]Sublab Production, 52 Avenue Pierre Semard, 94200 Ivry-sur-Seine, France

ABSTRACT

Flicker removal consists of filtering out rapid, artefactual changes of luminosity and colorimetry from image sequences in order to improve colorimetry consistency between video frames. It is a necessary and fundamental task in multiple applications, for instance in archived film sequences, image/video compression and time-lapse videos. In recent years, the wider availability of fast video acquisition technology has renewed the interest in the flicker removal problem, in particular periodic flicker. In this context, rapid, undesirable intensity and chroma variations are due to acquisition hardware performing faster than the frequency of alternating current powering artificial light sources. This paper proposes both theoretical and efficient experimental solutions for flicker removal from image sequences by performing simultaneous motion tracking and color correction using a superpixel image representation.

***Index Terms*—** Flicker removal, Superpixel segmentation, Motion tracking, Spatial interpolation.

1. INTRODUCTION

Many methods for colorimetry stabilization in videos exist in the literature, for instance for underwater image sensing, surveillance systems, amateur videos, archived videos, image/video compression and time lapse videos. However, modeling the flicker effect differs from one application to the next. For instance, the sunflicker effect, a common challenge in underwater image sensing, stems from refracted sunlight casting fast moving patterns on the seafloornon and is non-periodic [1]. Flicker also commonly affects surveillance cameras in the presence of fluorescent lamps, making tracking persons more complicated. B. Ozer et al. [2] proposed to eliminate background which contains reflective surfaces and shadows, and track the regions of interest only. Consumer videos are often marred by scrolling stripes under artificial lighting, due to interference with the sampling frequency. In this case, flicker is periodic. In [3], authors used the temporal discrete Fourier transform (DFT) to estimate flicker parameters and adjust the acquisition sampling accordingly. In image sequence coding, flicker may appear when similar regions between two images are incoherently encoded. Ren et al.[4] proposed a block-matching, adaptive multiscale motion method. Existing methods for flicker removal in archived videos include both linear and non-linear models. Linear methods use an affine model to compensate flicker in archived videos [5, 6, 7, 8, 9]. In addition, non linear brightness distortion correction have been studied by [10, 11, 12, 13], which use histogram-matching algorithms to find flicker parameters. [14, 15] are pixel-wise methods based on a non-linear model.

Most state of the art methods on flicker removal consider only grey-level sequences and do not specifically deal with periodic flicker. They are typically based on classical regions of interest tracking. These model most motion as a local translation, as more complex models can be very expensive in terms of computation time.

However, using regions of interest is the most common but not the only tracking method. Using features of interest can be used instead. Features can have any shape but are most often associated with point locations in images. Points of interest are usually called keypoints [16, 17, 18]. In previous work, we proposed a rigid registration method using point descriptors that is robust to colorimetry variations, in which we stabilized these variations using a Least Square approach [19]. However, we found that tracking using only points can be difficult, depending on various factors related to the data or the application, for instance the level of noise, illumination and chroma variations, motions complexity, occlusions and so on. This deflickering method was only suitable in the presence of a single source illuminating the scene, and global rigid displacement between frames.

An object may also be represented by a simple shape such as a rectangle [20, 21, 22], an ellipse [23], etc. This representation is suitable for tracking rigid objects (i.e. vehicles). In this context, we have proposed a local flicker removal method, based on joint tracking/color correction scheme using a block matching technique paired with color variation estimation [24]. Edges can also be used to represent an object [25]. This representation can track non-rigid objects with a complex shape. However classical object segmentation may produce very sparse boundaries, which are insufficient to ensure consistent tracking as this sparsity magnifies the aperture problem. To ensure a suitable and adjustable boundary density, using superpixels has been proposed in [26]. In the context of flicker correction, superpixels can be exploited to avoid tracking featureless or noisy areas in video frames.

The objective of this paper is to study new avenues for video color correction, and typically for flicker removal applications. We propose a new local method for flicker removal in high speed videos based on superpixels segmentation, that is able to track objects with arbitrary complex shapes. We also propose a local color correction method based on sparse sampling and robust spatial interpolation.

The rest of this paper is organized as follows. In Section 2, we briefly formalize the strategy of flicker correction in videos. In section 3, we introduce a motion tracking approach based on SLIC superpixel segmentation. In section 4, we propose a color correction approach suitable for flicker removal and followed by a spatial interpolation step.Section 5, reports some experimental and comparative results between our method and a global approach for flicker correction, and finally, we conclude this paper and present ideas for future work in Section 6.

2. GENERAL STRATEGY FOR VIDEO CORRECTION

In this paper, we are concerned with correcting illumination artifacts associated with high-speed video acquisition under artificial lighting. At sampling rate higher than the mains power frequency (100 to 120Hz depending on the location, i.e. double the voltage frequency), lighting power fluctuations become noticeable. To address this problem, we assume that we know the acquisition frame rate and the mains AC frequency. We therefore can estimate the period of the flickering effect. We also assume that general intensity variation integrated over a few flickering periods is slow. Given this, it is easy to find the two successive peaks of illumination intensity in the local period. In Fig. 1, the acquisition frame rate is 1000 images/second and the AC mains power frequency is 100Hz. As expected, the period of the flicker is 10 frames long.

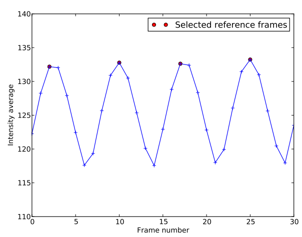


Fig. 1: The blue curve shows the luminosity variation in an affected image sequence.

2.1. Selecting reference frames

Any frame in a flicker period could serve as a reference, however, it is easier to detect peak illumination in a period, and use the last maximum intensity frame in a period, at time t_{ref}, as a reference image. This frame may be considered a good choice in terms of signal-to-acquisition noise ratio.

In this target, we calculated the average of pixel intensities in each image along the sequence, so a selected reference frame in such a flicker period should be the one having the highest average in this period.

2.2. Flicker correction model

Referring to previous works in [19, 24], we can model the color illumination transform by a 3×3 matrix $\boldsymbol{M}_{\mathbf{s},t}$ between two pixels $\mathbf{s}$ and $\mathbf{s}'$ referring to the same physical area in the scene at times t and t_{ref}:

$$\boldsymbol{M}_{\mathbf{s},t} = \begin{bmatrix} r_1(\mathbf{s},t) & g_1(\mathbf{s},t) & b_1(\mathbf{s},t) \\ r_2(\mathbf{s},t) & g_2(\mathbf{s},t) & b_2(\mathbf{s},t) \\ r_3(\mathbf{s},t) & g_3(\mathbf{s},t) & b_3(\mathbf{s},t) \end{bmatrix}. \tag{1}$$

Using this matrix, we have

$$\mathbf{f}(\mathbf{s}', t_{\text{ref}}) = \boldsymbol{M}_{\mathbf{s},t}\,\mathbf{f}(\mathbf{s},t) + \mathbf{w}(\mathbf{s},t), \tag{2}$$

where $\mathbf{f}(\mathbf{s},t) \in \mathbf{R}^3$ is the vector of chrominance values of pixel $\mathbf{s}$ at time t and $\mathbf{w}(\mathbf{s},t) \in \mathbf{R}^3$ is a term accounting for acquisition and modelling noises.

Our proposed method is implemented as a two-step procedure. In a first stage, regions of interest are tracked from the reference image to the target images in order to estimate a color correction matrix. Secondly, an interpolation between the color estimates is performed to process the unknown estimated points.

3. MOTION TRACKING

Construction of superpixels on the images allows us to work locally on the extracted information. Our method begins by calculating a superpixels segmentation of the reference image $\mathbf{f}(t_{\text{ref}})$ and the target images $\mathbf{f}(t)$. For this, we initialize our algorithm with the Simple Linear Iterative Clustering algorithm [27], which is briefly explained below.

3.1. SLIC segmentation

The Simple Linear Iterative Clustering algorithm clusters pixels in the combined five-dimensional color and image plane space to efficiently generate compact, nearly uniform superpixels.

It allows to generate superpixels regularly on the image surface. First, the centers of superpixels are initialized on a regular grid, spaced S pixels, with $S = \sqrt{\frac{K}{N}}$ where N is the total number of desired superpixels and K is the approximative number of superpixels. They can be optionnaly moved to avoid being on image boundaries.

SLIC is an iterative method consisting of two steps:

- assigning pixels to a center C_k basing on a similarity criteria,
- updating of the centers.

This approach seeks to minimize; in the first step; the similarity criteria corresponding to a distance between C_k and the current pixel p, defined by:

$$D^2_{SLIC}(C_k, p) = d^2_{lab}(C_k, p) + \frac{d^2_{xy}(C_k, p)}{S^2} m^2 \tag{3}$$

where m is the compactness, when m is large, spatial proximity is more important and the resulting superpixels are more compact (i.e. they have a lower area to perimeter ratio). When m is small, the resulting superpixels adhere more tightly to image boundaries, but have less regular size and shape. When using the CIE$L^*a^*b^*$ color space, m can be in the range $[1, 40]$. d_{lab} is a colorimetric distance and d^2_{xy} is the geometric distance between two positions in the current image,

$$d^2_{lab}(p_i, p_j) = (l_i - l_j)^2 + (a_i - a_j)^2 + (b_i - b_j)^2 \tag{4}$$

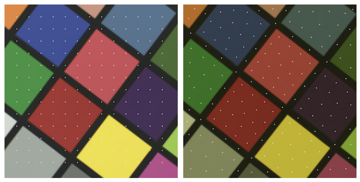
$$d^2_{xy}(p_i, p_j) = (x_i - x_j)^2 + (y_i - y_j)^2 \tag{5}$$

For each center, a similarity search area is defined, of $2S \times 2S$ size and cetered on C_k. Only pixels of this zone are examined.

Thereafter, we show how we can track the motion using this segmentation process.

3.2. Label tracking

For a specific flicker period, the reference frame is segmented using the SLIC algorithm using a regular initialization of superpixels centers (See Figure 2). Updated centers are associated for all superpixels at the end of the segmentation process. Each superpixel is



(a) (b)

Fig. 2: Centers initialization for the reference frame segmentation (a) and a subsequent frame (b).

identified by its own label number and its center coordinates. SLIC parameters for the segmentation of next subsequent frames are initialized with the last centers grid obtained from the segmentation of the previous frame. This improves the likelihood that a tracked superpixel maintains the same label number during the whole flicker period. A validation test is performed in order to verify that the same label number in two consecutive images does indeed correspond to the same region of interest. In our context, displacements are considered over a small range. As a first check, a validation test depends on the displacement value between the corresponding matched superpixels centers, i.e. if this value is higher than a given threshold, tracking of the corresponding superpixel is considered to have failed. It will therefore be ignored for the color correction estimation. We also assume that a given superpixel retains the same neighbors between two successive frames to validate its matching procedure. Once a superpixel is successfully tracked and maintains

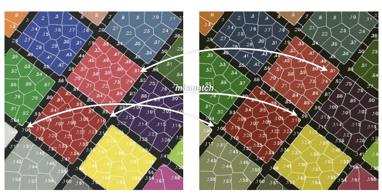


Fig. 3: Example of some good and bad matches between segmented reference and target frames.

its label number between the reference frame and the current frame, a color correction estimation is subsequently performed.

4. COLOR CORRECTION STEP

Due to the presence of motion and illumination variations between the segmented frames, the shape and area of a superpixel S_k at label k can vary significantly. Thus, a color distribution comparison between superpixels with the same label would not yield good results.

In addition, the SLIC algorithm classifies the regions of interest in images based on a color similarity criteria. As a result, superpixels tend to be homogeneous in color. This implies that attempting to correlate the color content of superpixels between frames would be a poorly conditioned match. We observed that classical histogram-based approaches fail in the context of superpixel matching.

We propose a ROI based color comparison in order to estimate the colorimetric transformation for each valid tracked superpixel.

4.1. ROI based color matching

We consider a superpixel $S_k(t)$ in a current frame $f(t)$ and centered by $C_k(s,t)$, that is successfully tracked from its initial position centered on $C_k(s+\widehat{\mathbf{d}}_S, t_{ref})$ in the reference frame $f(t_{ref})$. $\widehat{\mathbf{d}}_S$ is the displacement vector between the initial k label position in the reference frame and its position in the current fame. We then compare two ROI regions $B_k(t)$ and $B_k(t_{ref})$ surrounding $C_k(t)$ and $C_k(t_{ref})$ respectively. The flicker matrix is estimated by minimiz-

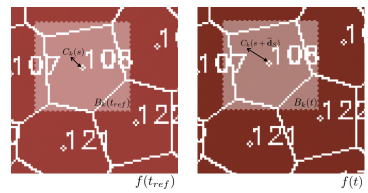


Fig. 4: Pixel wise matching for similar superpixels.

ing the following energy function

$$J(\widetilde{\boldsymbol{M}}_{k,t}, \mathbf{d}_S) = \sum_{\mathbf{s}\in\mathcal{B}_k} \Phi\big(\widetilde{\boldsymbol{M}}_{k,t}\,\mathbf{f}(\mathbf{s},t) - \mathbf{f}(\mathbf{s}-\mathbf{d}_S, t_{\text{ref}})\big), \quad (6)$$

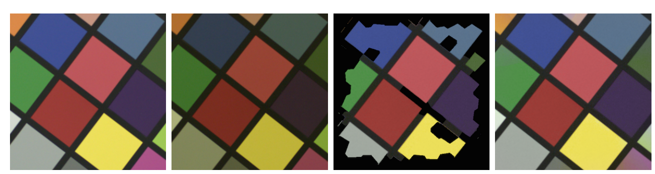
where $\Phi\colon \mathbb{R}^3 \to [0,+\infty[$ is a cost function. The simplest and fastest choice for Φ is the quadratic norm where $\Phi = \|\cdot\|^2/2$. The quadratic solution is given by

$$\widehat{\boldsymbol{M}}_{k,t} = \Big(\sum_{\mathbf{s}\in\mathcal{B}_k} \mathbf{f}(\mathbf{s}-\mathbf{d}_S, t_{\text{ref}})\mathbf{f}(\mathbf{s},t)^\top\Big)\cdot\Big(\sum_{\mathbf{s}\in\mathcal{B}_k} \mathbf{f}(\mathbf{s},t)\,\mathbf{f}(\mathbf{s},t)^\top\Big)^\dagger. \quad (7)$$

Once a color correction matrix $\widehat{\boldsymbol{M}}_{k,t}$ is computed for a successfully tracked superpixel, we apply it uniformly on all superpixel pixels:

$$\forall \mathbf{x}\in S_k(t)\,,\ \mathbf{f}_p(\mathbf{x},t) = \widehat{\boldsymbol{M}}_{k,t}\times\mathbf{f}(\mathbf{x},t), \quad (8)$$

where $\mathbf{f}_p(t)$ is the processed frame at time t.



(a) Reference (b) Degraded (c) Correction (d) Interpolation

Fig. 5: Superpixel based method for color correction: In (c) only superpixels with succesfull tracking are processed, other superpixels are ignored. In (d) interpolation of the correction matrix provides a solution for the whole image.

Figure 5 shows that post processing is an essential step in order to estimate the unknown color correction parameters for the ignored superpixels at the first stage, and thus to perform the color correction on the whole image, and also to eliminate some superpixel borders artifacts if they exist.

4.2. Post processing step

Post-processing consists of finding the unknown transformation estimates for the remaining superpixels on the one hand, and to remove borders artifact on the other. For this we interpolate the transformation matrix over the whole image using a kriging model [28].

5. RESULTS AND DISCUSSION

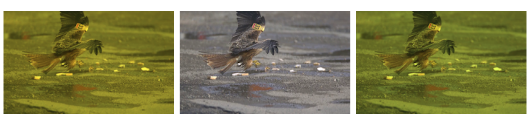
The proposed approach was tested on real studio-lit videos affected with local flicker and various motion including rotations (see Fig. 6). We also validate our method on a synthetic flicker sequence in order to be able to compute similarity measures between original and processed sequences (see Fig. 7). In Figures 6 and 7, (b) presents the first processing step on the successfully tracked superpixels, the black regions are the ignored superpixels. (c) shows the post-processing step based on the spatial interpolation of color correction parameters, it shows that the color correction of all image regions are succesfully predicted.

(a) Degraded frames

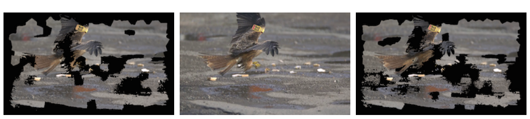
(b) Superpixel results

(c) Post-processing

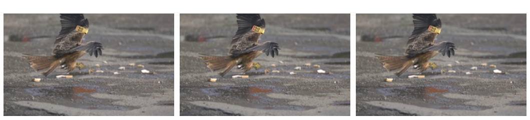
Fig. 6: Video 1: Superpixel-based method for flicker removal.



(a) Degraded frames



(b) Superpixel results



(c) Post-processing

Fig. 7: Video 2: Superpixel-based method for flicker removal on synthetic flicker sequence.

We compared our proposed approach to a state-of-the-art local color correction algorithm which is based on joint block/keypoints tracking [29]. Figure 8 plots the luminosity variation of processed images of Video 1 vs. frame index, it shows that flickering is much reduced along the sequence, even if some very small, imperceptible fluctuations remain. Figure 9 indicates a significant image quality improvement after our processing. Comparing the global color correction method to the proposed approach, the PSNR average increases from 42 db to about 51 db. These show that our approach is promising for simultaneous motion tracking and color correction for high speed imaging.

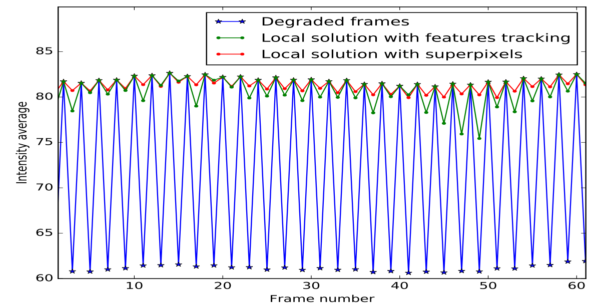


Fig. 8: Luminosity variation of Video 2: Test of performance of superpixel based method

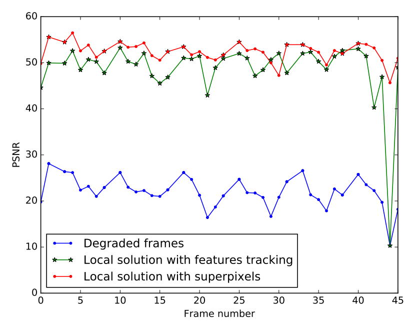


Fig. 9: PSNR similarity measure on Video 6: Test of performance of superpixel based method on synthetic flicker sequence.

6. CONCLUSION

In this paper, we were inspired by the superpixel-based segmentation algorithms for motion tracking in image sequences in the presence of brightness and chromatic changes. Based on this tracking, a pixel-wise matching approach was established to compare the color distributions between tracked regions of interest from a reference image to a target one in order to locally estimate flicker parameters in an image sequence.

The results show that our superpixel-based method yields good results, especially in the color distribution matching step. Using multiple constraints based on the neighboring superpixels allows for a reliable, though sparse, tracking method. However, superpixels need to be dense enough to precisely track the motion, and this can be demanding in terms of computation time in very high definition videos (8k or more). Several improvements are planned as a future work, such as using a 3D histogram matching using optimal transport methods between similar superpixels for predicting flicker parameters [30], which could accelerate the procedure and improve the accuracy of flicker parameter estimation.

7. REFERENCES

[1] A. S. M. Shihavuddin, N. Gracias, and R. García, "Online sunflicker removal using dynamic texture prediction.," in *International Conference on Computer Vision Theory and Applications*, Rome, Italy, 2012, pp. 161–167.

[2] B. Ozer and M. Wolf, "A train station surveillance system: Challenges and solutions," in *Computer Vision and Pattern Recognition Workshops*, Columbus, Ohio, 2014, IEEE, pp. 652–657.

[3] S. Baudry, B. Chupeau, M. De Vito, and G. Doerr, "Modeling the flicker effect in camcorded videos to improve watermark robustness," in *National Conference on Parallel Computing Technologies*, Bengaluru, India, 2015, IEEE, pp. 42–47.

[4] J. Ren, J. Liu, M. Li, and Z. Guo, "Postprocessing of block-coded videos for deflicker and deblocking," in *IEEE International Conference on Acoustics, Speech and Signal Processing*, Vancouver, BC, Canada, 2013, pp. 1631–1635.

[5] C-S. Fuh and P. Maragos, "Motion displacement estimation using an affine model for image matching," *Optical Engineering*, vol. 30, no. 7, pp. 881–887, 1991.

[6] É. Decencière, *Restauration automatique de films anciens*, Ph.D. thesis, École Nationale Supérieure des Mines de Paris, France, 1997.

[7] P. M. B. Van Roosmalen, R. L. Lagendijk, and J. Biemond, "Correction of intensity flicker in old film sequences," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 9, no. 7, pp. 1013–1019, 1999.

[8] A. C. Kokaram, R. Dahyot, F. Pitié, and H. Denman, "Simultaneous luminance and position stabilization for film and video," in *Electronic Imaging 2003*, San Jose, California, USA, 2003, International Society for Optics and Photonics, pp. 688–699.

[9] X. Zhang, A. B. E. Masahide, and M. Kawamata, "Flicker parameters estimation in old film sequences containing moving objects," *IEICE TRANSACTIONS on Fundamentals of Electronics, Communications and Computer Sciences*, vol. 94, no. 12, pp. 2836–2844, 2011.

[10] V. Naranjo and A. Albiol, "Flicker reduction in old films," in *IEEE International Conference on Image Processing*, Vancouver, Canada, 2000, vol. 2, pp. 657–659.

[11] J. Delon, "Movie and video scale-time equalization application to flicker reduction," *IEEE Transactions on Image Processing*, vol. 15, no. 1, pp. 241–248, 2006.

[12] F. Pitié, R. Dahyot, F. Kelly, and A. C. Kokaram, "A new robust technique for stabilizing brightness fluctuations in image sequences," in *Statistical Methods in Video Processing*, pp. 153–164. Springer Berlin Heidelberg, 2004.

[13] G. Forbin and T. Vlachos, "Nonlinear flicker compensation for archived film sequences using motion-compensated graylevel tracing," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 18, no. 6, pp. 803–816, 2008.

[14] F. Pitié, B. Kent, B. Collis, and A. C. Kokaram, "Localised deflicker of moving images," in *European Conference on Visual Media Production*, London, United Kingdom, 2006, pp. 134–143.

[15] F-J. Hampson and J-C. Pesquet, "Motion estimation in the presence of illumination variations," *Signal processing: Image communication*, vol. 16, no. 4, pp. 373–381, 2000.

[16] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International journal of computer vision*, vol. 60, no. 2, pp. 91–110, 2004.

[17] S. Leutenegger, M. Chli, and R. Y. Siegwart, "Brisk: Binary robust invariant scalable keypoints," in *IEEE International Conference on Computer Vision*, Barcelona, Spain, 2011, pp. 2548–2555.

[18] H. Bay, A. Ess, T. Tuytelaars, and L. Van Gool, "Speeded-up robust features (surf)," *Computer vision and image understanding*, vol. 110, no. 3, pp. 346–359, 2008.

[19] A. Kanj, H. Talbot, and R. Rodriguez Luparello, "Global image registration and video color correction in presence of illumination variations," in *Fifth International Conference on Digital Information and Communication Technology and its Applications*, Beirut, Lebanon, 2015, IEEE, pp. 92–95.

[20] A. Adam, E. Rivlin, and I. Shimshoni, "Robust fragments-based tracking using the integral histogram," in *IEEE Computer Society Conference on Computer vision and pattern recognition*, New York, NY, USA, 2006, vol. 1, pp. 798–805.

[21] W. Zhong, H. Lu, and M-H. Yang, "Robust object tracking via sparsity-based collaborative model," in *IEEE Conference on Computer vision and pattern recognition*, Providence, Rhode Island, 2012, pp. 1838–1845.

[22] X. Jia, H. Lu, and M-H. Yang, "Visual tracking via adaptive structural local sparse appearance model," in *IEEE Conference on Computer vision and pattern recognition*, Providence, Rhode Island, 2012, pp. 1822–1829.

[23] S. Birchfield, "Elliptical head tracking using intensity gradients and color histograms," in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, California, USA, 1998, pp. 232–237.

[24] A. Kanj, H. Talbot, J-C. Pesquet, and R. Rodriguez Luparello, "Color deflickering for high-speed video in the presence of artificial lighting," in *International Conference on Image Processing*, Quebec city, Canada, September 2015, IEEE, pp. 976–980.

[25] P. Smith, T. Drummond, and R. Cipolla, "Layered motion segmentation and depth ordering by tracking edges," *Transactions on Pattern Analysis and Machine Intelligence*, vol. 26, no. 4, pp. 479–494, 2004.

[26] S. Wang, H. Lu, F. Yang, and M-H. Yang, "Superpixel tracking," in *IEEE International Conference on Computer Vision*, Barcelona, Spain, 2011, IEEE, pp. 1323–1330.

[27] R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, and S. Süsstrunk, "SLIC superpixels compared to state-of-the-art superpixel methods," *IEEE transactions on pattern analysis and machine intelligence*, vol. 34, no. 11, pp. 2274–2282, 2012.

[28] Georges Matheron, *The theory of regionalized variables and its applications*, vol. 5, École national supérieure des mines, 1971.

[29] S. Battiato, G. Gallo, G. Puglisi, and S. Scellato, "SIFT features tracking for video stabilization," in *IEEE International Conference on Image Analysis and Processing*, Modena, Italy, 2007, pp. 825–830.

[30] Sira Ferradans, Nicolas Papadakis, Gabriel Peyré, and Jean-François Aujol, "Regularized discrete optimal transport," *SIAM Journal on Imaging Sciences*, vol. 7, no. 3, pp. 1853–1882, 2014.