Video shutter angle estimation using optical flow and linear blur

David Korčák   Jiří Matas

Faculty of Electrical Engineering

Czech Technical University in Prague

korcadav@fel.cvut.cz   matas@fel.cvut.cz

Abstract

We present a method for estimating the shutter angle, a.k.a. exposure fraction – the ratio of the exposure time and the reciprocal of frame rate – of videoclips containing motion. The approach exploits the relation of the exposure fraction, optical flow, and linear motion blur. Robustness is achieved by selecting image patches where both the optical flow and blur estimates are reliable, checking their consistency. The method was evaluated on the publicly available Beam-Splitter Dataset with a range of exposure fractions from 0.015 to 0.36. The best achieved mean absolute error of estimates was 0.039. We successfully test the suitability of the method for a forensic application of detection of video tampering by frame removal or insertion.

1. Introduction

The shutter angle, a.k.a the exposure fraction, is the ratio of the exposure time, i.e. the time period a film or a sensor is exposed to light, and the time between two consecutive frames, i.e. the reciprocal of the frame rate. The shutter angle determines the relation between object motion and image blur and thus influences viewer perception. This has been used in film-making as part of artistic expression and tutorials and websites are devoted to explaining the effect [6]. In computer vision and image processing, exposure fraction affects methods for temporal super-resolution and video frame interpolation, since the inserted images need to both interpolate motions and reduce motion blur.

For analog cameras, the exposure fraction remains constant throughout the duration of the video. In digital cameras, the exposure time and thus the shutter angle may be set dynamically, according to illumination intensity. Nevertheless, for most recorded videos it stays constant. For global shutter cameras, it is the same for every pixel of a frame. For rolling shutter cameras the same is true for horizontal motions; the exact modeling is more complex for vertical motions.

The exposure fraction provides a physics-based constraint that influences every pixel, and it thus has the potential for the detection of fake videos and local image edits. Violations of the constraint – the linear relationship between the local motion blur and optical flow with the shutter angle providing the scaling constant – are not immediately obvious to human observers, and thus might not be noticed by neither the authors nor the viewers of the altered or synthesized content. Moreover, many generators of synthetic content are often trained on sharp images corresponding to very short exposures or on graphics-generated data and thus might not represent motion blur as required by physics.

In the paper, we present a robust method for the estimation of the shutter angle that relies on explicitly running a state-of-the-art optical flow algorithm [11] and linear blur kernel estimator [2]. We are not aware of any existing method for shutter angle estimation from unconstrained video sequences. Barber *et al*. [1] formulate the problem for sequences containing only specific types of blur. We are the first to address the problem for sequences containing general motion.

In summary, we make the following contributions. The proposed method is novel, exploiting recent progress in deep optical flow and linear blur estimation. Both of these estimates are dense, permitting to achieve robustness by combining predictions from patches where both estimates are reliable and consistent. We show a forensic application of the method, considering detection of video tampering by frame removal or insertion.

2. Related work

The problem of estimating the shutter angle of a videoclip has been approached from the point of view of precisely measuring camera characteristics with the help of a bespoke setup. The method of Simon et al. [8] relies on special devices, such as turntables, CRT displays or arrays of LEDs lit in specific patterns. Barber *et al*. [1] addressed the problem for sequences containing blur induced by either zoom or camera rotation during exposure.

We formulate the problem of estimating video shutter angle with the use of optical flow and linear blur kernel esti-

mates from a general video clip containing motion. Methods for estimating linear blur parameters [2, 10, 13] often rely on deep neural networks trained on synthetically blurred images. Typically, they serve as an intermediate step towards image or video frame deblurring. While the details of particular implementations differ, the output is generally a set of estimated parameters of linear blur kernels.

Similarly, the topic of optical flow estimation has seen a lot of progress with methods such as Teed and Deng [11]'s RAFT. Optical flow methods seek to estimate pixel displacements between consecutive frames, and as of recently, are based on various deep neural network architectures. As obtaining ground truth of optical flow on real-world data is difficult, these methods are trained on synthetic datasets.

3. Method

The proposed method for shutter angle estimation relies on the calculation of dense optical flow, described in Sec. 3.1, and linear blur, described in Sec. 3.2. The shutter angle is the ratio of the length of blur and optical flow vectors (Sec. 3.4) if the direction of motion does not change within the exposure time, thus not all parts of the image are suitable for estimating this ratio. For instance, both estimates of the blur and the optical flow will be unreliable in parts of the image that contain the sky or similar smooth-texture surfaces. In Sec. 3.5 we describe our algorithm for the selection of patches suitable for shutter angle estimation.

3.1. Optical flow model

Optical flow is conventionally defined as per-pixel motion between video frames. Given 2 consecutive video frames F_i and F_{i+1}, the goal of optical flow estimation methods is to map the position (x_i, y_i) of a given pixel in frame F_i to its position (x_{i+1}, y_{i+1}) in frame F_{i+1}. This mapping can be modeled as a dense displacement field $(\mathbf{f^1}, \mathbf{f^2})$. The position of a given pixel in frame F_{i+1} can be then described as $(x_i + f^1(x_i), y_i + f^2(y_i))$ [11]. In this paper, we employ Teed and Deng's RAFT [11] for optical flow estimation, selected for its well-documented performance across various sequences [11] and strong benchmark results [12].

3.2. Linear blur model

The heterogeneous motion blur model commonly views the blurred (real) image $\mathbf{Y}$ as the product of a convolution of a sharp image $\mathbf{X}$ with an operator $\mathbf{K}$ and additive noise $\mathbf{N}$ [2].

$$\mathbf{Y} = \mathbf{K} * \mathbf{X} + \mathbf{N} \tag{1}$$

The motion blur kernel map, $\mathbf{K}$, in general, consists of different blur kernels for each pixel at position (x, y).

The linear blur assumes the kernels at (x, y) can be modeled as 1D, in the direction of the local motion. Such kernels can also be interpreted as two-dimensional motion vectors $\mathbf{K_{x, y}} = (\mathbf{k^1_{x, y}}, \mathbf{k^2_{x, y}})$. Linear blur kernels also characterize the motion of a pixel over the camera exposure time ε, as the blurring occurs by motion during camera light capture over the exposure period.

The assumption of linear blur is violated e.g. for hand-held cameras that may undergo Brownian-like motions. In such cases, the estimation of both blur kernels and optical flow is difficult and they present a challenge for our method. Since the exposure fraction is the same for all pixels in the image, it is sufficient to find a modest number of areas where the linear blur assumption holds, e.g. on a linearly moving object in the scene. In Sec. 3.5, we introduce techniques for identifying and selecting such areas of video frames.

In this paper, we apply the method of linear blur estimation by Gong *et al.* [2], which shows both good generalization ability and dataset performance. It is also one of the only methods that perform per-pixel estimates of blur kernels, i.e. the estimates are calculated, not interpolated from patches, in full resolution. This method, however, introduces a level of discretization error, as the deep neural network used for estimating blur kernels operates as a multi-class classifier with a discrete output space. We attempt to minimize the effect of such errors on our final estimate as described in Sec. 3.5.

During testing, we also evaluated the performance of Zhang *et al.*'s method [13] as it operates with a real output space but found that in our set up it performed worse than Gong *et al.*'s method [2].

3.3. Shutter angle from linear blur and optical flow

We consider a video camera with the following parameters. Let ε denote the exposure time of each frame, f the video framerate in frames per second and θ the video shutter angle in degrees. For the i-th video frame, we define t_i as the time of exposure start and $t_i + \varepsilon$ as the exposure end. The time difference between exposure starts of two consecutive frames, $\Delta t = t_{i+1} - t_i = 1/f = f^{-1}$, is equal to the reciprocal of the frame rate f.

For the purpose of simplicity, we use the exposure fraction notation, rather than the shutter angle, i.e. instead of $180°$ we speak of 0.5. The degree notation is widely used in cinematography, as it originates from the construction of historical cameras that utilized mechanical rotating shutters to set the exposure time ε. In many modern digital cameras, exposure time ε can be set explicitly. We define ***exposure fraction*** α as the ratio

$$\alpha = \frac{\varepsilon}{\Delta t} = \frac{\theta}{360°}. \tag{2}$$

3.4. Estimating the exposure fraction

Consider optical flow described in Sec. 3.1 and linear blur kernel described in Sec. 3.2. The magnitude of the optical flow vector $\|\mathbf{f_{x,y}}\|$ is equivalent to the distance displaced by pixel over the duration of a single frame, i. e. over the time interval of length Δt. Similarly, the magnitude of linear blur kernel $\|\mathbf{K_{x,y}}\|$ corresponds to the distance displaced by pixel (x, y) over the time interval ε. Here, we assume uniform motion, i. e. the pixel at (x, y) is traveling at a constant velocity $v_{x,y}$ between consecutive frames. Such assumption is reasonable, as the absolute frame duration Δt of video clips shot at multiple frames per second is often negligible compared to camera motion or motion of common objects. Under this assumption, we may express the norm of optical flow vector and linear blur kernel as distance displaced by pixel (x, y) at constant velocity $v_{x,y}$ over respective time intervals.

$$\|\mathbf{f_{x,y}}\|_2 = v_{x,y} \cdot \Delta t, \|\mathbf{K_{x,y}}\|_2 = v_{x,y} \cdot \varepsilon \qquad (3)$$

We substitute in Eq. (2) and obtain the following

$$\alpha_{x,y} = \frac{\|\mathbf{K_{x,y}}\|_2}{\|\mathbf{f_{x,y}}\|_2} \; ; \qquad (4)$$

i.e. given the magnitude of the optical flow $\mathbf{f_{x,y}}$ and linear blur kernel $\mathbf{K_{x,y}}$ at pixel (x, y) the value of α at position (x, y) as a ratio of magnitudes of the two vectors.

3.5. Computation

The proposed method for estimating the value of α builds on Sec. 3.4. As described in Sec. 1, modern video cameras operate either in a global shutter mode, where all pixels of the frame get exposed at the same point in time or, more commonly, in a rolling shutter mode, where exposure is performed row-wise. In both cases, the time of exposure ε remains constant for all pixels in the frame. Similarly, the time interval between exposures of pixels (both global and rolling shutter) remains constant. As a result of these physical constraints, the value of α must be consistent in an entire frame, and typically in the entire video clip. Therefore, the problem of estimating the value $\alpha_{x,y}$ pixel-wise is reduced to estimating one global value α_{glob} for the entire video clip.

As the sources of both optical flow and linear motion blur are not robust and prone to errors in their estimates, and the condition of motion in a single direction during exposure time may not be satisfied, the proposed method locates patches of pixels with the lowest error potential in both linear blur kernels and optical flow. We define multiple constraints and show that are effective in filtering erroneous predictions.

First, we discard estimates at all positions (x, y), where the angle between the linear blur kernel and optical flow vectors exceeds a threshold. The condition is evaluated in

$\varepsilon\,(ms)$	1	2	3	8	16	24
α	0.015	0.030	0.045	0.120	0.240	0.360

Table 1. Exposure fractions of BSD subsets based on exposure time ε. All videoclips have a framerate $f = 15$ FPS, $\Delta t = 0.0\overline{66}\,s$

the cosine domain which avoids wrap-around effects and also addresses the problem that the blur kernel is estimated modulo π, it does not have a direction:

$$\frac{|\langle \mathbf{f_{x,y}} | \mathbf{K_{x,y}} \rangle|}{\|\mathbf{f_{x,y}}\|_2 \cdot \|\mathbf{K_{x,y}}\|_2} \geq \cos\varphi \qquad (5)$$

where φ is the maximum angle threshold in degrees; $\langle \cdot | \cdot \rangle$ denotes the dot product.

Second, for Eq. (4) it follows that the norm of the ground truth optical flow vector is always larger than the norm of the ground truth linear blur kernel – a pixel cannot be physically captured for a longer period than the maximum inter-frame period Δt. We therefore remove all values from positions (x, y) where:

$$\|\mathbf{K_{x,y}}\|_2 > \|\mathbf{f_{x,y}}\|_2 \qquad (6)$$

The blur kernel estimation method [2] outputs values in a discrete domain, with the discretization error introduced in both vertical and horizontal directions equal to 1. This renders predictions with small motions arbitrarily, and we thus remove positions with small flow and blur magnitudes:

$$\|\mathbf{K_{x,y}}\|_2 \leq 1, \|\mathbf{f_{x,y}}\|_2 \leq 1 \;. \qquad (7)$$

Next, we find a D x D patch that contains the highest number of valid positions. The value of α for a given frame is estimated from this patch. We calculate the estimate of $\alpha = \alpha_{\text{patch}}$ for the current frame as the ratio of norms of means of linear blur kernels and optical flow vectors

$$\alpha_{\text{patch}} = \frac{\|\frac{1}{N} \sum\limits_{i=1}^{N} \mathbf{K_{x_i,y_i}}\|_2}{\|\frac{1}{N} \sum\limits_{i=1}^{N} \mathbf{f_{x_i,y_i}}\|_2} \qquad (8)$$

where N is the number of valid positions (x, y) inside of the selected patch.

Finally, we calculate α_{glob} as the median of estimates of all individual frames

$$\alpha_{\text{glob}} = \text{med}\{\alpha_{\text{patch}_1}, \alpha_{\text{patch}_2}, ..., \alpha_{\text{patch}_N}\} \qquad (9)$$

where N is the number of frames in the video clip.

	ε (ms)	1	2	3	8	16	24	Average
D	φ							
10	3°	0.058	0.034	0.035	0.024	0.048	0.091	0.049
10	5°	0.054	0.030	0.033	0.025	0.052	0.095	0.048
10	7°	0.051	0.030	0.032	0.026	0.055	0.096	0.048
20	3°	0.054	0.033	0.031	0.023	0.041	0.080	0.044
20	5°	0.048	0.029	0.029	0.022	0.042	0.081	0.042
20	7°	0.046	0.029	0.029	0.026	0.043	0.082	0.042
30	3°	0.054	0.033	0.031	0.022	0.038	0.077	0.042
30	5°	0.047	**0.028**	**0.028**	**0.020**	0.035	0.075	**0.039**
30	7°	**0.045**	0.029	0.029	0.025	**0.034**	**0.072**	**0.039**

Table 2. BSD dataset - mean absolute error of exposure fraction $\hat{\alpha}$ estimates for a range of patch sizes D and the tolerated angular difference φ between the optical flow and blur directions. The best results, in bold, were achieved for the largest window size and angular tolerance of 5-7 °.

4. Experiments and evaluation

In this section, after presenting the values of the two parameters of the proposed method - the patch size D and the angular threshold φ, we perform testing on a public dataset and in-depth experiments on individual video clips. We investigate both well-performing video clips and failure cases in an attempt to find the limitations of the proposed method.

4.1. Parameter selection

We tested all configurations with patch sizes of $D = \{10, 20, 30\}$ and angular constraints, $\varphi = \{3°, 5°, 7°\}$. For optical flow estimates, we utilized RAFT model [11] pretrained on the Sintel dataset with 12 iterations per two consecutive frames. The results are summarized in Tab. 2.

4.2. Evaluation datasets

Finding a suitable dataset was difficult, as the exposure time data is often erased from video clip metadata or not saved by the video camera at all. Many popular video datasets such as GoPro [7] or DeepVideoDeblurring Dataset [9] are either stripped of this information or are available as individual frames, converted post-capture with camera details unavailable.

The largest public dataset containing exposure time data for every video clip is the Beam-Splitter Dataset (BSD) [14]. The Beam-Splitter Dataset consists of pairs of identical videoclips captured with different exposure settings by 2 independently controlled cameras. Due to the framerate f being known, we are able to calculate the ground-truth of α for each videoclip. We test on the full 600-videoclip dataset. Exposure parameters for each subset are in Tab. 1. There are 100 videoclips in each distinct subset. We used this dataset for quantitative testing as well as qualitative results on well-performing video clips and failure cases.

4.3. Estimation of exposure fraction on the BSD dataset

We performed quantitative evaluation on all subsets of the BSD dataset for all parameter combinations mentioned in Sec. 4.1. We use Mean Absolute Error (MAE) as the performance measure:

$$MAE = \frac{1}{N} \sum_{i=1}^{N} |\alpha - \hat{\alpha}_i| \qquad (10)$$

Results are presented in Tab. 2. We observe a mild positive relationship between method accuracy (MAE) and the increasing size of selected patches D. Testing also shows that a more relaxed cosine constraint φ leads to a lower error for all fixed patch sizes. We attribute this to the property of the adopted linear blur estimation method, which occasionally produces results with a correct magnitude but incorrect orientation or vice versa, further amplified by its discrete output space [2].

The estimated $\hat{\alpha}$ is less accurate for very small and very large values of the true exposure fraction α. Analysis of the behavior is a part of our future work. We conjecture that for very low values of α, the discrete estimates of blur are highly inaccurate. For large values of α, the optical flow, operating on blurred images, is possibly losing accuracy.

Fig. 1 and Fig. 2 show the distribution of $\hat{\alpha}$ from a test with parameters $\varphi = 5°$, $D = 30$. Larger levels of noise are present in the estimates of $\alpha < 0.1$. This supports our conjecture that the accuracy of linear blur kernel estimation on very small values of α is rather low. For values $\alpha > 0.1$,
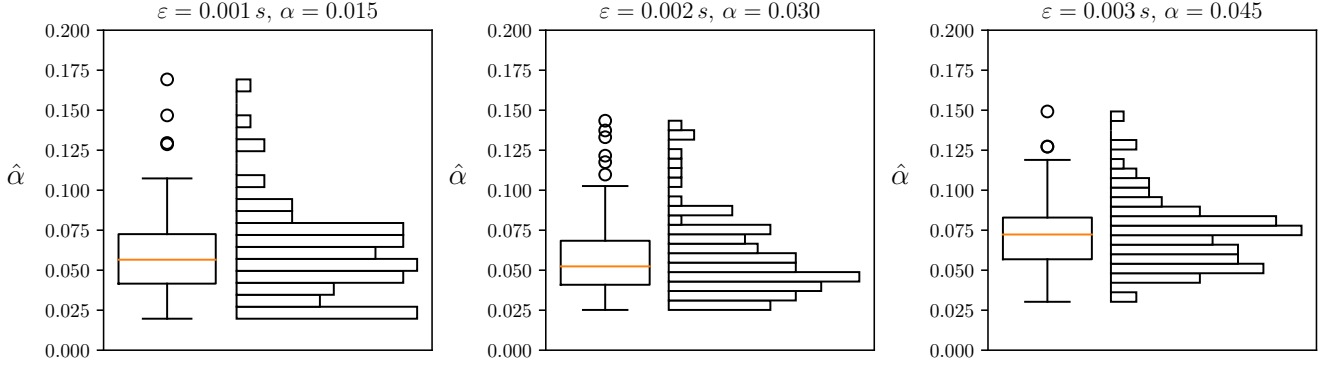
Figure 1. Histograms and box plots of $\hat{\alpha}$ estimates on clips from the BSD dataset with $\varepsilon = \{0.001\,s, 0.002\,s, 0.003\,s\}$. Estimation parameters $\varphi = 5°$, $D = 30$. Note that only the $(0, 0.2)$ range is displayed.
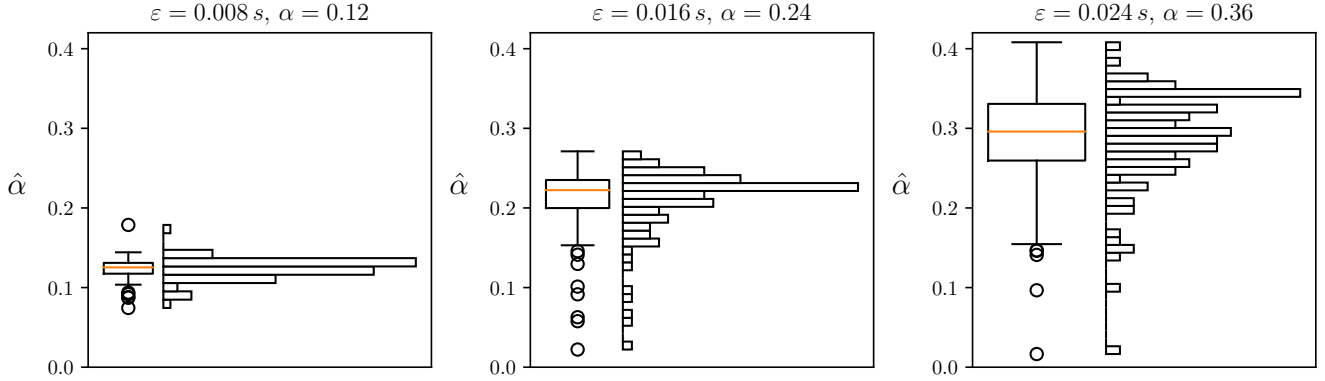

Figure 2. Histograms and box plots of $\hat{\alpha}$ estimates on clips from the BSD dataset with $\varepsilon = \{0.008\,s, 0.016\,s, 0.024\,s\}$. Estimation parameters $\varphi = 5°$, $D = 30$. Note that only the $(0, 0.4)$ range is displayed.

we observe the majority of estimates within close intervals of ground truth values. The dependency of estimation accuracy on the ground truth value of α is the largest limitation of the method.

4.4. Qualitative analysis of results on BSD dataset

From the estimates performed with parameters $\varphi = 5°$, $D = 30$ detailed in Fig. 1, Fig. 2, we selected two videoclips for in-depth analysis in an attempt to further compare the values of linear blur kernel estimates and optical flow, and their effect on per-frame estimates of α.

As an example of the ideal case, we selected videoclip no. 16 from the BSD-16ms subset. The estimated value $\hat{\alpha} = 0.22$; ground truth $\alpha = 0.24$. In Fig. 3, we display frames F_{38}, F_{39}, the patch selected by the method, and detailed visualization of both linear blur kernel estimates and optical flow vectors. In this ideal case, we observe near-perfect collinearity of linear blur kernels and optical flow

vectors, as well as relatively uniform magnitudes of both vectors. We attribute the good performance of both linear blur kernel estimation and optical flow to the largely linear, lateral motion of the camera and the presence of areas with blurred textured surfaces in the scene. Similar situations with camera motion are ideal for the utilized linear blur kernel estimator, as Gong *et al.*'s method [2] was trained on synthetic data modeled as blur by camera motion.

We also analyze an example of a failure case in Fig. 4 where the estimate $\hat{\alpha}$ is grossly erroneous (ground truth $\alpha = 0.015$, estimate $\hat{\alpha} = 0.065$). Here, we observe an incorrectly estimated magnitude of linear blur kernels, resulting in an overestimate of α. The method of [2] seems to fail in dark areas with low contrast and no pronounced textures. As discussed in Sec. 4.3, the linear blur kernel estimator does not estimate low levels of motion blur accurately. In consequence, the method often fails to produce accurate estimates for ground truth values of $\alpha < 0.1$.
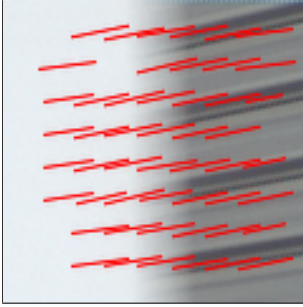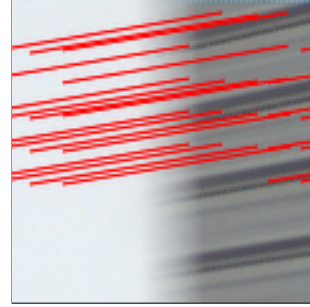
(a) Video frame F_{38}, selected patch highlighted in red.

(b) Video frame F_{39}

(c) Linear blur kernels inside and around the selected patch.

(d) Optical flow between F_{38}, F_{39} inside and around the selected patch.

Figure 3. Example patch with nearly perfect agreement with the assumption expressed by Eq. (4). The blur kernel and optical flow estimates are collinear, $\hat{\alpha}_{patch} = 0.26$ and $\alpha = 0.24$. The selected patch from frame F_{38}, video clip no. 16 from BSD-16ms subset. Estimation parameters $\varphi = 5°$, $D = 30$.

Clip #	k	α	$\hat{\alpha}$	α'	$\hat{\alpha}'$	Abs. error
78	3	0.360	0.349	0.120	0.117	0.003
14	3	0.240	0.230	0.080	0.075	0.005
115	2	0.120	0.121	0.060	0.054	0.006

Table 3. Detection of video clip subsampling by integer factor k. The ground truth α and estimated $\hat{\alpha}$ on the original sequence and the GT α' and the estimated $\hat{\alpha}'$ exposure fractions on the tampered video. In all cases, the value of $\hat{\alpha}'$ was estimated accurately. The test was performed on videoclips 78 (BSD-24ms), 14 (BSD-16ms), and 115 (BSD-8ms) containing traffic and moving vehicles. Estimation parameters $\varphi = 5°$, $D = 30$.

4.5. Detection of video alteration by frame deletion

Video frame deletion is a form of video clip tampering that directly affects the temporal consistency introduced by the camera physical properties and its exposure mechanism. When consecutive video frames are deleted, objects in the scene exhibit progressively larger inter-frame motions, pro-portionally to the number of removed frames, yet the motion blur remains the same. If the time scale, i.e. the playback frame rate, is edited or ignored by the player, the re-played video will appear to the viewer to have faster motions. As a result of frame deletion, the estimated value of α will not be consistent with the original video clip; the tampered section will have values of the α different from the rest of the video.

Frame deletion and insertion may be used for malicious purposes in video clips where the speed of motion provides significant information value, such as video clips from automotive dash cameras that could be used for speed measurements. Similarly, it may be utilized to remove frames containing sensitive or identifying information, such as license plates or faces on footage from surveillance cameras. In the case of dash and surveillance cameras, the capture framerate f and ε are often available as camera metadata, allowing a comparison between altered video clips and ground truth α.

We selected three video clips of scenes containing traffic and moving vehicles with $\alpha > 0.1$. This choice was based on the fact that the method performs better on larger

(a) Video frame F_{52}, selected patch highlighted in red.

(b) Video frame F_{53}

(c) Linear blur kernels inside and around the selected patch.

(d) Optical flow between F_{52}, F_{53} inside and around the selected patch.

Figure 4. A failure case of linear blur kernel estimates in a dark, low contrast area with no texture; $\hat{\alpha}_{patch} = 0.065$, $\alpha = 0.015$. The linear blur kernel estimator fails to accurately model the blur magnitudes, resulting in an inaccurate estimate of $\hat{\alpha}_{patch}$. Since the kernels still satisfy the orientation and magnitude constraints defined in Eq. (5), Eq. (6), Eq. (7), they are considered valid. Similar situations remain challenging for both the linear blur kernel estimator and the method. Frame F_{52}, estimation parameters $\varphi = 5°$, $D = 30$. Video clip no. 74 from BSD-1ms subset.

values of α due to the limitations of the linear blur estimator. For each of the selected video clips, we performed frame subsampling with an integer factor k, i.e. every k-th frame was preserved; the intermediate frames were discarded. The new apparent ground truth value of $\alpha' = \frac{\alpha}{k}$ where α is the ground truth value of the source video clip. The results are displayed in Tab. 3. For all subsampled videoclips, the method produced estimates within a close margin of the apparent ground truth value. Note significantly lower error than on video clips where unmodified $\alpha = \{0.015, 0.030, 0.045\}$. We attribute this to more accurate estimates in the selected videoclips, as their unmodified $\alpha = \{0.120, 0.240, 0.360\}$ results in larger motion blur and therefore more accurate linear blur kernel estimates.

4.6. Detection of video alteration by frame interpolation

Video frame interpolation is a technique of temporal alteration that synthesizes new intermediate frames in order to increase video framerate and for motion to appear smoother.

Clip #	Interpolation factor	α	$\hat{\alpha}$	$\hat{\alpha}'$
78	2x	0.360	0.349	0.668
	4x	–	–	0.536
14	2x	0.240	0.230	0.412
	4x	–	–	0.562

Table 4. Detection of video clip interpolation. The ground truth α and estimated $\hat{\alpha}$ on the original sequence and the estimated $\hat{\alpha}'$ exposure fractions on the tampered video. In all cases, the value of $\hat{\alpha}'$ increased noticeably. The test was performed on videoclips 78 (BSD-24ms) and 14 (BSD-16ms) containing traffic and moving vehicles. Estimation parameters $\varphi = 5°$, $D = 30$

It may be followed by a change of playback timescale, in which case it results in a slow-motion video. A videoclip altered in such a way can then be used as fake evidence of vehicle speed from a surveillance or dash camera. Techniques derived from video frame interpolation may also be

used to add new content to videos or change their appearance, such as interpolation between still photographs of a person's face for the creation of so-called "deepfakes" [3]. As described in Sec. 4.5, surveillance or dash cameras often save the value of ε and f by directly imprinting it on video frames or by saving it to the metadata. This provides the ground-truth value of α for comparison with the method estimate.

Based on the definition of exposure fraction α (Sec. 3.3), it is expected its value to increase if the newly-synthesized intermediate frames reduce the inter-frame motion of objects without affecting the amount of motion blur, i.e. the newly synthesized frames appear as blurry as the source frames. It is, however, not possible to compute the value of α' of interpolated videoclips accurately, as modern deep neural network-based interpolation methods such as RIFE [4] do not perform parametrized blurring or deblurring.

We performed interpolation on videoclips no. 78 and 14 from the experiment in Sec. 4.5 with the state-of-the-art RIFE method [4]. For each videoclip, we tested 2x interpolation and 4x interpolation. Results are presented in Tab. 4. We observe an increase in estimates of α' in all cases of interpolation, pointing to the method synthesizing new frames with similar amounts of motion blur as the source.

5. Further applications

In applications concerning video frame interpolation and video frame deblurring, the value of α might help parameterize blur for more accurate deblurring or motion modeling. The synthetization of new frames from a blurry source remains a challenge for modern interpolation methods, and accurate blur modeling might provide the necessary information for performance improvements [5]. In the case of linear blur estimates and optical flow, the estimate of α is useful as a complement in computing values for positions $(x,\,y)$ where one of the methods failed (under the assumption that is it possible to estimate the value of α from other frames and positions in the videoclip). This might be useful for the creation of new datasets for linear blur kernel estimation or optical flow, as the parameters may be estimated from existing datasets and computed for entire frames or videoclips.

6. Conclusion

We proposed a novel method for estimating the exposure fraction based on dense optical flow and linear blur estimates. The method was evaluated on the publicly available BSD Dataset. The mean absolute error was 0.039; the method performed best in the range $(0.12, 0.36)$. We observed reduced accuracy for ground truth values below 0.1, leading us to conjecture that the use of discrete linear blur kernel estimates may be a limiting factor. De-

veloping an improved method for the estimation of linear blur kernels is a key part of our future work. Lastly, we presented a possible application of exposure fraction estimation for video tampering detection, specifically of frame deletion and frame insertion. The implementation is available at https://github.com/edavidk7/exposure_fraction_estimation.

References

[1] Alastair Barber, Matthew Brown, Paul Hogbin, and Darren Cosker. Inferring changes in intrinsic parameters from motion blur. *Computers Graphics*, 52:155–170, 2015. 1

[2] Dong Gong, Jie Yang, Lingqiao Liu, Yanning Zhang, Ian Reid, Chunhua Shen, Anton van den Hengel, and Qinfeng Shi. From motion blur to motion flow: A deep learning solution for removing heterogeneous motion blur. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017. 1, 2, 3, 4, 5

[3] David Güera and Edward J. Delp. Deepfake video detection using recurrent neural networks. In *2018 15th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)*, pages 1–6, 2018. 8

[4] Zhewei Huang, Tianyuan Zhang, Wen Heng, Boxin Shi, and Shuchang Zhou. Real-time intermediate flow estimation for video frame interpolation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2022. 8

[5] Xiang Ji, Zhixiang Wang, Zhihang Zhong, and Yinqiang Zheng. Rethinking video frame interpolation from shutter mode induced degradation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 12259–12268, October 2023. 8

[6] RED Digital Cinema LLC. Shutter angles and creative control. 1

[7] Seungjun Nah, Tae Hyun Kim, and Kyoung Mu Lee. Deep multi-scale convolutional neural network for dynamic scene deblurring. In *CVPR*, July 2017. 4

[8] Gyula Simon, Gergely Vakulya, and Márk Rátosi. The way to modern shutter speed measurement methods: A historical overview. *Sensors*, 22(5), 2022. 1

[9] Shuochen Su, Mauricio Delbracio, Jue Wang, Guillermo Sapiro, Wolfgang Heidrich, and Oliver Wang. Deep video deblurring for hand-held cameras. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1279–1288, 2017. 4

[10] Jian Sun, Wenfei Cao, Zongben Xu, and Jean Ponce. Learning a convolutional neural network for non-uniform motion blur removal, 2015. 2

[11] Zachary Teed and Jia Deng. RAFT: recurrent all-pairs field transforms for optical flow. *CoRR*, abs/2003.12039, 2020. 1, 2, 4

[12] Mingliang Zhai, Xuezhi Xiang, Ning Lv, and Xiangdong Kong. Optical flow and scene flow estimation: A survey. *Pattern Recognition*, 114:107861, 2021. 2

[13] Youjian Zhang, Chaoyue Wang, Stephen J. Maybank, and Dacheng Tao. Exposure trajectory recovery from motion blur. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(11):7490–7504, 2022. 2

[14] Zhihang Zhong, Ye Gao, Yinqiang Zheng, and Bo Zheng. Efficient spatio-temporal recurrent neural network for video deblurring. In *European Conference on Computer Vision*, pages 191–207. Springer, 2020. 4