User-Assisted Reflection Detection and Feature Point Tracking

Mohamed A. Elgharib
Boston University
MA, USA
gharib@bu.edu

François Pitite
Trinity College Dublin
Dublin, Ireland
fpitie@mee.tcd.ie

Anil Kokaram
Google Inc.
CA, USA
akokaram@gmail.com

Venkatesh Saligrama
Boston University
MA, USA
srv@bu.edu

ABSTRACT

Reflections in image sequences violate the single layer model used by most current image processing techniques. As a result reflections cause many techniques to fail e.g. detection, tracking, motion estimation, etc. Recent work was proposed by Ahmed et al. [5] to detect reflections. Their technique is robust to pathological motion and motion blur. This paper has three main contributions. The first simplifies and fully automates the technique of Ahmed et al. User feedback is common in post-production video manipulation tools. Hence in the second contribution we propose an effective way of integrating few user-assisted masks to improve detection rates. The third contribution of this paper is an application for reflection detection. Here we explore better feature point tracking for the regions detected as reflection. Tracks usually die quickly in such regions due to temporal color inconsistencies. In this paper we show that the lifespan of such tracks can be extended through layer separation. Results show reduction in missed detections and in computational load over Ahmed et al. Results also show the generation of more reliable tracks despite strong layer mixing.

Keywords

Reflection Detection, Feature Point Tracks, Layer Separation, User-Assisted Masks

1. INTRODUCTION

Reflections are common in video and often arise due to photographing a (background) object situated behind a semi reflective (foreground) medium i.e. glass window. As a result the captured image is a mixture between the background and the foreground scenes (see Fig. 1). When viewed from a moving camera two different layers are observed each moving with a different motion. This phenomenon violates many of the existing models for image sequences as these models assume the presence of one motion per pel. The result is the failure of many important video tools e.g. feature point tracking, motion estimation, etc. Such failure can be solved by first detecting regions containing reflections and then assigning a special treatment to the detected regions.

Reflection detection is hard as reflections manifest in various forms (see Fig. 1). Although many work exists on reflection separation [14, 18, 8, 6, 16, 11], it is only recently that a technique for refection detection was proposed by Ahmed et al. [5, 7]. In their work reflections are detected using a combination of nine weak detectors. The weak detectors measure, along KLT tracks [17], the sharpness, temporal discontinuity and layer separability of the local image patches centered around the examined feature points. Those weak detectors are combined into one strong detector using Adaboost [10]. The detection map is spatially refined using the geodesic distance [13]. It is then made temporally consistent by propagating the best detection frame mask to the rest of the examined sequence. Ahmed et al. show high reflection detection rate with rejection to pathological motion such as motion blur and fast motion.

This paper has three contributions. The first simplifies and fully automates the spatio-temporal mask refinement step of [5]. Here euclidean distance is used instead of geodesic distance. Furthermore instead of propagating the detection mask of a user-suggested frame, a key frame is automatically selected. Results show reduction in computational load over [5] while maintaining the detection rates. The second contribution of this paper is a mechanism which allows the user to improve the detection results. It is well acknowledged in the post-production industry that fully automated techniques are not satisfactory if the user cannot intervene to recover from failures. We propose that the user can supply a small number of manually generated detection masks to the examined sequence. Those masks are supplied after the fully automated approach is run once. The masks either indicate missed sites to be detected or false detections to be removed. Manual intervention is justifiable as our motivation is driven by improving video processing tools. Such tools are usually present in user assisted post production software i.e. After Effects [1], NUKE [2].

The third contribution of this paper is an application for reflection detection. Here we propose an approach which improves feature point tracking for the detected sites. Feature point tracking is used to track objects through time. Such tracks die quickly in reflections due to temporal light illumination inconsistency. Previous work ex-

Figure 1: ***Reflection examples with the foreground layers shown in green. From left; a person (foreground) superimposed on a shop indoor, a building (foreground) superimposed on white posters, pedestrians (foreground) superimposed on a red background.***

ist for motion estimation for regions of reflections [4, 3, 15]. However no work attempted to improve feature point tracking for such regions. In this paper we propose that feature point tracks can be made temporally and spatially longer by processing the separated layers instead of the original sequence. Here we use the separation techniques of Ahmed et al. [5] and Weiss [18]. Results show more reliable tracks.

In the next section we present an overview of the reflection detector of Ahmed et al. [5]. Here we show how to fully automate and reduce the computational load of [5]. In section 3 we show how detection can be improved with user feed-back. Section 4 proposes our approach of improving feature point tracks for regions of reflections. Section 5 is results followed by discussion. Full image sequence results of the work presented in this paper are available on www.sigmedia.tv/misc/icip2011.

2. REFLECTION DETECTION

2.1 Analyzing Feature Point Trajectories For Reflection

We present in this section an overview of the detection technique of Ahmed et al. [5]. We first discuss the weak detectors $\mathcal{D}_{1-9}$ and how they form a strong detector $\mathcal{D}_s$. We then show spatio-temporal refinement of the detection masks and our improvements. The technique analyzes feature point trajectories. All analyzes are performed on a 50×50 patch $\mathcal{F}$ centered on the examined feature point .

Layer Separation via Color Independence $\mathcal{D}_1$: The examined patch $\mathcal{F}$ is separated by minimizing the number of observed corners [11] in its underlying layers ($\mathcal{L}_1$,$\mathcal{L}_2$) through $\mathcal{L}_1 = \mathcal{F}^R - \alpha\mathcal{F}^B$ and $\mathcal{L}_2 = \mathcal{F}^B - \beta\mathcal{F}^R$. Here ($\mathcal{F}^R$,$\mathcal{F}^B$) are the red and blue channels of $\mathcal{F}$ respectively. (α,β) are parameters calculated using an exhaustive search in the range of [0:0.1:1] (MATLAB notation). Reflection is then detected if the temporal similarity before separation is lower than the temporal similarity after separation by a value of $\mathcal{T}_1$.
Intrinsic Layer Extraction $\mathcal{D}_2$: The 50×50 intrinsic image (INT) of the examined trajectory is extracted using Weiss [18]. Reflection is detected if the similarity between INT and $\mathcal{F} < \mathcal{T}2$.
Color Channels Independence $\mathcal{D}_3$: Reflection in $\mathcal{F}$ is examined by measuring the level of correlation between the red and blue channels. Correlation is measured by GNGC [14] and reflection is detected if GNGC($\mathcal{F}^R$,$\mathcal{F}^B$)$< \mathcal{T}_3$.
Image Sharpness ($\mathcal{D}_4$,$\mathcal{D}_5$): Reflections have low image sharpness as they are mixtures of multiple images. We asses image sharpness using two detectors. The first D_4 flags $\mathcal{F}$ as reflection if the mean of its gradient magnitude is $< \mathcal{T}_4$. The second detector D_5 uses Frezil et. al approach [9] to calculate image sharpness and flags $\mathcal{F}$ as reflection if the calculated value is less than $\mathcal{T}_5$.
SIFT Temporal Discontinuity $\mathcal{D}_6$: SIFT features [12] are extracted for each 50×50 patch along the examined trajectory. Lowe matching technique [12] is used to calculate the cost of matching the SIFT features in the examined patch with the features in the previous and next frames. The examined patch is flagged as containing reflection if the minimum of these costs exceeds a threshold value of $\mathcal{T}_6$.
Color Temporal Profile $\mathcal{D}_7$: $\mathcal{F}$ is flagged as reflection if the temporal change of its mean grayscale value $> \mathcal{T}_7$.
AutoCorrelation Temporal Profile $\mathcal{D}_8$: $\mathcal{F}$ is flagged as reflection if the temporal change in the autocorrelation $> \mathcal{T}_8$.
Motion Field Divergence $\mathcal{D}_9$: $\mathcal{F}$ is flagged as reflection if the motion field divergence $> \mathcal{T}_9$.

A Strong Detector with Adaboost: $\mathcal{D}_{2-8}$ are combined together to form one strong detector $\mathcal{D}_s$ using Adaboost [5]. Reflection is then detected as the output of **AND**$(\mathcal{D}_s > \mathcal{T}_s, \mathcal{D}_1 > \mathcal{T}_1, \mathcal{D}_9 > \mathcal{T}_9)$. Each detected point flags the 50×50 region centered on it as reflection.

2.2 Spatio Temporal Refinement

The previous solution generates very sparse detections. In this section we exploit spatio-temporal information to generate dense detection that is spatio-temporally consistent. In [5] spatio-temporal information are exploited in a semi-automated and computational intensive approach. **Here we propose an approach to fully automate this step and to reduce computational complexity**.

Thresholding with hysteresis: This stage rejects false detections that are spatially inconsistent. 1) Let $(\mathcal{T}_1, \mathcal{T}_s, \mathcal{T}_9) = (-0.22, 4.5, 10)$ 2) Estimate the detection using **AND**$(\mathcal{D}_s > \mathcal{T}_s, \mathcal{D}_1 > \mathcal{T}_1, \mathcal{D}_9 > \mathcal{T}_9)$ 3) Remove all detection points that have total euclidean distance of > 200 with the two closest detection points. In [5] the geodesic distance was used instead of the euclidean distance. To reduce the computational cost of the geodesic distance all frames were resized by a factor of 50 4) For each track, treat

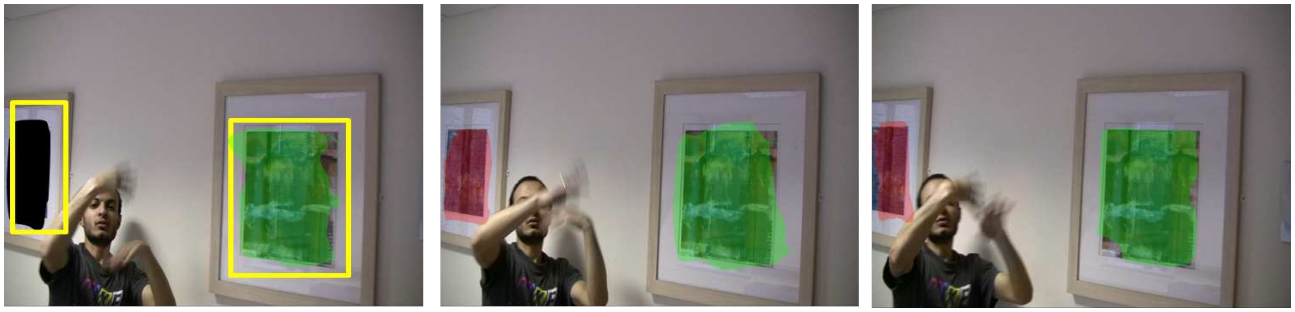

Figure 2: ***Reflection detection on real sequence using automated approach (shown in green) and user-assisted missed reflection recovery (shown in red). Here ground-truth is shown in yellow. Just one user-supplied mask (shown in black) every 10 frames was able to recover the missed detections in the rest of the sequence (shown in red). More results are available on www.sigmedia.tv/misc/icip2011.***

all of it as reflection detection if one of its feature points is classified as reflection. Each detected point flags the 50×50 region centered on it as reflection. We call this *spatio-temporal dilation* and the result is much denser detection masks 5) Group detection points and assign a new point to a group if it is within 200 pels from that group. Update group centroid 6) Lower thresholds and repeat the above steps till $(\mathcal{T}_1, \mathcal{T}_s, \mathcal{T}_9) = (0, 3.15, 10)$. The highest and lowest thresholds of $(\mathcal{T}_1, \mathcal{T}_s, \mathcal{T}_9)$ are fixed to $(-0.22, 4.5, 10)$ and $(0, 3.15, 10)$ respectively in all experiments. All parameter values here were found empirically after examining 2 sequences containing 100 frames. Those values were used to process 15 sequences containing 932 frames of size 576×720 (see Ahmed et al. [5] for more detail).

Imposing Temporal Consistency: The previous stage generates temporally inconsistent dense detection. Detection is made temporally consistent by temporally propagating a detection mask to the remaining of the examined sequence. This mask is referred to as the **keymask** in this paper and is manually selected in the previous work of [5]. In this paper **keymask** is set using an automated approach. Here **keymask** is set to be the frame that has the largest spatially connected detected reflection in the examined sequence. This assumption is supported by the previous step (previous paragraph) which rejects spatially inconsistent detections and hence avoids the generation of large spatially connected false detections. Temporal propagation is then applied. To fully exploit the **keymask** temporal propagation here is modified from [5]. Here the propagation starts by warping the **keymask** of the Key frame K on the previous frame $K-1$. The **keymask** motion is set to the dominant motion between K and $K-1$. This motion is modeled as a 2D affine transformation. The transformation parameters are calculated using least square fitting on the KLT tracks [17] at K and $K-1$. The propagated mask at frame $K-1$ is then propagated to frame $K-2$ using the same approach. This process is performed iteratively in both backward and forward directions till the start and end of the examined sequence.

Fig. 2 shows detection results (shown in green) of processing a 100 frame sequence. The spatial resolution of this sequence is 576×720 pels. The right picture on the wall (shown in green) is correctly detected as containing reflection. However the reflection on the left picture (shown in red) is ignored. This is mainly because it contains weak feature points. In addition it is distant from the right picture and hence got rejected in the previously discussed hysteresis step. The Correct Detection/False Detection Rate for the automated detection approach with no geodesic distance is 0.83/0.01 for the whole examined sequence. This is comparable with [5] detection rate of 0.847/0.02. The removal of the geodesic distance had a slight effect on detection rate. However it led to significant reduction in computational time. The new fully automated approach is 46% faster than [5]. Reported time for [5] is 581 seconds while 317 seconds for our approach. The reported time is the average frame processing time of the sequence. Here a 4.53 GHz Quad Core Processor is used and coding is done with MATLAB.

3. USER ASSISTANCE FOR ROBUST DETECTION

Missed detections could occur if the reflections contain weak feature points (see Fig. 2, red). After running the fully automated detection approach once (Sec. 2.2), missed reflections are identified by asking the user to indicate some of the missed regions in form of rough hand-drawn masks. Those masks are referred to as the **user-masks** in this paper. Those **user-masks** do not need to be supplied at every frame. Instead, when manual intervention is necessary, we found **user-masks** should be supplied in average every tenth frame. This was taken as an average of processing four sequences of a total of 250 frames. Each **user-mask** should encompass as much missed detections as possible and should contain temporally consistent (long) feature point trajectories. Such tracks are easily identified visually by the user. All feature points encompassed in the **user-masks** are flagged as reflection. Detections are then extended in time to other feature points along their trajectories. Basically a trajectory is fully flagged as reflection if at least one of its points lie on the **user-masks**. This propagates the **user-masks** to the rest of the examined sequence. Finally the propagated masks are added to the fully automatically generated detection.

False detections are rejected using an approach similar to missed detection recovery. First the fully automated detection technique is run once. Then **user-masks** are manually provided by indicating sites of false detections. When manual intervention is necessary, we found **user-masks** should be supplied in average every tenth frame. Those masks are temporally propagated to the rest of the examined sequence as in false detection recovery. Finally the propagated masks are subtracted from the fully automatically generated detections.

Fig. 2 shows the ability of recovering missed detections using the proposed approach (shown in red). One of the nine supplied masks

for this 100 frame sequence is shown in black (see first column). The Correct Detection/False Detection rate is now 0.98/0.03. This compares favorably with 0.847/0.02 for [5] and 0.83/0.01 for the automated approach without manual feed-back. The **user-masks** spatio-temporal dilation took around 3 seconds for the examined sequence. Drawing the user-assisted mask takes an average of 32 seconds. Hence overall our approach with manual feed-back is still faster than [5] by around 44%. That is the average processing frame time over this examined 100 frames sequence.

4. FEATURE POINT TRACKING

Feature point tracks in regions of reflections are usually short due to temporal illumination inconsistency (see Fig. 3, bottom row, left). Such tracks can be improved by performing layer separation before track estimation. Different layer separation techniques impose different constraints on the underlying layers [18, 14, 5]. Here we propose two feature point tracking approaches for reflections, each using a different layer separation technique.

1) **Layer Separation using Color Independence:** The observed sequence is decomposed into its underlying layers by using the separation technique proposed in [5]. This technique does not impose any constraints on the layers' motions as it operates on still images. Fig. 3 (top) shows an examined reflection and the extracted layers using this approach. Here the background (shown in green) is well separated from the foreground (shown in yellow). Fig. 3, the second row, shows the extracted tracks for the examined frame. Processing the reflection directly with no layer separation often generates short tracks (see second row, first column). Here the tracks of the foreground and background layers are overlayed over each other. Nevertheless through layer separation we were able to generate much longer tracks (see black and blue rectangles). Some of these tracks undergo strong temporal color inconsistency changing its color from red to orange to green to orange (see blue rectangle). Our technique separates the background tracks from the foreground tracks successfully. All tracks here are calculated using the technique of Kanade [17].

2) **Layer Separation using Intrinsic Images:** A sequence of M frames is decomposed into M backgrounds and M foregrounds using Weiss technique [18]. In order to extract the background layer for the examined frame, [18] requires the background to be stationary for some duration. This is done by temporally aligning the backgrounds in the previous four frames and in the next four frames with the background in the examined frame. The background motion is assumed to be the frame dominant motion. This motion is modeled as a 2D affine transformation. The parameters of this transformation are calculated using least square fitting on the KLT tracks between the examined frame and the neighboring frames. The background is then extracted by directly applying Weiss [18] on the temporally aligned frames. This follows by foreground layer estimation as the residual between the observed sequence and estimated background. Fig. 4 shows the layer extraction and corresponding tracks using this approach. Similarly to the previously discussed technique, separated layers tracks (see second row) are much longer than the original tracks and well separated from each other.

	Sequence 5	Sequence 6	Sequence 7
Image mixture	24.17 ± 26.45 234.9	13.3 ± 13.7 108.9	13.3 ± 15.3 133.7
Foreground layer	30.31 ± 37.01 342	23.3 ± 25.2 203.2	25.08 ± 34.39 456.2
Background layer	6.5 ± 7.35 75.5	9.54 ± 3.07 36.25	9.03 ± 6.02 65.2

Table 2: ***Feature point tracking results. For each sequence we show the mean spatial track length ($\pm$ standard deviation) and the length (in pels) of the longest track. Please refer to Fig. 7(a) for more detail. Note how foreground tracks are longer (spatially) than background and image mixture tracks. Furthermore the background tracks are the shortest as they correspond to the almost stationary backgrounds.***

	Sequence 5	Sequence 6	Sequence 7
Image mixture	21.82 ± 28.23 119	46.1 ± 19.75 59	34 ± 27.3 74
Foreground layer	17.48 ± 23.83 119	6.27 ± 7.4 59	2.8 ± 4.1 63
Background layer	28.9 ± 23.28 120	54.8 ± 12 59	42 ± 29.2 74

Table 3: ***Feature point tracking results. For each sequence we show the mean temporal track length ($\pm$ standard deviation) and the length (in frames) of the longest track. Please refer to Fig. 7(b) for more detail. Note how background tracks are longer (temporally) than foreground and image mixture tracks. Furthermore the foreground tracks here are the shortest in term of temporal extend as they are moving the fastest.***

5. RESULTS

5.1 Reflection Detection

Four image sequences of 250 frames with a spatial resolution of 576×720 pels are examined. Fig. 5 shows some detection results from the examined sequences. Here ground-truth are shown in yellow and automated reflection detection is shown in green. Note that our detector correctly classifies the pathological motion of the actor's hands as non-reflection. In the first two rows our detector does not detect the region shown in black (first column) as it contains weak feature points. To detect this region the user supplies a detection mask (shown in black) every 10 frames. The result is successful recovery of such missed sites in the rest of the examined sequences (see red regions, first two rows). Furthermore, in the last two rows the detector generates some false alarms (see black region, first column). To reject such detections the user supplied few masks (shown in black) every 10 frames. Such masks were able to reject the false alarms in the rest of the examined sequences (see red regions, last two rows).

Table 1 summarizes the detection results and processing times of the examined techniques. Here a 4.53 GHz Quad Core Processor is used and coding is done with MATLAB. The fully automated detector that we proposed in this paper is around 47% faster than the detector of Ahmed et al. [5]. Here the detection rate of Ahmed et al. is maintained. Table 1 also shows that user feedback was able to boost correct detection in Sequence 1 and 2 and to reduce false detection in Sequence 3 and 4. Here the user-assisted approach is 45% faster than the detector of Ahmed et al.

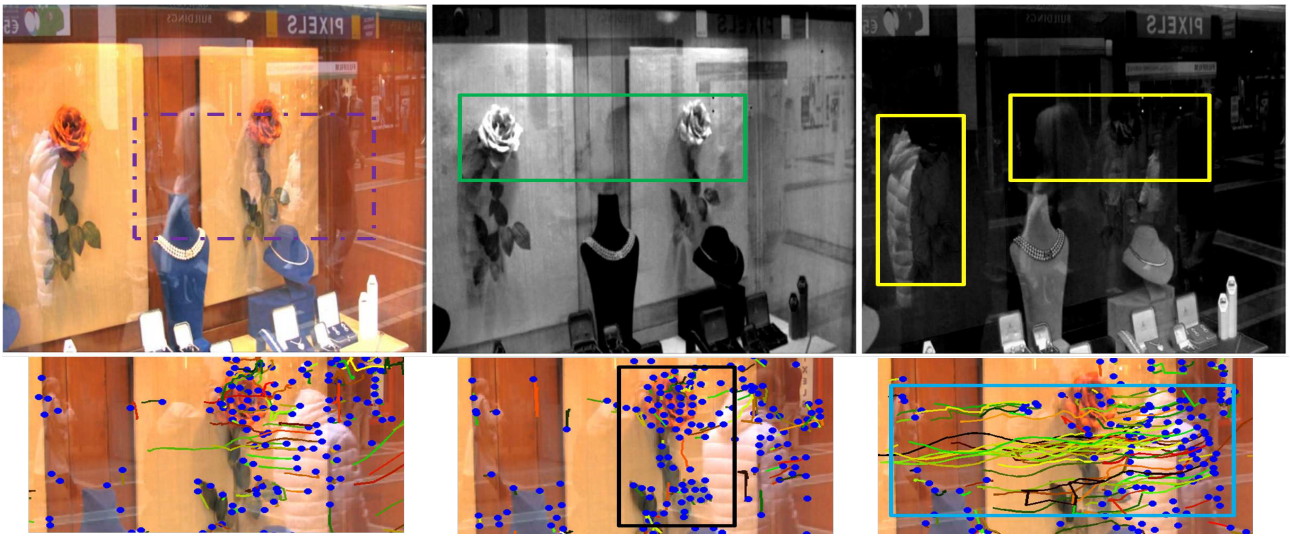

Figure 3: ***First row, from left: A frame containing reflection, background and foreground layer extraction using [5]. Here the foreground objects (see yellow) are well separated from the background layer (see green). Note that here we do not care about perfect layer separation. Instead we only care about separation that is good enough for generating reliable tracks. Second row: dashed purple area zoomed on. From left: KLT tracks of the observed reflection and of the separated background and foreground layers respectively. Tracks start with blue circle shaped heads and their tails have different colors. The black and blue rectangles show that tracks of the separated layers are longer than the tracks of the original frame (first column). Full image sequence results are on www.sigmedia.tv/misc/icip2011.***

	Sequence 1	Sequence 2	Sequence 3	Sequence 4
Ahmed et al. [5]	0.847/0.02 581	0.853/0.03 562	0.97/0.02 574	0.98/0.07 602
Fully Automated	0.83/0.01 317	0.843/0.01 288	0.96/0.03 301	0.96/0/08 319
User-Assistance	0.98/0.03 327	0.97/0.03 298	0.97/0.01 314	0.98/0.02 334

Table 1: ***Correct Detection/False Detection rate and computational time (in seconds) for the examined techniques. User-assistance improves correct detection considerably in Sequence 1 and 2. In addition it reduces False detections in Sequence 3 and 4. Incorporating the euclidean distance (see Fully Automated) instead of the geodesic distance of Ahmed et al. reduces computational time by around*** 47%***. The reported times here are the average frame processing time over the examined sequences.***

5.2 Feature Point Tracking

Three image sequences of 250 frames with a spatial resolution of 576×720 pels are examined. Fig. 6 shows the generated tracks for the observed image sequences and the separated layers. The sequence in the second row is processed using Ahmed et al. [5] layer separation technique. The remaining sequences are processed with Weiss separation approach [18]. In all cases the original image sequences generated short tracks. However the feature point tracker proposed in this paper was able to generate longer tracks for both foreground and background layers. In addition tracks of different layers are separated despite strong layer mixing (see third row).

Table 2 shows some results generated by the proposed feature point trackers. For each sequence we show the mean spatial track length. That is the total number of pels traveled by all tracks divided by the number of tracks of the examined sequence. For each sequence we also show the length (in pels) of the spatially longest track. Table 2 is calculated using Fig. 7(a), the histograms of the spatial track lengths. Table 2 shows that the mean spatial track length of the foreground layer is usually larger than the mean spatial track length of the observed image mixtures. This confirms that our proposed trackers do generate tracks longer than the tracks of the image mixtures. The mean spatial track length of the background layers however are shorter than the mean spatial track length of the observed image mixtures. This is mainly because the backgrounds of the examined sequences are moving very slowly, close to being stationary.

Table 3 shows more results generated by the proposed feature point trackers. For each sequence we show the mean temporal track length. That is the total number of frames traveled by all tracks divided by the number of tracks of the examined sequence. For each sequence we also show the length (in frames) of the temporally longest track. Table 3 is calculated using Fig. 7(b), the histograms of the temporal track lengths. Table 3 shows that the mean temporal track length of the background layer is usually larger than the mean temporal track length of the observed image mixtures. This confirms that our proposed trackers do increase the temporal extend of tracks. The mean temporal track length of the foreground layers however are shorter than the mean temporal track length of the observed image mixtures. This is mainly because the foreground is moving fast and hence its feature points enter and exit the scene in a short period of time.

Figure 4: ***First row, from left: A frame containing reflection, background and foreground layer extraction using [18]. Here the foreground objects (see yellow) are well separated from the background layer (see green). Note that here we do not care about perfect layer separation. Instead we only care about separation that is good enough for generating reliable tracks. Second row: dashed purple area zoomed on. From left: KLT tracks of the observed reflection and of the separated background and foreground layers respectively. Tracks start with blue circle shaped heads and their tails have different colors. The separated layers tracks are longer than the tracks of the original frame (first column). Full image sequence results are on www.sigmedia.tv/misc/icip2011.***

6. CONCLUSION

This paper proposed an approach for reflection detection. It builds on and improves the previous work of [5]. One aspect of novelty here is in exploiting spatio-temporal information in a more automated and computational efficient approach. We then showed that slight manual intervention can be exploited for better detection. Such manual intervention is common in the post-production video manipulation tools. Our results show higher detection rate, lower false detections and reduction in computational time by around 45% over Ahmed et al. We also proposed a system for feature point tracking on regions of reflections. The novelty here is in generating feature point tracks for the separated layers. Results show the generation of longer tracks over than directly processing the observed image mixtures. In addition results show that the generated foreground and background tracks are separated from each other.

7. ACKNOWLEDGMENTS

This work has been funded by Science Foundation Ireland (grant no. 08/IN.1/I2112) and Department of Homeland Security (DHS-ALERT-Video 9500240513).

8. REFERENCES

[1] Adobe Inc., After Effects. www.adobe.com.

[2] The Foundry, Nuke, Furnace Suite. www.thefoundry.co.uk.

[3] T. Aach, C. Mota, I. Stuke, M. Muhlich, and E. Barth. Analysis of Superimposed Oriented Patterns. *TIPS*, 15(12):3690–3700, 2006.

[4] M. Ahmed, F. Pitie, and A. Kokaram. Motion Estimation for Regions of Reflections through Layer Separation. In *IEEE Conference on Visual Media Production (CVMP)*, pages 49–58, 2011.

[5] M. Ahmed, F. Pitie, and A. Kokaram. Reflection Detection in Image Sequences. In *Computer Vision and Pattern Recognition (CVPR)*, pages 705–712, 2011.

[6] A. M. Bronstein, M. M. Bronstein, M. Zibulevsky, and Y. Y. Zeevi. Sparse ICA for blind separation of transmitted and reflected images. *International Journal of Imaging Systems and Technology*, 15(1):84–91, 2005.

[7] M. A. Elgharib. Handling Transparency in Digital Video. PhD Thesis, Trinity College Dublin.

[8] H. Farid and E. Adelson. Separating reflections from images by use of Independent Components Analysis. *Journal of the Optical Society of America*, 16(9):2136–2145, 1999.

[9] R. Ferzli and L. J. Karam. A no-reference objective image sharpness metric based on the notion of just noticeable blur (JNB). *Transactions on Image Processing (TIPS)*, 18(4):717–728, 2009.

[10] Y. Freund and R. E. Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. In *European Conference on Computational Learning Theory*, pages 23–37, 1995.

[11] A. Levin, A. Zomet, and Y. Weiss. Separating reflections from a single image using local features. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 306–313, 2004.

[12] D. G. Lowe. Distinctive Image Features from Scale-Invariant Keypoints. *International Journal of Computer Vision (IJCV)*, 60(2):91–110, 2004.

[13] D. Ring and F. Pitie. Feature-Assisted Sparse to Dense Motion Estimation using Geodesic Distances. In *International Machine Vision and Image Processing Conference*, pages 7–12, 2009.

[14] B. Sarel. Separation Transparent Layers in Images and Video. PhD Thesis, Weizmann Institute of Science.

[15] I. Stuke, T. Aach, E. Barth, and C. Mota. Multiple-Motion-Estimation by Block-matching using MRF. *International Journal of Computer and Information Science*, 5(1), 2004.

[16] R. Szeliski, S. Avidan, and P. Anandan. Layer extraction from multiple images containing reflections and transparency. In *Computer Vision and Pattern Recognition (CVPR)*, pages 246–253, 2000.

[17] C. T. Takeo and T. Kanade. Detection and tracking of point features. Carnegie Mellon University Technical Report CMU-CS-91-132, 1991.

[18] Y. Weiss. Deriving intrinsic images from image sequences. In *International Conference on Computer Vision (ICCV)*, pages 68–75, 2001.

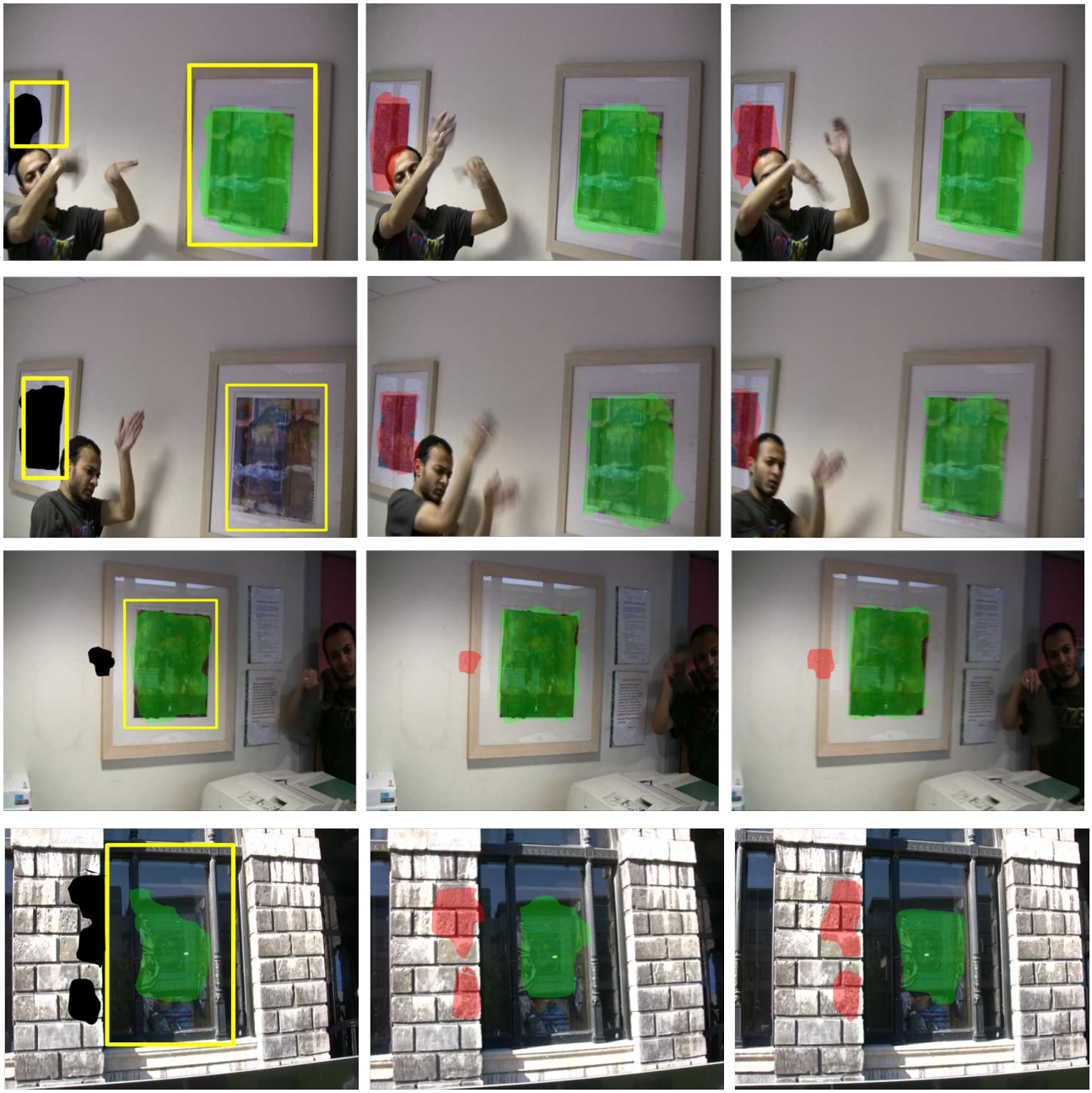

Figure 5: ***Reflection detection on real sequence using automated approach (shown in green) and user assisted detection (shown in red). Here ground-truth is shown in yellow. Just one user-supplied mask every 10 frames (shown in black) was able to recover missed detections (shown in red, first two rows) and to reject false detections (shown in red, last two rows). Full image sequence results are available on www.sigmedia.tv/misc/icip2011.***

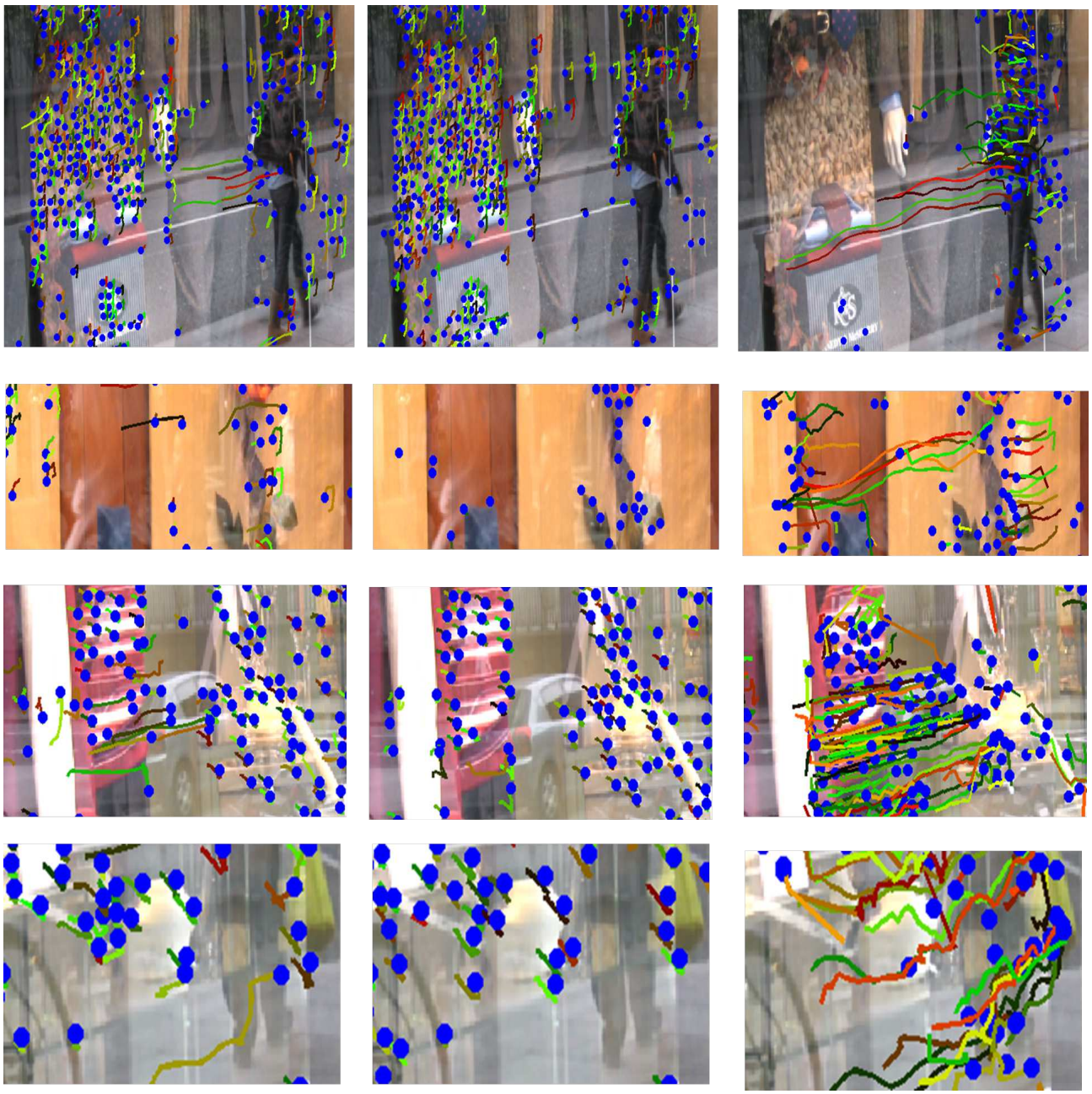

Figure 6: ***Feature point tracking on real image sequences. From left; Tracks of the observed reflection and of the separated background and foreground layers respectively. Tracks start with blue circle shaped heads and their tails have different colors. The underlying layers tracks (last two columns) are longer than the tracks of the original sequence (first column). In addition they are better separated from each other over than the original sequence. Full image sequence results are on www.sigmedia.tv/misc/icip2011.***

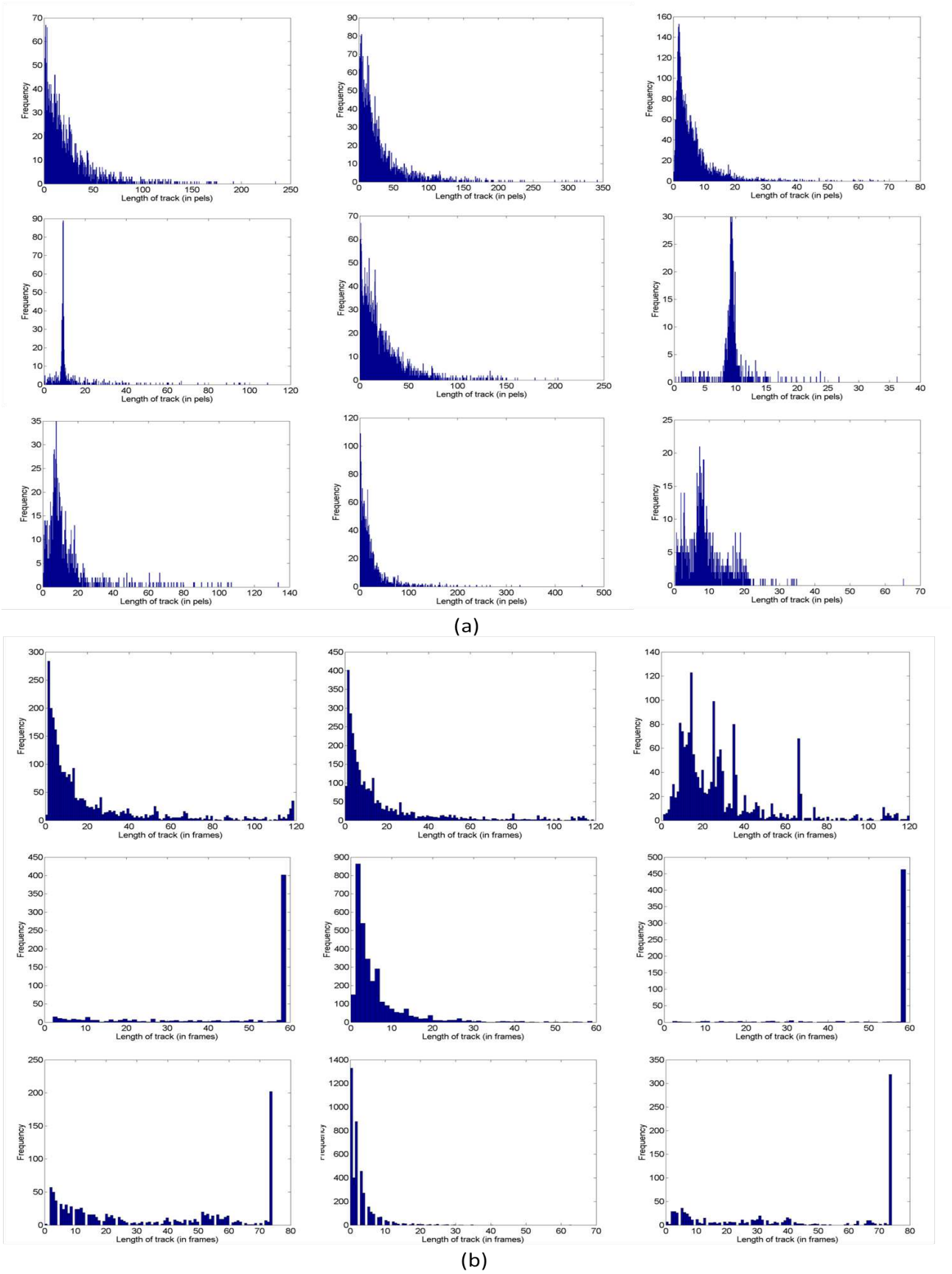

Figure 7: ***Histograms of the spatial track length (a) and the temporal track length (b) for (from top, for both a & b); Sequence 5, Sequence 6 and Sequence 7 respectively. For each sequence we show the histogram for (from left); observed image mixture, foreground layer and background layer respectively. Note that our tracker was able to increase the spatial length of the foreground tracks (see a). In addition it was able to increase the temporal extend of the background tracks (see b). The background tracks in (a) correspond to the almost stationary backgrounds. While the foreground tracks in (b) are the shortest in term of temporal extend as they are moving the fastest.***