Reflection Detection in Image Sequences

Mohamed Abdelaziz Ahmed Francois Pitie Anil Kokaram

Sigmedia, Electronic and Electrical Engineering Department, Trinity College Dublin {www.sigmedia.tv/People}

Abstract

Reflections in image sequences consist of several layers superimposed over each other. This phenomenon causes many image processing techniques to fail as they assume the presence of only one layer at each examined site e.g. motion estimation and object recognition. This work presents an automated technique for detecting reflections in image sequences by analyzing motion trajectories of feature points. It models reflection as regions containing two different layers moving over each other. We present a strong detector based on combining a set of weak detectors. We use novel priors, generate sparse and dense detection maps and our results show high detection rate with rejection to pathological motion and occlusion.

1. Introduction

Reflections are often the result of superimposing different layers over each other (see Fig. 1, 2, 4, 5). They mainly occur due to photographing objects situated behind a semi reflective medium (e.g. a glass window). As a result the captured image is a mixture between the reflecting surface (background layer) and the reflected image (foreground). When viewed from a moving camera, two different layers moving over each other in different directions are observed. This phenomenon violates many of the existing models for video sequences and hence causes many consumer video applications to fail e.g. slow-motion effects, motion based sports summarization and so on. This calls for the need of an automated technique that detects reflections and assigns a different treatment to them.

Detecting reflections requires analyzing data for specific reflection characteristics. However, as reflections can arise by mixing any two images, they come in many shapes and colors (Fig. 1, 2, 4, 5). This makes extracting characteristics specific to reflections not an easy task. Furthermore, one should be careful when using motion information of reflections as there is a high probability of motion estimation failure. For these reasons the problem of reflection detection is hard and was not examined before.

Reflection can be detected by examining the possibility of decomposing an image into two different layers. Several work exist on separating mixtures of semi-transparent layers [17, 11, 12, 7, 4, 1, 13, 3, 2]. Nevertheless, most of the still image techniques [11, 4, 1, 3, 2] require two mixtures of the same layers under two different mixing conditions while video techniques [17, 12, 13] assume a simple rigid motion for the background [17, 13] or a repetitive one [12]. These assumptions are hardly valid for reflections on moving image sequences.

This paper presents an automated technique for detecting reflections in image sequences. It is based on analyzing spatio-temporal profiles of feature point trajectories. This work focuses on examining three main features of reflections: 1) the ability of decomposing an image into two independent layers 2) image sharpness 3) the temporal behavior of image patches. Several weak detectors based on analyzing these features through different measures are proposed. A final strong detector is generated by combining the weak detectors. The problem is formulated within a Bayesian framework and priors are defined in a way to reject false alarms. Several sequences are processed and results show high detection rate with rejection to complicated motion patterns e.g. blur, occlusion, fast motion.

Aspects of novelty in this paper include: 1) A technique for decomposing a color still image containing reflection into two images containing the structures of the source layers. We do not claim that this technique could be used to fully remove reflections from videos. What we claim is that the extracted layers can be useful for reflection detection since on a block basis, reflection is reduced. This technique can not compete with state of the art separation techniques. However we use this technique because it works on single frames and thus does not require motion, which is not the case with most of the existing separation techniques. 2) Diagnostic tools for reflection detection based on analyzing feature points trajectories 3) A scheme for combining weak detectors in one strong reflection detector using Adaboost 4) Incorporating priors which reject spatially and temporally impulsive detections 5) The generation of dense detection maps from sparse detections and using thresholding by hys-

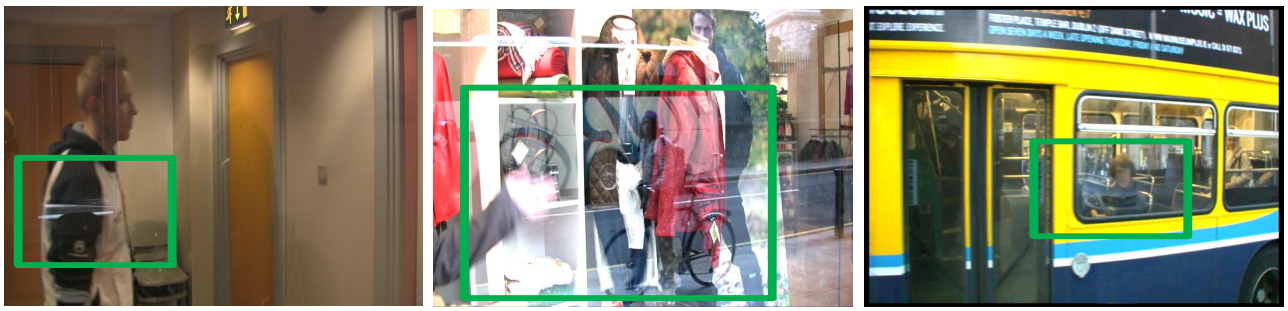

Figure 1. *Examples of different reflections (shown in green). Reflection is the result of superimposing different layers over each other. As a result they have a wide range of colors and shapes.*

teresis to avoid selecting particular thresholds for the system parameters 6) Using the generated maps to perform better frame rate conversion in regions of reflection. Frame rate conversion is a computer vision application that is widely used in the post-production industry. In the next section we present a review on the relevant techniques for layer separation. In section 3 we propose our layer separation technique. We then go to propose our Bayesian framework followed by the results section.

2. Review on Layer Separation Techniques

A mixed image M is modeled as a linear combination between the source layers L_1 and L_2 according to the mixing parameters (a, b) as follows.

$$M = aL_1 + bL_2 \tag{1}$$

Layer separation techniques attempt to decompose reflection M into two independent layers. They often do so by exchanging information between the source layers (L_1 and L_2) until their mutual independence is maximized. In many techniques this requires the presence of two mixtures of the same layers under two different mixing proportions [11, 4, 1, 3, 2]. Different separation techniques use different forms of expressing the mutual layer independence. Current forms used include minimizing the number of corners in the separated layers [7] and minimizing the grayscale correlation between the layers [11].

Other techniques [17, 12, 13] avoid the requirement of having two mixtures of the same layers by using temporal information. However they often require either a static background throughout the whole image sequence [17], constraint both layers to be of non-varying content through time [13], or require the presence of repetitive dynamic motion in one of the layers [12]. Yair Weiss [17] developed a technique which estimates the intrinsic image (static background) of an image sequence. Gradients of the intrinsic layer are calculated by temporally filtering the gradient field of the sequence. Filtering is performed in horizontal and vertical directions and the generated gradients are used to reconstruct the rest of the background image.

3. Layer Separation Using Color Independence

The source layers of a reflection M are usually color independent. We noticed that the red and blue channels of M are the two most uncorrelated RGB channels. Each of these channels is usually dominated by one layer. Hence the source layers (L_1, L_2) can be estimated by exchanging information between the red and blue channels till the mutual independence between both channels is maximized. Information exchange for layer separation was first introduced by Sarel et. al [12] and it is reformulated for our problem as follows

$$\begin{aligned} L_1 &= M^R - \alpha M^B \\ L_2 &= M^B - \beta M^R \end{aligned} \tag{2}$$

Here (M_R, M_B) are the red and blue channels of the mixture M while (α, β) are separation parameters to be calculated. An exhaustive search for (α, β) is performed. Motivated by Levin et. al. work on layer separation [7], the best separated layer is selected as the one with the lowest cornerness value. The Harris cornerness operator is used here. A minimum texture is imposed on the separated layers by discarding layers with a variance less than T_x. For an 8-bit image, T_x is set to 2. The removal of this constraint can generate empty meaningless layers. The novelty in this layer separation technique is that unlike most previous techniques [11, 4, 1, 3, 2, 17, 12, 13], it only requires one image.

Fig.2 shows separation results generated by the proposed technique for different images. Results show that our technique reduces reflections and shadows. Results are only displayed to illustrate a preprocess step that is used for one of our reflection measures and not to illustrate full reflection removal. Blocky artifacts are due to processing images in 50×50 blocks. These artifacts are irrelevant to reflection detection.

4. Bayesian Inference for Reflection Detection (BIRD)

The goal of the algorithm is to find regions in image sequences containing reflections. This is achieved by examin-

Figure 2. *Reducing reflections/shadows using the proposed layer separation technique. Frames are processed in* 50×50 *blocks. Color images are the original images with reflections/shadows (shown in green). The uncolored images represent one source layer (calculated by our technique) with reflections/shadows reduced. Car Window example in Last Row, Middle Column; reflection still remains apparent however the person in the car is fully removed.*

ing trajectories of feature points. Trajectories are generated using KLT feature point tracker [9, 14]. Denote $\mathcal{P}_n^i$ as the feature point of i^{th} track in frame n and $\mathcal{F}_n^i$ as the 50×50 image patch centered on $\mathcal{P}_n^i$. For each $\mathcal{P}_n^i$, analyses are carried over the three image patches $(\mathcal{F}_{n-1}^i, \mathcal{F}_n^i, \mathcal{F}_{n+1}^i)$. Based on the outcome, a binary label field l_n^i is assigned to each $\mathcal{F}_n^i$. l_n^i is set to 1 for reflection and 0 otherwise.

4.1. Bayesian Framework

The system derives an estimate for l_n^i from the posterior $P(l|\mathcal{F})$ (where (i,n) are dropped for clarity). The posterior is factorized in a Bayesian fashion as follows

$$P(l|\mathcal{F}) = P(\mathcal{F}|l)P(l|l^{\mathcal{N}}) \tag{3}$$

The likelihood term $P(\mathcal{F}|l)$ consists of 9 detectors $\mathcal{D}_1 - \mathcal{D}_9$ each performing different analyses on $\mathcal{F}$ and operating at thresholds $\mathcal{T}_{1-9}$ (see Sec. 4.3.1). The prior $P(l|l^{\mathcal{N}})$ enforces various smoothness constraints in space and time to reject spatially and temporally impulsive detections and to generate dense detection masks. Here $\mathcal{N}$ denote the spatio-temporal neighborhood of the examined site.

4.2. Feature Point Analyses for Reflection Detection

Reflections can not be described by a physical model as they come in a large variety of shapes and colors. However, they can be described by a number of physical characteristics. We propose that one can detect reflections by examining three main characteristics. The first characteristic is that reflections can be decomposed into two independent layers and the second is that they have low image sharpness. The final characteristic is that regions of reflections often undergo large temporal discontinuities. However, to avoid classifying complicated motion patterns such as pathological motion as being reflection, all analysis are performed on feature point trajectories of length more than 4 frames.

Layer Separation via Color Independence $\mathcal{D}_1$: Our technique (presented in Sec.3) is used to decompose the image patch $\mathcal{F}_n^i$ into two layers $\mathcal{L}1_n^i$ and $\mathcal{L}2_n^i$. This is applied

for every point along every track. Patches containing reflection are defined as ones with higher temporal discontinuity before separation than after separation. Temporal discontinuity is measured using structure similarity index SSIM [16] as follows

$$\begin{aligned}\mathcal{D}1_n^i &= \max(\text{SS}(\mathcal{G}_n^i, \mathcal{G}_{n-1}^i), \text{SS}(\mathcal{G}_n^i, \mathcal{G}_{n+1}^i)) \\ &\quad - \max(\text{SS}(\mathcal{L}_n^i, \mathcal{L}_{n-1}^i), \text{SS}(\mathcal{L}_n^i, \mathcal{L}_{n+1}^i)) \\ \text{SS}(\mathcal{L}_n^i, \mathcal{L}_{n-1}^i) &= \max(\text{SS}(\mathcal{L}1_n^i, \mathcal{L}1_{n-1}^i), \text{SS}(\mathcal{L}2_n^i, \mathcal{L}2_{n-1}^i))) \\ \text{SS}(\mathcal{L}_n^i, \mathcal{L}_{n+1}^i) &= \max(\text{SS}(\mathcal{L}1_n^i, \mathcal{L}1_{n+1}^i), \text{SS}(\mathcal{L}2_n^i, \mathcal{L}2_{n+1}^i))\end{aligned}$$

Here $\mathcal{G} = 0.1\mathcal{F}^R + 0.7\mathcal{F}^G + 0.2\mathcal{F}^B$ where $(\mathcal{F}^R, \mathcal{F}^G, \mathcal{F}^B)$ are the red, green and blue components of $\mathcal{F}$ respectively. $\text{SS}(\mathcal{G}_n^i, \mathcal{G}_{n-1}^i)$ denotes the structure similarity between the two images $\mathcal{F}_n^i$ and $\mathcal{F}_{n-1}^i$. We only compare the structures of $(\mathcal{G}_n^i, \mathcal{G}_{n-1}^i)$ by turning off the luminance component of SSIM [16]. $\text{SS}(.,.)$ returns an a value between $0-1$ where 1 denotes identical similarity. Reflection is detected if $\mathcal{D}1_n^i$ is less than $\mathcal{T}1$.

Intrinsic Layer Extraction $\mathcal{D}_2$: Let INTR_i denote the intrinsic (reflectance) image extracted by processing the 50×50 i^{th} track using Yair technique [17]. In case of reflection the structure similarity between the observed mixture $\mathcal{F}_n^i$ and INTR_i should be low. Therefore, $\mathcal{F}_n^i$ is flagged as containing reflection if $\text{SS}(\mathcal{F}_n^i, \text{INTR}_i)$ is less than $\mathcal{T}2$.

Color Channels Independence $\mathcal{D}_3$: This approach measures the Generalized Normalized Cross Correlation (GNGC) [11] between the red and blue channels of the examined patch $\mathcal{F}_n^i$ to infer whether the patch is a mixture between two different layers or not. GNGC takes values between 0 and 1 where 1 denotes perfect match between the red and blue channels (M^R and M^B respectively). This analysis is applied to every image patch $\mathcal{F}_n^i$ and reflection is detected if $\text{GNGC}(M^R, M^B) < \mathcal{T}3$.

Image Sharpness Likelihood: $\mathcal{D}_4$,$\mathcal{D}_5$ Two approaches for examining image sharpness are used. The first, $\mathcal{D}4$, estimates the first order derivatives for the examined patch $\mathcal{F}_n^i$ and flags it as containing reflection if the mean of the gradient magnitude within the examined patch is smaller than a threshold $\mathcal{T}4$. The second approach, $\mathcal{D}5$, uses the sharpness metric of Ferzil et. al. [5] and flags a patch as reflection if its sharpness value is less than $\mathcal{T}5$.

SIFT Temporal Profile $\mathcal{D}_6$: This detector flags the examined patch $\mathcal{F}_n^i$ as reflection if its SIFT features [8] are undergoing high temporal mismatch. A vector **p** = [**x s g**] is assigned to every interest point in $\mathcal{F}_n^i$. The vector contains the position of the point **x**$= (x, y)$, scale and dominate orientation from the SIFT descriptor, **s** $= (\delta, o)$, and the 128 point SIFT descriptor **g**. Interest points are matched with neighboring frames using [8]. $\mathcal{F}_n^i$ is flagged as reflection if the average distance between the matched vectors **p** is larger than $\mathcal{T}6$.

Color Temporal Profile $\mathcal{D}_7$: This detector flags the image patch $\mathcal{F}_n^i$ as reflection if its grayscale profile does not change smoothly through time. The temporal change in color is defined as follows

$$\mathcal{D}7_n^i = \min(\|\mathcal{C}_n^i - \mathcal{C}_{n-1}^i\|, \|\mathcal{C}_n^i - \mathcal{C}_{n+1}^i\|) \quad (4)$$

Here $\mathcal{C}_n^i$ is the mean value for $\mathcal{G}_n^i$, the grayscale representation of $\mathcal{F}_n^i$. $\mathcal{F}_n^i$ is flagged as reflection if $\mathcal{D}7_n^i > \mathcal{T}7$.

AutoCorrelation Temporal Profile $\mathcal{D}_8$: This detector flags the image patch $\mathcal{F}_n^i$ as reflection if its autocorrelation is undergoing large temporal change. The temporal change in the autocorrelation is defined as follows

$$\mathcal{D}8_n^i = \sqrt{\min(\frac{1}{N}\|\mathcal{A}_n^i - \mathcal{A}_{n-1}^i\|^2, \frac{1}{N}\|\mathcal{A}_n^i - \mathcal{A}_{n+1}^i\|^2)} \quad (5)$$

$\mathcal{A}_n^i$ is a vector containing the autocorrelation of $\mathcal{G}_n^i$ while N is the number of pels in $\mathcal{A}_n^i$. $\mathcal{F}_n^i$ is flagged as reflection if $\mathcal{D}8_n^i$ is bigger than $\mathcal{T}8$.

Motion Field Divergence $\mathcal{D}_9$: D_9 for the examined patch $\mathcal{F}_n^i$ is defined as follows

$$\mathcal{D}9_n^i = \text{DFD}\left(\|\text{div}(\mathbf{d}(n))\| + \|\text{div}(\mathbf{d}(n+1))\|\right)/2 \quad (6)$$

DFD and $\text{div}(\mathbf{d}(n))$ are the Displaced Frame Difference and Motion Field Divergence for $\mathcal{F}_n^i$. $\mathbf{d}(n)$ is the 2D motion vector calculated using block matching. DFD is set to the minimum of the forward and backward DFDs. $\text{div}(\mathbf{d}(n))$ is set to the minimum of the forward and backward divergence. The divergence is averaged over blocks of two frames to reduce the effect of possible motion blur generated by unsteady camera motion. $\mathcal{F}_n^i$ is flagged as reflection if $\mathcal{D}_9 > \mathcal{T}9$.

4.3. Solving for l_n^i

4.3.1 Maximum Likelihood (ML) Solution

The likelihood is factorized as follows

$$P(\mathcal{F}|l) = P(l|\mathcal{D}_1)P(l|\mathcal{D}_{2-8})P(l|\mathcal{D}_9) \quad (7)$$

The first and last terms are solved using $\mathcal{D}_1 < \mathcal{T}1$ and $\mathcal{D}_9 > \mathcal{T}9$ respectively. $\mathcal{D}_{2-8}$ are used to form one strong detector $\mathcal{D}_s$ and $P(l|\mathcal{D}_{2-8})$ is solved by $\mathcal{D}_s > \mathcal{T}s$. We found that not including ($\mathcal{D}_1$,$\mathcal{D}_9$) in $\mathcal{D}_s$ generates better detection results than when included. Feature analyses of each detector are averaged over a block of three frames to generate temporally consistent detections. $\mathcal{T}9$ is fixed to 10 in all experiments. In Sec. 4.3.2 we avoid selecting particular thresholds for $(\mathcal{T}_1, \mathcal{T}s)$ by imposing spatial and temporal priors on the generated maps.

Calculating $\mathcal{D}_s$: The strong detector $\mathcal{D}_s$ is expressed as a linear combination of weak detectors operating at different

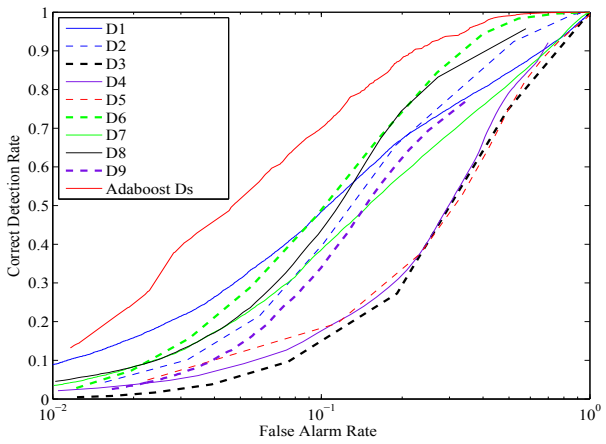

Figure 3. *ROC for D_{1-9} and D_s. The Adaboost detector D_s outperforms all other techniques and D_1 is the second best in the range of false alarms < 0.1.*

thresholds $\mathcal{T}$ as follows

$$P(l|\mathcal{D}_{2-8}) = \sum_{k=1}^{M} \mathcal{W}_{(V(k),\mathcal{T})} P(\mathcal{D}_{V(k)}|\mathcal{T}) \qquad (8)$$

Here M is the number of weak detectors (fixed to 20) used in forming $\mathcal{D}_s$ and $V(k)$ is a function which returns a value between 2-8 to indicate which detectors from $\mathcal{D}_{2-8}$ are used. k indexes the weak detectors in order of their importance as defined by the weights $\mathcal{W}$. $\mathcal{W}$ and $\mathcal{T}$ are learned through Adaboost [15] (see Tab. 1). Our training set consist of 89393 images of size 50×50 pels. Reflection is modeled in 35966 images each being a synthetic mixture between two different images.

Fig. 3 shows the the Receiver Operating Characteristic (ROC) of applying D_{1-9} and $\mathcal{D}_s$ on the training samples. $\mathcal{D}_s$ outperforms all the other detectors due to its higher correct detection rate and lower false alarms.

	$\mathcal{D}6$	$\mathcal{D}8$	$\mathcal{D}5$	$\mathcal{D}3$	$\mathcal{D}2$	$\mathcal{D}4$	$\mathcal{D}7$
$\mathcal{W}$	1.31	0.96	0.48	0.52	0.33	0.32	0.26
$\mathcal{T}$	0.29	6.76e−6	0.04	0.95	0.61	7	2.17

Table 1. *Weights $\mathcal{W}$ and operating thresholds $\mathcal{T}$ for the best seven detectors selected by Adaboost.*

4.3.2 Successive Refinement for Maximum A-Posteriori (MAP)

The prior $P(l|l^{\mathcal{N}})$ of Eq. 3 imposes spatial and temporal smoothness on detection masks. We create a MAP estimate by refining the sparse maps from the previous ML steps. We first refine the labeling of all the existing feature points $\mathcal{P}$ in each image and then use the overlapping 50×50 patches around the refined labeled points as a dense pixel map.

ML Refinement: First we reject false detections from ML which are spatially inconsistent. Every feature point $l = 1$ is considered and the sum of the geodesic distance from that site to the two closest neighbors which are labeled $l = 1$ is measured. When that distance is more than 0.005 then that decision is rejected i.e. we set $l = 0$. Geodesic distances allow the nature of the image material between point to be taken in to account more effectively and have been in use for some time now [10]. To reduce the computational load of this step, we downsample the image massively by 50 in both directions. This retains gross image topology only.

Spatio-Temporal Dilation: Labels are extended in space and time to other feature points along their trajectories. If $l_n^i = 1$, all feature points lying along the track i are set to $l = 1$. In addition, l is extended to all image patches ($\mathcal{F}_n$) overlapping spatially with the examined patch. This generates a denser representation of the detection masks. We call this step ML-Denser.

Hysteresis: We can avoid selecting particular thresholds $[\mathcal{T}_1, \mathcal{T}_s]$ for BIRD by applying Hysteresis using a set of different thresholds. Let $\mathcal{T}_H = [-0.4, 5]$ and $\mathcal{T}_L = [0, 3]$ denote a high and low configuration for $[\mathcal{T}_1, \mathcal{T}_s]$. Detection starts by examining ML-Denser at high thresholds. High thresholds generate detected points $\mathcal{P}_h$ with high confidence. Points within a small geodesic distance ($< D_{geo}$) and small euclidean distance ($< D_{euc}$) to each other are grouped together. Here we use (D_{geo},D_{euc}) $= (0.0025, 4)$ and resize the examined frames as mentioned previously. The centroids of each group is then calculated. Thresholds are lowered and a new detection point is added to an existing group if it is within D_{geo} and D_{euc} to the centroid of this group. This is the hysteresis idea. If however the examined point has a large euclidean distance ($> D_{euc}$) but a small geodesic distance ($< D_{geo}$) to the centroid of all existing groups, a new group is formed. Points at which distances $> D_{geo}$ and $> D_{euc}$ are regarded as outliers and discarded. Group centroids are updated and the whole process is repeated iteratively till the examined threshold reaches $\mathcal{T}_L$. The detection map generated at $\mathcal{T}_L$ is made more dense by performing **Spatio-Temporal Dilation** above.

Spatio-Temporal 'Opening': False alarms of the previous step are removed by propagating the patches detected in the first frame to the rest of the sequence along the feature point trajectories. A detection sample at fame n is kept if it agrees with the propagated detections from the previous frame. Correct detections missed from this step are recovered by running **Spatio-Temporal Dilation** on the 'temporally eroded' solution. This does mean that trajectories which do not start in the first frame are not likely to be considered, however this does not affect the performance in our real examples shown here. The selection of an optimal frame from which to perform this opening operation is the

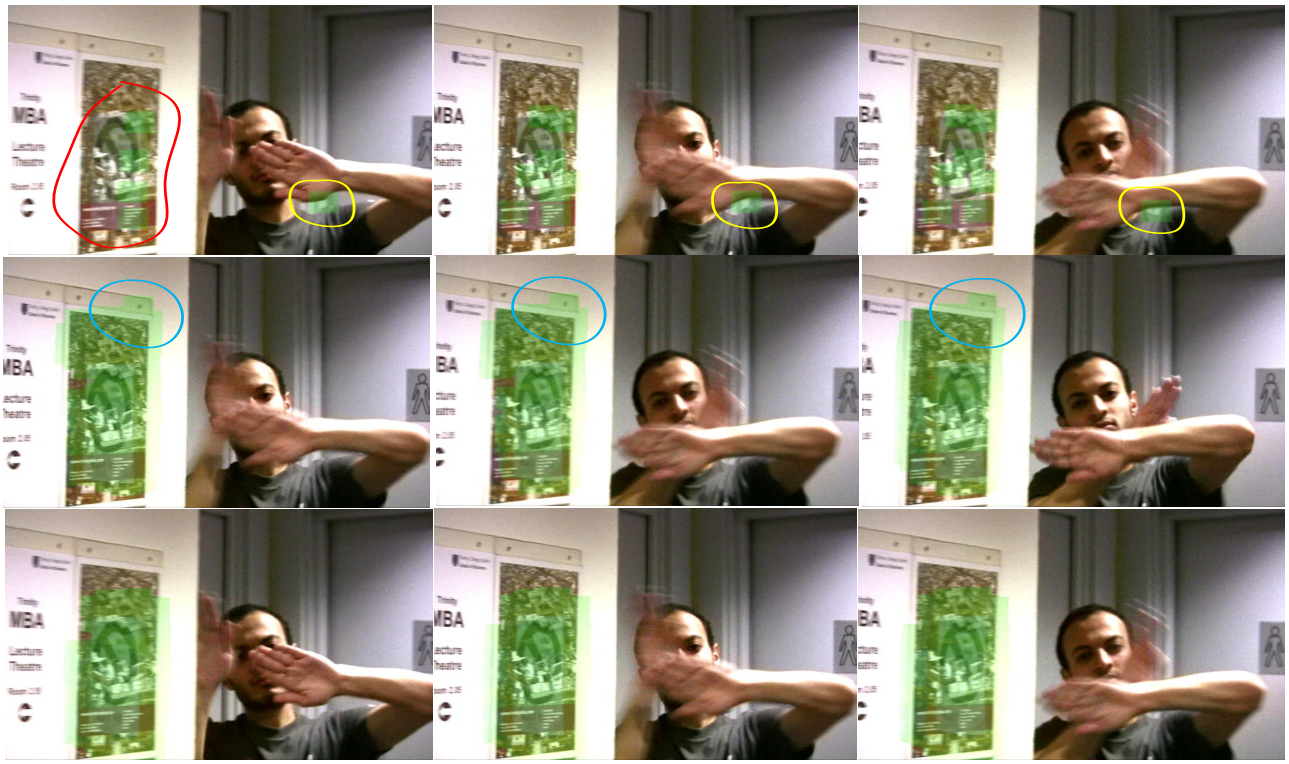

Figure 4. *From Top: ML (calculated at $(\mathcal{T}_1, \mathcal{T}_s) = (-0.13, 3.15)$), Hysteresis and Spatio-Temporal 'Opening' for three consecutive frames from the* **SelimH** *sequence. Reflection is shown in red and detected reflection using our technique is shown in green. Spatio-Temporal 'Opening' rejects false alarms generated by ML and by Hysteresis (shown in yellow and blue respectively).*

subject of future work.

5. Results

5.1. Reflection Detection

15 sequences containing 932 frames of size 576×720 are processed with BIRD. Full sequences with reflection detection can be found in www.sigmedia.tv/Misc/CVPR2011. Fig. 4 compares the ML, Hysteresis and Spatio-Temporal 'Opening' for three consecutive frames from the **SelimH** sequence. This sequence contains occlusion, motion blur and strong edges in the reflection (shown in red). The ML solution (first line) generates good sparse reflection detection (shown in green), however it generates some errors (shown in yellow). Hysteresis rejects these errors and generates dense masks with some false alarm (shown in blue). These false alarms are rejected by Spatio-Temporal 'Opening'.

Fig. 5 shows the result of processing four sequences using BIRD. In the first two sequences, BIRD detected regions of reflections correctly and discarded regions of occlusion (shown in purple) and motion blur (shown in blue). In **GirlRef** most of the sequence is correctly classified as reflection. In **SelimK1** the portrait on the right is correctly classified as containing reflection even in the presence of motion blur (shown in blue). Nevertheless, BIRD failed in detecting the reflection on the left portrait as it does not contain strong distinctive feature points.

Fig. 6 shows the ROC plot for 50 frames from **SelimH**. Here we compare our technique BIRD against DFD and Image Sharpness[5]. DFD, flags a region as reflection if it has high displaced frame difference. Image Sharpness flags a region as reflection if it has low sharpness. Frames are processed on 50×50 blocks. Ground truth reflection masks are generated manually and detection rates are calculated on pel basis. The ROC shows that BIRD outperforms the other techniques by achieving a very high correct detection rate of 0.9 for a false detection rate of 0.1. This is a major improvement over a correct detection rate of 0.2 and 0.1 for DFD and Sharpness respectively.

5.2. Frame Rate Conversion: An application

One application for reflection detection is improving frame rate conversion in regions of reflection. Frame rate conversion is the process of creating new frames from existing ones. This is done by using motion vectors to interpolate objects in the new frames. This process usually fails in regions of reflection due to motion estimation failure.

Fig. 7 illustrates the generation of a slow motion effect for the person's leg in **GirlRef** (see Fig. 5, third line). This is done by doubling the frame rate using the Foundry's Kro-

Figure 5. *Detection results of BIRD (shown in green) on, From top:* ***BuilOnWind*** $[10, 35, 49]$*,* ***PHouse*** *9-11,* ***GirlRef*** $[45, 55, 65]$*,* ***SelimK1*** *32-35. Reflections are shown in red. Good detections are generated despite occlusion (shown in purple) and motion blur (shown in blue). For* ***GirlRef*** *we replace Hysteresis and Spatio-Temporal 'Opening' with a manual parameter configuration of* $(\mathcal{T}_1, \mathcal{T}_s) = (-0.01, 3.15)$ *followed by a* ***Spatio-Temporal Dilation*** *step. This setting generates good detections for all examined sequences with static backgrounds.*

nos plugin [6]. Kronos has an input which defines the density of the motion vector field. The larger the density the more detailed the vector and hence the better the interpolation. However, using highly detailed vectors generate artifacts in regions of reflections as shown in Fig. 7 (second line). We reduce these artifacts by lowering the motion vector density in regions of reflection indicated by BIRD (see Fig. 7, third line). Image sequence results and more examples are available in www.sigmedia.tv/Misc/CVPR2011.

6. Conclusion

This paper has presented a technique for detecting reflections in image sequences. This problem was not addressed before. Our technique performs several analyses on feature point trajectories and generates a strong detector by combining these analyses. Results show major improvement over techniques which measure image sharpness and temporal discontinuity. Our technique generates high correct detection rate with rejection to regions containing complicated motion eg. motion blur, occlusion. The technique was fully automated in generating most results. As an application, we showed how the generated detections can be used to improve frame rate conversion. A limiting factor of our technique is that it requires source layers with strong distinctive feature points. This could lead to incomplete detections.

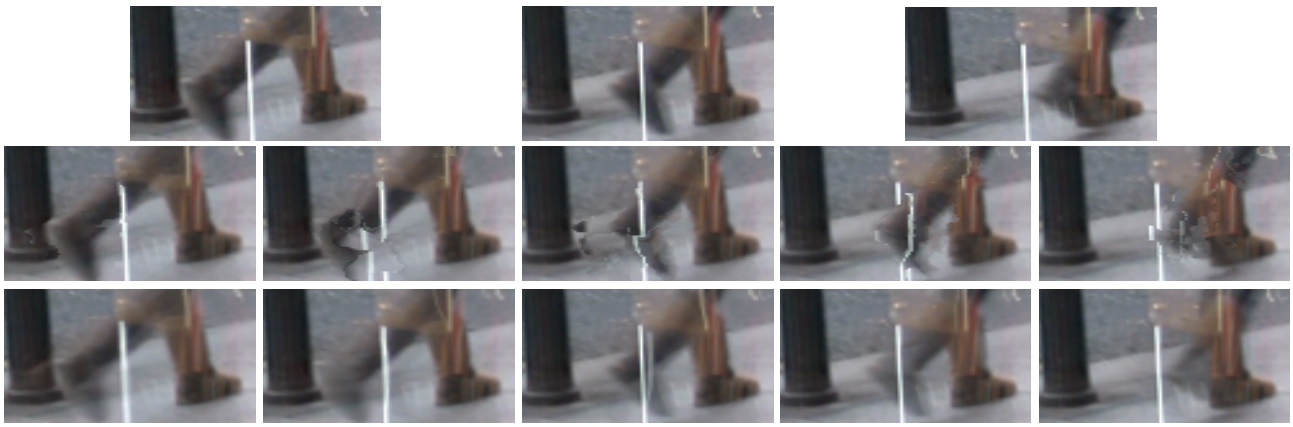

Figure 7. *Slow motion effect for the person's leg of* ***GirlRef*** *(see Fig: 5 third line). Top: Original frames 59-61; Middle: generated frames using the Foundry's plugin Kronos [6] with one motion vector calculated for every 4 pels; Bottom; with one motion vector calculated for every 64 pels in regions of reflection.*

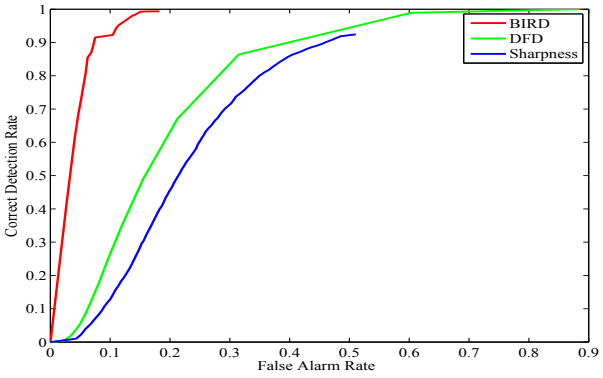

Figure 6. *ROC plots for our technique BIRD, DFD and Sharpness for* ***SelimH****. Our technique BIRD outperforms DFD and Sharpness with a massive increase in the Correct Detection Rate.*

Acknowledgment: This work is funded by the Irish Research Council for Science Engineering and Technology (IRCSET), EU i3DPOST Project FR7-211471, Science Foundation Ireland (SFI) PI Award 08/IN.1/I2112 and SFI TIDA Award 08/IN.1/I2112 TIDA 09.

References

[1] 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.

[2] N. Chen and P. De Leon. Blind image separation through kurtosis maximization. In *Asilomar Conference on Signals, Systems and Computers*, volume 1, pages 318–322, 2001.

[3] K. Diamantaras and T. Papadimitriou. Blind separation of reflections using the image mixtures ratio. In *ICIP*, pages 1034–1037, 2005.

[4] 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.

[5] R. Ferzli and L. J. Karam. A no-reference objective image sharpness metric based on the notion of just noticeable blur (jnb). *IEEE Trans. on Img. Proc. (TIPS)*, 18(4):717–728, 2009.

[6] T. Foundry. Nuke, furnace suite. www.thefoundry.co.uk.

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

[8] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *Int. J. Comput. Vision*, 60(2):91–110, 2004.

[9] B. D. Lucas and T. Kanade. An iterative image registration technique with an application to stereo vision (darpa). In *DARPA Image Understanding Workshop*, pages 121–130, 1981.

[10] 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.

[11] B. Sarel and M. Irani. Separating transparent layers through layer information exchange. In *European Conference on Computer Vision (ECCV)*, pages 328–341, 2004.

[12] B. Sarel and M. Irani. Separating transparent layers of repetitive dynamic behaviors. In *ICCV*, pages 26–32, 2005.

[13] R. Szeliski, S. Avidan, and P. Anandan. Layer extraction from multiple images containing reflections and transparency. In *CVPR*, volume 1, pages 246–253, 2000.

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

[15] P. Viola and M. Jones. Robust real-time object detection. In *International Journal of Computer Vision*, 2001.

[16] Z. Wang, A. Bovik, H. Sheikh, and E. Simoncelli. Image quality assessment: from error visibility to structural similarity. *TIPS*, 13(4):600–612, April 2004.

[17] Y. Weiss. Deriving intrinsic images from image sequences. In *ICCV*, pages 68–75, 2001.