Reflective Features Detection and Hierarchical Reflections Separation in Image Sequences

Di Yang*, Srimal Jayawardena†, Stephen Gould*, Marcus Hutter*
*Research School of Computer Science
The Australian National University
Canberra, Australia
†QLD Centre for Advanced Technologies
CSIRO, Brisbane, Australia

***Abstract*—Computer vision techniques such as Structure-from-Motion (SfM) and object recognition tend to fail on scenes with highly reflective objects because the reflections behave differently to the true geometry of the scene. Such image sequences may be treated as two layers superimposed over each other - the non-reflection scene source layer and the reflection layer. However, decomposing the two layers is a very challenging task as it is ill-posed and common methods rely on prior information. This work presents an automated technique for detecting reflective features with a comprehensive analysis of the intrinsic, spatial, and temporal properties of feature points. A support vector machine (SVM) is proposed to learn reflection feature points. Predicted reflection feature points are used as priors to guide the reflection layer separation. This gives more robust and reliable results than what is achieved by performing layer separation alone.**

I. Introduction

Objects with visual artifacts introduced by highly reflective surfaces plague many computer vision algorithms like Structure-from-Motion (SfM) and object recognition. Such techniques perform poorly on scenes with highly reflective structures because the reflections behave differently to the true geometry of the scene. Common objects with highly reflective subjects may include, but are not limited to, a stainless steel mug, a puddle, the windows of a skyscraper and vehicles. Figure 1 shows photographs of objects with reflective mediums (*e.g.* window glass and metallic panels) resulting in different types of reflections. Our focus of interest is on video footage of vehicles, because vehicles are always made by reflective materials such as metallic panel and transparent glasses, which can yield almost all kinds of reflections from spectacular reflection to diffuse reflection, from glossy reflection to total internal reflection, and also understanding road scenes is a very important practical problem. The proposed method should generalise to other objects as well since we do not make any assumptions on the type of object in our method. Imagery of highly reflective objects can be considered to be a superposition of two layers - the reflection layer and the non-reflective scene source layer [1], [2], [3].

Layer separation aims to separate the input image into the two unknown layers, However, this is an ill-posed problem and common methods use strong priors. Reflections are usually the low-contrast layer, appearing as mirror-images of surrounding objects (*e.g.* trees, roads, walls, sky). Additionally, the reflection layer is typically warped according to the shape of the reflective surface. Furthermore, this low-contrast layer may be superimposed over parts of the object containing semi reflective mediums (*i.e.* glass parts in a vehicle). Therefore extracting characteristics specific to reflections proves to be a very challenging task.

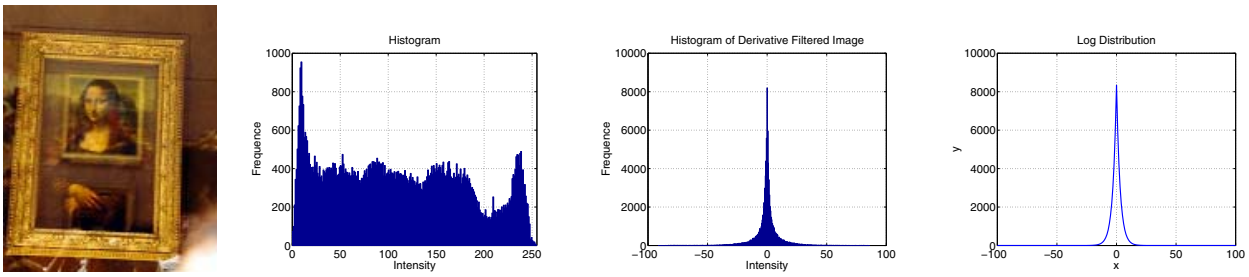
Fig. 1: Reflections are yielded by objects with reflective medium. As many common objects, cars have very reflective surfaces (*e.g.* metallic paint and glass) which produce almost all kinds of reflections, such as specular, diffuse, glossy and total internal reflections.

3 proposed a user-interactive separation of reflections that utilises manually marked labels (indicating the layer) as sparse prior information. However, because only a small number of edges are marked by the user, the problem is still ill-posed. Moreover, with large video sequences, manual labelling is not a feasible option. 12 presented a technique for detecting regions containing reflections in image sequences. However, regions is too generic to formulate features that could assist reflection separating method to generate accurate and reliable results.

In this paper, we propose a method to automatically detect reflective features that later assist recent reflection separating method so as to replace manual labelling. Predicted reflection feature points are used as priors for layer separation in the frames of the video sequence.

II. Background And Related Work

The proposed method consists of two aspects - reflection layer separation and reflection detection. We review prior work related to both of these as follows.



(a) (b) (c) (d)

Fig. 2: Natural image intrinsic property (a) Mona Lisa (ML); (b) Histogram of ML; (c) Histogram of horizontal derivative filtered ML; (d) Laplacian density function used to model.

A. *Reflection layer separation*

An image with reflections $\mathcal{M}$ can be modeled as a linear combination of the source layer $\mathcal{L}_1$ and the reflection layer $\mathcal{L}_2$ as: $\mathcal{M}(x,\ y) = \alpha\mathcal{L}_1(x,\ y) + \beta\mathcal{L}_2(x,\ y)$. The goal of reflection layer separation is to decompose image $\mathcal{M}$ into the two independent layers $\mathcal{L}_1$ and $\mathcal{L}_2$, where α and β are coefficients default setting to 1. As mentioned before, this is an ill-posed problem. Therefore certain assumptions are introduced to make the problem solvable. One commonly used assumptions exploits remarkably robust intrinsic property of the natural scene. As proposed by 10, the output of natural images processed by derivative filters f tend to be sparse. Figure 2 shows this property, where the histogram of the derivative filtered image $f * \mathcal{M}$ (where $*$ denotes convolution) has a peak at zero which drops much more quickly than in a Gaussian distribution. This fact can be mathematically formulated in a number of ways. The most widely used method is to use a Laplacian probability density function to fit the histogram [2], [4]. Therefore, decomposing $\mathcal{M}$ may be done by considering a finite Laplacian mixture model describing the histogram of the derivative filtered image $f * \mathcal{M}$ based on the assumption of derivative filters being independent over space and orientation as follows.

$$\begin{aligned}\rho(f * \mathcal{M}) &= \rho(f * (\mathcal{L}_1 + \mathcal{L}_2)) \approx \rho(f * \mathcal{L}_1) + \rho(f * \mathcal{L}_2) \\ &= \frac{\pi_1}{2s_1} e^{-|\tilde{x}|/s_1} + \frac{\pi_2}{2s_2} e^{-|\tilde{x}|/s_2}, \end{aligned} \tag{1}$$

where $\rho(\cdot)$ denotes the gradient magnitude histogram of the derivative filtered image, and $\tilde{x}$ indicates gradient magnitude. Two Laplacian distributions are used to describe the histograms of the derivative filtered source layer $\mathcal{L}_1$ and the reflection layer $\mathcal{L}_2$.

Separation of reflections in image sequences Based on the above assumption, an approach proposed by 4 separates the reflections from an image sequence where source layer $\mathcal{L}_1$ is almost stationary and reflection layer $\mathcal{L}_2$ changes over time. Since the approach requires the source layer has to be fixed over time, it has significantly limited application. We incorporate a feature based on this approach in our method as explained in Section III-C, but not required either layer to be fixed.

Separation of reflections in a single image As mentioned previously, 3 proposed an approach for separating the reflections from a single image. Their method makes the same assumption on the histogram of the derivative filtered image but requires user interaction. Consider

$$\mathcal{E}(\mathcal{L}_2) = \min_{\mathcal{L}_1} \left\| \rho(f * \mathcal{M}) - \sum_{i=1}^{2} \rho(f * \mathcal{L}_i) \right\|_2, \tag{2}$$

where $\mathcal{E}(\cdot)$ indicates the least squared error probability of derivative image and the synthesised layers. Using decomposition of $\mathcal{M}$ as a linear constraint, $\sum_{i=1}^{2} \rho(f * \mathcal{L}_i)$ is written as:

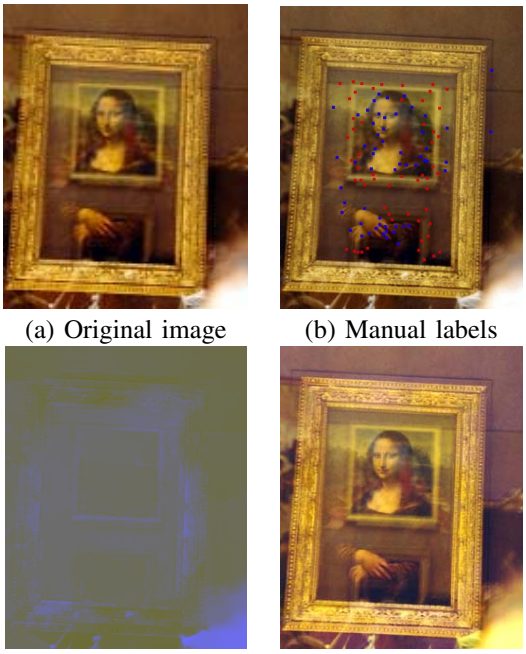
$$\begin{aligned}\sum_{i=1}^{2} \rho(f * \mathcal{L}_i) &= \rho(f * \mathcal{L}_1) + \rho(f * (\mathcal{M} - \mathcal{L}_1)) \\ &\quad + \lambda\rho(f * (\mathcal{M}|_{R_1} - \mathcal{L}_1|_{R_1})) + \lambda\rho(f * \mathcal{L}_1|_{R_2}),\end{aligned} \tag{3}$$

where R_1 and R_2 denotes the location of non-zero magnitude pixels in $f * \mathcal{L}_1$ and $f * \mathcal{L}_2$, respectively. Thus the last two terms both equal 0 in the ideally situation, which are used to enforce the linear constraint. equation 2 alone is insufficient to minimise $\mathcal{E}(\cdot)$, as the problem is still ill-posed and extra priors are required. With sufficient prior information, the most likely solution for equation 1 can be found using convex optimisation [2]. Therefore obtaining the right priors play an important role. Because equation 1 is obtained by applying the derivative filter to the image (*e.g.* extracted edge features), edge features play an important role. Therefore, the priors can be obtained by asking users to manually label points on the edges as belonging to the reflection layer or the source layer. These priors help the method to distinguish between the reflection layer $\mathcal{L}_1$ and source layer $\mathcal{L}_2$ (see Figure 3(b)).

However, as shown in Figure 3, even with such labels obtained from users, the problem remains ill-posed for images with a large number of pixels (in the order of a million pixels), and leads to the approach being extremely fragile and sensitive to user labelling errors. Whereas, our work is to solve this problem by introducing reflective features detection procedure which can eliminate human interaction and efficiently label as many features as possible so as to significantly improve its performance.

B. *Reflection detection in image sequences*

The goal here is to detect regions in the image containing the reflections. However, the reflections are essentially mirror images that could be anything (*e.g.* trees, human, sky, cloud, etc.). As mentioned previously, it is very challenging to detect the reflections in a single image without any priors. An image sequence, on the other hand, provides more information for reflection detection [3].



(a) Original image (b) Manual labels

(c) Reflection layer (d) Source layer

Fig. 3: Insufficient priors provided by the user resulting in failed layer separation. (a) a photo of Mona Lisa with interference of the reflections; (b) insufficient manual-labeled priors (red for reflection, blue for source); (c) the failed separated reflection layer; (d) the failed separated source layer.

Our approach is inspired by the framework proposed by 12 which employs several temporal and intrinsic features along with Bayesian inference to detect regions containing reflections in image sequences. Large reflection regions can make the region-wise reflection detection meaningless as the whole picture could be covered by reflections as shown in Figure 3(a). Moreover, regions are too generic to formulate priors that can be used in the reflection layer separation method of 3. Therefore, instead of regions, we propose a framework to automatically determine if feature points lying on image edges belong to the reflection layer or the source layer as shown in Figure 4.

III. Reflective Feature Detection

Our method for reflective features detection aims to determine if image edges belong to reflections by analysing the trajectories of feature points lying on image edges by tracking them through the image sequence. Labels of edge points are subsequently used to formulate priors required to guide the reflection layer separation method of 3.

A. Obtaining patch matches

We use DAISY [5], [6], a fast local descriptor for dense matching, to track feature points on image edges through the image sequences. Let $e_n^j = \{p_n^{(1,j)}, p_n^{(2,j)}, \ldots, p_n^{(i,j)}\}$ denote the j^{th} edge in the n^{th} frame detected by Canny method. Let $p_n^{(i,j)} \in e_n^j$ indicate the i^{th} point on the j^{th} edge tracked in the n^{th} frame. A 21×21 local patch centred at $p_n^{(i,j)}$ denoted by $\mathcal{N}(p_n^{(i,j)})$ is to compute the DAISY descriptor at an edge point.

If feature point $p_n^{(i,j)}$ can be tracked via DAISY method 2 through more than three discontinuous frames within an interval of 5 frames, a set of three matching patches $\tilde{\mathcal{N}}(p_n^{(i,j)}) = \{\mathcal{N}(p_{n-5}^{(i,j)}), \mathcal{N}(p_n^{(i,j)}), \mathcal{N}(p_{n+5}^{(i,j)})\}$ is used for our analysis, where the indices of the points and their edge in the other two frames (*i.e.* $n-5$ and $n+5$) are re-organised to match indices of them in n-th frame. As a result, a set of binary labels $l(p_n^{(i,j)}) \in \{0, 1\}$ with 1 for reflection and 0 otherwise, is assigned to each edge point as shown in Figure 4. For clarity, the indices n, i, and j will be dropped in the following sections.

B. Bayesian Inference

The framework derives an estimate for $l(p)$ using the posterior $\mathcal{P}(l(p) \mid \mathcal{F}(\tilde{\mathcal{N}}(p)), e)$.The posterior can be factorised in the Bayesian framework as follows.

$$\mathcal{P}(l(p) \mid \mathcal{F}(\tilde{\mathcal{N}}(p)), e) = \mathcal{P}(\mathcal{F}(\tilde{\mathcal{N}}(p)) \mid l(p), e)\mathcal{P}(l(p) \mid e) \tag{4}$$

where $\mathcal{F}(\cdot)$ denotes a set of features for a matched patch $\tilde{\mathcal{N}}(p)$. In likelihood term $\mathcal{P}(\mathcal{F}(\tilde{\mathcal{N}}(p)) \mid l(p), e)$, $\mathcal{F}(\tilde{\mathcal{N}}(p))$ is a vector containing seven different features described later based on approaches $\mathcal{F} = \{F_1, \ldots, F_7\} \in \mathcal{R}^7$ applied to matched patch $\tilde{\mathcal{N}}(p)$. Prior $\mathcal{P}(l(p) \mid e)$ enforces smoothness constraints based on spatial proximity.

C. Maximum likelihood estimation for reflective features

As mentioned previously, likelihood term $\mathcal{P}(\mathcal{F}(\tilde{\mathcal{N}}(p)) \mid l(p), e)$ in equation 4 is based on seven different features. A pre-trained Support Vector Machine (SVM) with a Radial Basis Function (RBF) kernel was used to predict the labels of patches based on these features. For each image sequence, the training set containes over $60,000$ matched patches of size $\mathcal{N}(p) = 21 \times 21$. The features ($F_1$ to F_7) can be divided into three categories - intrinsic features, image sharpness measurements and temporal discontinuity measurements.

Intrinsic layer extraction F_1: Intrinsic layer extraction [4] is originally intended to decompose each frame in an image sequence of a stationary object with illumination changes into a stationary source layer $\mathcal{L}_1$ and a set of reflection layers $\mathcal{L}_2^n$ as follows.

$$\mathcal{M}^n(x,\ y) = \mathcal{L}_1(x,\ y) + \mathcal{L}_2^n(x,\ y), \tag{5}$$

where $\mathcal{M}^n$ and $\mathcal{L}_2^n$ are the n^{th} frame and its reflection in the image sequence respectively. This approach, however, assumes that the source layer $\mathcal{L}_1$ is nearly constant throughout the image sequence. equation 5 is ill-posed. In order to make it solvable, a finite mixture model of Laplacian density functions $\rho(f * \mathcal{L}) = \frac{\pi}{s} e^{-\alpha|\tilde{x}|/s}$ is employed to describe $\mathcal{M}^n$ as the histogram of derivative filtered image approximates a Laplacian distribution, which is peaked at zero and drops much faster than in a Gaussian distribution (Section II-B). It is assumed that the outputs from derivative filters have a Laplacian density that is independent over space and time. Thereby, applying the same derivative filter f to each frame in the image sequence,

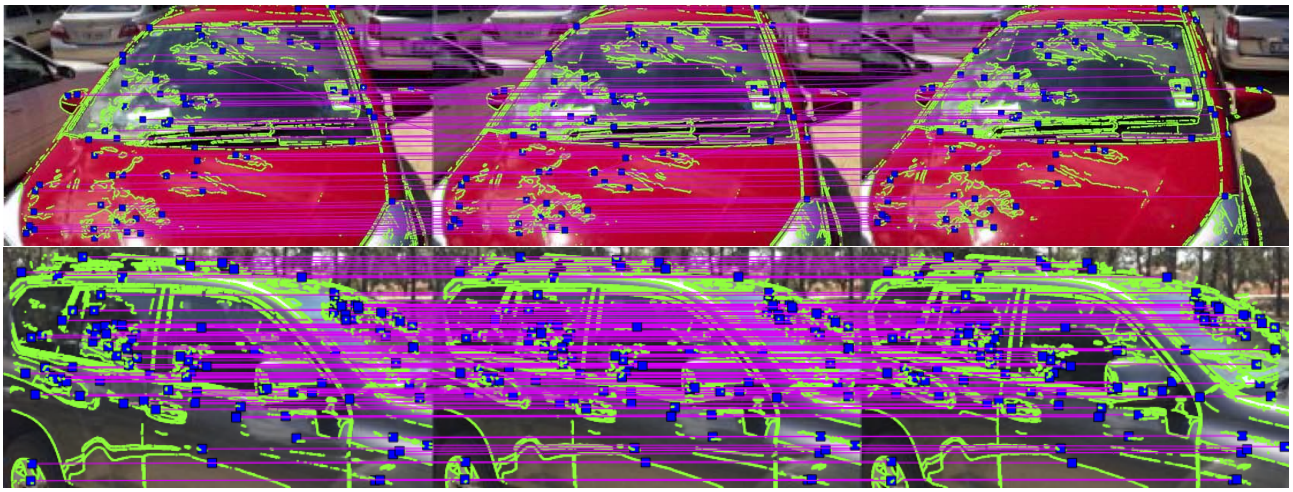


Fig. 4: Example matched patches over 3 discontinuous frames with an interval of 5 frames in between: Green dots denote feature points on the edges in each frame. Magenta lines show corresponding feature points across frames. Blue rectangles represent patches centered at the feature points. For visual clarity, only one feature point per each edge is shown.

the output of each filtered frame $\mathcal{M}^n$ can be formulated as follows.

$$f * \mathcal{M}^n(x,\ y) = f * \mathcal{L}_1(x,\ y) + f * \mathcal{L}_2^n(x,\ y) \tag{6}$$

where $*$ denotes convolution. $\mathcal{L}_1^f$ denotes $\mathcal{L}_1^f = \mathcal{L}_1(x,\ y) * f$. Assuming that filtered $\mathcal{L}_2^n$ has a Laplacian distribution and is independent over space and time, the maximum likelihood estimate of the filtered source $\hat{\mathcal{L}}_1^f$ is given by: $\hat{\mathcal{L}}_1^f(x,\ y) = \text{median}\,\{\mathcal{M}^n(x,\ y)\}$, because the likelihood term $\mathcal{P}(\mathcal{M}^n * f|\mathcal{L}_1 * f)$ is obtained by,

$$\begin{aligned}\mathcal{P}(\mathcal{M}^n * f|\mathcal{L}_1 * f) &= \frac{1}{s}\prod_n e^{-\alpha|(\mathcal{M}^n - \mathcal{L}_1) * f|/s} \\ &= \frac{1}{s} e^{-\alpha \sum_n |\mathcal{M}^n * f - \mathcal{L}_1^f|/s}.\end{aligned} \tag{7}$$

In order to maximise the likelihood term, the sum of the absolute value $\sum_n \left|\mathcal{M}^n * f - \mathcal{L}_1^f\right|$ is minimised in the median term.

In our case, this process is conducted over matched patches $\tilde{\mathcal{N}}(p)$ in order to obtain the reflectance layer $\hat{\mathcal{L}}_1^f$. The similarity measurement F_1 between the reflectance layer $\mathcal{L}_1$ and the source layer $\mathcal{L}_2$ is a reflective feature. When $p_n^{(i,j)}$ belongs to the reflection layer, the similarity measurement takes a lower value.

Colour channels independence F_2: The generalised normalised cross correlation (*GNCC*) [1], derived from the normalised grayscale correlation (*NGC*), is used to examine the blue $\mathcal{C}_B$ and red $\mathcal{C}_R$ channels of the each examined patch $\mathcal{N}(p) \in \tilde{\mathcal{N}}(p)$, in order to ascertain whether the patch belongs to two different layers. That is because the red and blue channels of a colour image generally uncorrelated [7]. *NGC* is obtained as follows.

$$\begin{aligned} NGC(\mathcal{C}_B, \mathcal{C}_R) &= \frac{Corr(\mathcal{C}_B, \mathcal{C}_R)}{\sqrt{Var(\mathcal{C}_B)Var(\mathcal{C}_R)}} \\ Var(\mathcal{N}_B) &= \frac{1}{N}\sum_{\{x,y\}} \mathcal{C}_B^2(x,y) - \overline{\mathcal{C}_B^2} \\ Corr(\mathcal{C}_B, \mathcal{C}_R) &= \frac{1}{C}\sum_{\{x,y\}} \mathcal{C}_B(x,y)\cdot\mathcal{C}_R(x,y) - \overline{\mathcal{N}_B}\cdot\overline{\mathcal{C}_R} \end{aligned} \tag{8}$$

where $Corr(\cdot,\cdot)$ is the correlation between the blue and red channels of the examined patch $\mathcal{N}(p)$ and $Var(\cdot)$ is the variance of each channel in a patch. The *GNCC* consists of a set of *NGC* measurements obtained on each small 3×3 window in the 21×21 patch $\mathcal{N}(p)$ as:

$$GNCC(\mathcal{C}_B, \mathcal{C}_R) = \frac{\sum_{k=1} Corr_k^2(\mathcal{C}_B, \mathcal{C}_R)}{\sum_{k=1} Var_i(\mathcal{C}_B)Var_k(\mathcal{C}_R)} \tag{9}$$

$F_2 = GNCC$ will return values in the interval $[0, 1]$, where 1 indicates a perfect match between red and blue channels and 0 indicates a complete mismatch. If point p belongs to the reflections, the *GNCC* value should be small.

Multi-scale cue combination F_3: As proposed by 7, this feature is computed by estimating the derivatives for the patch converted to gray-scale using $\mathcal{G}(p) = 0.4\mathcal{C}_R(p) + 0.2\mathcal{C}_G(p) + 0.4\mathcal{C}_B(p)$. We apply set of 8 oriented even and odd symmetric Gaussian derivative filters with $\sigma = 0.04$ and a centre-surround filter (difference of two Gaussians with $\sigma_1 = 0.04$ and $\sigma_2 = 0.07$ respectively). The optimal derivative is obtained by considering the maximal derivative at the given patch location as follows.

$$mPb(\mathcal{G}(p)) = \arg\max_{\theta}\{\mathcal{G}(p) * G(\sigma, \theta)\} \tag{10}$$

where $G(\sigma, \theta)$ is a set of candidate oriented Gaussian derivative filters, and $mPb(\cdot)$ is the optimal derivative. $F_3 = E(mPb(\cdot))$ should be low for a patch for the reflection layer.

CPBD sharpness metric F_4: The CPBD sharpness metric is the cumulative probability of image blur detection as proposed

by 8. This measurement aims to assess video quality by measuring image sharpness. In our case, sharpness is measured over matched patches $\tilde{\mathcal{N}}(p)$. A feature point p_n belonging to the reflection layer will have a relatively low sharpness measurement.

DAISY Temporal Profile F_5: As mentioned previously, the DAISY algorithm [5], [6] uses a dense descriptor, which can be used to track point features $p_n^{(i,j)}$ through the image sequence. DAISY can also be used to measure the degree of temporal discontinuity.

EPG Outliers F_6: Given two images of a 3D scene, its epipolar geometry (EPG) gives information about the camera setup in a projective sense. The EPG can be used to infer knowledge about the 3D scene and is used in 3D scene reconstruction and stereo matching. The key insight is that matched points in reflections need not in general agree with EPG of the scene. A given point in one image will lie on its epipolar line (which is a projection of the back projected ray from the first image on to the second image) in the second image,. The EPG is described algebraically using the fundamental matrix F [8], which is based on this relationship. Ideally, suppose two points on $\mathcal{M}$ and $\mathcal{M}'$ have homogeneous coordinates $X_n = [p_n, \mathbf{1}]$ and $X_{n-5} = [p_{n-5}, \mathbf{1}]$. F is a 3×3 matrix of rank 2 such that: $X_{n-5}^T F X_n = 0$.

Given a set of noisy point correspondences, the fundamental matrix F may be robustly computed using robust estimation methods such as RANdom Sample Consensus (RANSAC) [9]. Such methods are robust in the presences of noisy outliers with considerable errors. In our case, we used M-estimator Sample Consensus (MSAC) [10] as it is known to converge faster than standard RANSAC. When we use the error distance (proposed by [11]) of the outliers (*i.e.* reflective points in our case) to the recovered F to detect points belonging to reflections and to assign a binary label, where $F_6 = 0$ if p is outliers, and $F_6 = 1$ if otherwise.

Colour temporal profile F_7: This feature measures the gray-scale profile of matched patches $\tilde{\mathcal{N}}(p)$. If the gray-scale profile does not change smoothly through the image sequences, the centre point p of its patch $\mathcal{N}(p)$ is likely the reflection. The temporal change is defined as:

$$\min \{MSE(\mathcal{G}_n(p)), \mathcal{G}_{n-5}(p)), MSE(\mathcal{G}_n(p)), \mathcal{G}_{n+5}(p))\} \tag{11}$$

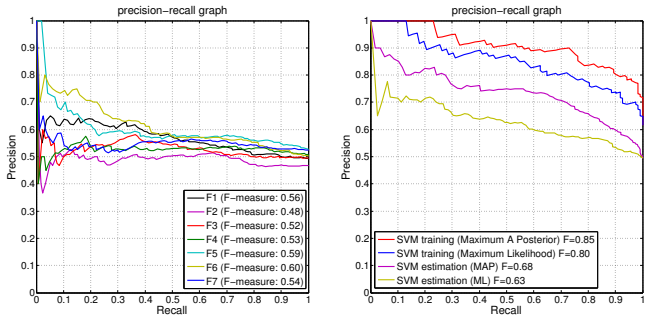
where $MSE(\cdot)$ denotes mean squared error between two gray-scale representation patches.

Features summary: Figure 5(a) shows the precision and recall (PR) plot of applying representing features F_1 - F_7 alone on the training samples.

D. Spatial priors for Bayesian refinement

The prior $\mathcal{P}(l(p) \mid e)$ in equation 4 imposes spatial smoothness constraints on the labeled feature points. This which enable us to create a Maximum A Posterior (MAP) estimate by refining the sparse maps from previous maximum likelihood (ML) estimation.

For the each given edge e_n in n-th frame and its labeled feature points $l(p_n^1, l(p_n^2, \ldots, l(p_n^i$, we employ a Markov-Chain



(a) (b)

Fig. 5: (a) Precision-Recall plot for each representing features F_1 - F_7 alone in reflective detection training process. F_6 is the best feature with F-measurement (0.60), followed by F_5 (0.59); (b) Precision-Recall plot for our technique with and without spatial priors for both training and test sets.

model in order to account for the continuity of the edge.

$$\mathcal{P}(l(p^{1,\ldots,n}) \mid e) = \frac{1}{Z} \prod_{i=1}^{n} Q(l(p^i)) \prod_{i=1}^{n} R(l(p^i), l(p^{i+1})) \tag{12}$$

$$Q\left(l(p^i)\right) = e^{-\phi(l(p^i,\xi))} \; R\left(l(p^i), l(p^{i+1}\right) = e^{-\psi(l(p^i),l(p^{i+1}))}$$

This refinement term will penalise false detections by ensuring that each edge contains a set of continuously labeled feature points. $\phi(\cdot,\cdot)$ is a binary function to indicate whether the point's label should be switched to the opposite, which is defined as follows:

$$\phi(l(p^i), \xi) = \begin{cases} \alpha, & \mathcal{T}(i) > \xi \\ 0, & \text{otherwise} \end{cases}$$

$$\mathcal{T}(i) = \frac{1}{W} \sum_{k=1}^{n} \delta(|l(p^i) - l(p^k)|) \frac{1}{\sqrt{2\pi}} e^{-(k-i)^2/2} \tag{13}$$

where $\mathcal{T}(p^i)$ is the transition probability function and $\delta(\cdot)$ is the impulse function. While $\psi(\cdot,\cdot)$ is a penalty function that flags the continuity, which is defined as follows.

$$\psi(l(p^i), l(p^{i+1})) = \begin{cases} \beta, & l(p^i) \neq l(p^{i+1}) \\ 0, & l(p^i) = l(p^{i+1}) \end{cases}, \tag{14}$$

where $\lambda > 0$ is a constant. In order to solve the problem, we obtain a MAP estimate as described in equation 4. Thereby, we need to maximise the above equation as well. This is equivalent to seek the best threshold ξ as follows.

$$\arg\min_{\xi} \left\{ \sum_i \phi(p^i, \xi) + \sum_i \psi\left(p^i, p^{i+1}\right) \right\}, \tag{15}$$

where α and β are coefficients so as to balance smoothness and transition movements. The optimisation is conducted using the Viterbi algorithm [12], which is a dynamic programming algorithm to find the proper threshold.

IV. Experimental Results and Conclusion

A. Reflective Feature Detection

Six sequences of different vehicles containing 984 frames of size of 960×540 are processed with our proposed method to distinguish reflective features. For each sequence, 21 frames

(a) Reflective features (b) Reflection layer (c) Source layer

Fig. 6: Results of Reflection Layer Separation using the Reflective Features. Two results are shown row-wise with the columns showing: (a) reflective features overlapping the original image (green = reflection, blue = source); column (b) reflection layer (c) source layer

are manually labelled which form our training set. Besides, each sequence contains 6 labelled frames as testing set.

Selected results are shown in Figure 6(a), in which edges in green denote reflections and blue ones indicate objects. Most edges labelled as source layer $\mathcal{L}_1$ are continuous boundaries of objects. For example, boundaries of screen window and engine hood are detected as object in Figure 6(a)-top. Moreover, some small parts, for example the vehicle registration card attached to the screen window, are successfully detected as objects. In the other hand, edges belonging to the reflections $\mathcal{L}_2$ are relatively short, trivial, and coarse, usually located inside the boundaries of the vehicle parts.

Figure 5(b) shows the precision and recall (PR) plot for 20 frames from the sequence of the red car in Figure 6(a). Here, we demonstrate the performance of the influence of incorporating the spatial prior into our method under both training and testing circumstances. In the training process, the F-measurement of our method without help of spatial prior is 0.80 while the F-measurement of the method incorporated with spatial prior is over 0.85. In testing process, the F-measurements are 0.68 and 0.63 for with and without spatial prior incorporating, respectively.

B. Reflection Layer Extraction: An application

After successfully detecting reflective features of each frame in the sequences, these features can now be applied as the priors required for the reflection layer separation method proposed by 3 allowing to replace massive human interactions and significantly simplify the reflection layer separation process as well as increasing its accuracy. That is because their method requires users to manually label some of the edge features in order to produce a convincing resulting reflection layer. Such human interaction can be very subjective, thus making the reflection layer separation quite fragile and unreliable, and its results unrepeatable. Additionally, the required human interaction scales exponentially with the size/resolution of the imagery.

In Figure 6, we show that introducing our reflective feature detection procedure to generate priors, makes reflection layer separation algorithm more robust and reliable. In Figure 6(bottom - c) thanks to the labelled edge-features yielded by our reflective feature detection approach, extracted source layer has preserved many interior parts (*e.g.* seats) of the vehicles behind the transparent medium (*e.g.* window glass).

C. Conclusion

In this paper, we have presented a technique to automatically detect reflective features in the image sequence. Such reflective features can be used to guide reflection separation algorithm to produce reliable and accurate results without human interaction. Our technique employed several analysis to extract intrinsic, temporal, and spatial features from the image sequence so as to enable reflective features detection via support vector machine. Experimental results show our technique is fully capable of detecting reflective features and separating reflections.

REFERENCES

[1] B. Sarel and M. Irani, "Separating transparent layers through layer information exchange," in *ECCV 2004*, 2004, pp. 328–341.

[2] A. Levin and Y. Weiss, "User assisted separation of reflections from a single image using a sparsity prior," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 29, no. 9, pp. 1647–1654, 2007.

[3] M. Ahmed, F. Pitie, and A. Kokaram, "Reflection detection in image sequences," in *CVPR 2011*, 2011, pp. 705–712.

[4] Y. Weiss, "Deriving intrinsic images from image sequences," in *ICCV 2001*, vol. 2, 2001, pp. 68–75.

[5] E. Tola, V. Lepetit, and P. Fua, "DAISY: An Efficient Dense Descriptor Applied to Wide Baseline Stereo," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 32, no. 5, pp. 815–830, 2010.

[6] E. Tola, V.Lepetit, and P. Fua, "A fast local descriptor for dense matching," in *Proceedings of Computer Vision and Pattern Recognition*, 2008.

[7] B. Sarel and M. Irani, "Separating transparent layers of repetitive dynamic behaviours," in *ICCV 2005*, vol. 1, 2005, pp. 26–32.

[8] R. I. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*, 2nd ed. Cambridge University Press, 2004.

[9] M. A. Fischler and R. C. Bolles, "Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography," *Commun. ACM*, vol. 24, no. 6, pp. 381–395, 1981.

[10] M. Rogers and J. Graham, "Robust active shape model search," in *ECCV 2002*, vol. 2353, 2002, pp. 517–530.

[11] Z. Zhang, R. Deriche, O. Faugeras, and Q.-T. Luong, "A robust technique for matching two uncalibrated images through the recovery of the unknown epipolar geometry," *Artificial intelligence*, vol. 78, no. 1, pp. 87–119, 1995.

[12] A. Viterbi, "Error bounds for convolutional codes and an asymptotically optimum decoding algorithm," *IEEE Transactions on Information Theory*, vol. 13, no. 2, pp. 260–269, 1967.