Image-Based Models for Specularity Propagation in Diminished Reality

Souheil Hadj Said, Mohamed Tamaazousti, and Adrien Bartoli

Abstract—The aim of Diminished Reality (DR) is to remove a target object in a live video stream seamlessly. In our approach, the area of the target object is replaced with new texture that blends with the rest of the image. The result is then propagated to the next frames of the video. One of the important stages of this technique is to update the target region with respect to the illumination change. This is a complex and recurrent problem when the viewpoint changes. We show that the state-of-the-art in DR fails in solving this problem, even under simple scenarios. We then use local illumination models to address this problem. According to these models, the variation in illumination only affects the specular component of the image. In the context of DR, the problem is therefore solved by propagating the specularities in the target area. We list a set of structural properties of specularities which we incorporate in two new models for specularity propagation. Our first model includes the same property as the previous approaches, which is the smoothness of illumination variation, but has a different estimation method based on the Thin-Plate Spline. Our second model incorporates more properties of the specularity's shape on planar surfaces. Experimental results on synthetic and real data show that our strategy substantially improves the rendering quality compared to the state-of-the-art in DR.

Index Terms—Diminished Reality, Specularity, Propagation, Rendering, Isocontours, Brightest point, Illumination variation.

1 Introduction

The aim of Diminished Reality (DR) is to delete a selected object from a video stream in real time [12,17, 18,19,21,26,27,28,36,37]. This technique may be used in many applications. For example, some pieces of furniture may be removed to simulate different arrangements in a room [32, 39]. In live streams, advertising signs can be removed or replaced by new ones depending on the viewer. In Augmented Reality (AR) applications, markers are often used and they can be hidden to achieve seamless fusion between virtual objects and the real world [23,25]. In DR, the user-experience is largely enhanced by a realistic rendering quality. The state-of-the-art image completion methods [2,15,20] allow a coherent replacement of the deleted region which blends perfectly with the rest of the image, even for textured surfaces. However, for temporal consistency in the video and for a real-time application, one cannot apply these methods in each frame. A solution is to use these methods on a selected image which we call the "keyframe". For the next frames of the video stream, one simply needs to copy the inpainting result, considering the camera movement as well as the illumination change around the target area. Here, we address the illumination change problem, which is a crucial stage in any DR pipeline. This is a difficult problem due to the complex nature of light reflections in the presence of glossy

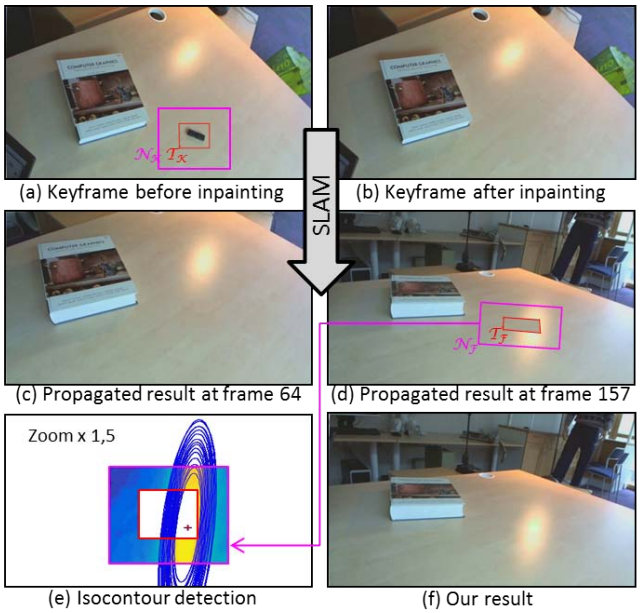


Fig. 1: Specularity propagation in DR. In (a), at frame 1 (the keyframe), the user selects a target region $\mathcal{T}_K$. In (b), we copy patches from the rest of the image to obtain the inpainted keyframe. In (c), the result is transformed using SLAM to frame 64 without specularity propagation. In (d), the transformed result at frame 157 is not visually convincing because a specularity is around $\mathcal{T}_K$. In (e), at frame 157, the image is rectified to the keyframe's image plane and the isocontours of light intensity are fitted with ellipses. This information is used by our model, whose output is shown in (f), to synthesize the specularity.

- *S. Hadj Said is with CEA, LIST, Gif-sur-Yvette, France and Institut Pascal, Clermont-Ferrand, France. E-mail: souheil.hadj-said@cea.fr,*
- *M. Tamaazousti is with CEA, LIST. E-mail: mohamed.tamaazousti@cea.fr*
- *A. Bartoli is with Institut Pascal. E-mail: adrien.bartoli@gmail.com*

surfaces. In fact, illumination variation is often observed when the viewpoint changes. Only two of the previous DR methods explicitly address this problem [18, 22]. We show that even in simple scenarios, with planar surfaces and a single point light source, these methods produce unconvincing results (see figure 4). This is because they only assume the continuity of the illumination variation. In this paper, we analyze the origin of this problem and show that it originates from the specular component in the image. Actually, local illumination models such as Phong's [31], Blinn-Phong's [6] and Cook-Torrance's [9] confirm that only the specular component of light reflection depends on the viewpoint. Moreover, it was shown that the specularities play a key role in scene perception by the human brain [5]. Therefore, we formulate this problem into a specularity propagation problem. More specifically, we consider the case of deleting an object lying on a planar specular surface illuminated by a point light source. From a set of real videos under these assumptions, we observe the structural properties of specularities and propose two new models:

- Our first model is called Smooth Propagation Model (SPM). It is generic and exploits the continuity and smoothness of light intensity. We use the *Thin-Plate Spline* (TPS) as a smooth function representation. Intensity's smoothness was exploited by Kim *et al.* [24] to separate the specular and diffuse components in a single image. SPM achieves state-of-the-art performances and works for general case scenarios, but incorporates few structural properties, and has similarities to previous work [18, 22].
- Our second model is called Constrained Propagation Model (CPM). It incorporates the observed structural properties of the specularity. It extends the first model by imposing additional structural constraints: the ellipticity of the intensity isocontours and the existence of a unique maximum intensity within a specularity. We refer to this as the *brightest point*. CPM is more specific to our assumptions but gives better results than previous methods.

Section 2 describes the main structural properties of a specularity. Section 3 formally states the problem we aim to solve. Section 4 reviews previous solutions. Section 5 introduces our proposed models and algorithms. Finally, section 6 shows and discusses our experimental results.

2 The Structural Properties of a Specularity

By observing images of specularities on planar surfaces such as the ones in figure 2, we established some structural properties of a specularity. These are described in terms of how the light intensity behaves across a specularity:

1) **Smoothness.** The light variation is smooth, and thus continuous.

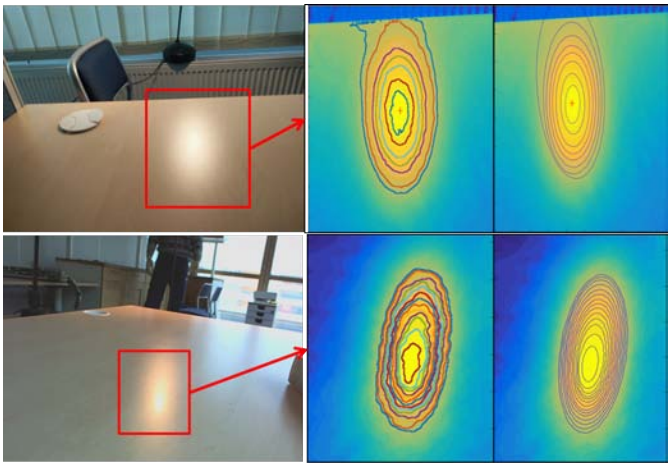


Fig. 2: In (a) the images show specular highlights on a flat surface. In (b) we show the corresponding light map around the specularity for each image and the intensity's isocontours. In (c), we show the fitted ellipses for these isocontours.

2) **Brightest point.** The specularity has a single brightest point located approximately at its center.
3) **Ellipticity.** The isocontours of a specularity are approximately elliptic.
4) **Monotonicity.** The further away from the brightest point, the lower the intensity. This implies that the isocontours do not intersect.
5) **Additivity.** Following the local illumination models, the specular component is a term added to the ambient and diffuse terms.

Some of these properties were theoretically and empirically verified on models from Computer Graphics (specifically Phong's [31] and Blinn-Phong's [6]). In particular, it has been empirically verified in [30] that the elliptic shape is a good approximation for the specularity's isocontours in practice. The fifth property is directly deduced from the Phong illumination model, which suggests that the color intensity I at a given point is expressed as the sum of three components:

$$I = I_{\text{ambient}} + I_{\text{diffuse}} + I_{\text{specular}}. \quad (1)$$

These properties have not been considered for propagating specularities in DR in existing methods [18, 22]. Our goal is to exploit them in order to improve the realism of specularity rendering in DR.

3 Background and Problem Statement

3.1 Notation

Scalars are in italics (*e.g.* x), vectors in bold upright (*e.g.* $\mathbf{v}$) and matrices in sans-serif (*e.g.* M). The elements of a vector are written as in $\mathbf{a}^\top = (a_1\ a_2\ a_3)$ where $^\top$ is vector and matrix transpose. The coordinates of a point in the image are written with a 2-vector $\mathbf{q}^\top = (x\ y)$. An image

domain is written in uppercase calligraphic (*e.g.* $\mathcal{R}$). A group of points is written with uppercase italic (*e.g.* B) and the number of points in a group as $|B|$. Functions are written in upright Greek letters (e.g. ψ) or Latin lower case in italics. The Euclidean distance between two pixels $\mathbf{p}$ and $\mathbf{q}$ is denoted $d(\mathbf{p}, \mathbf{q})$.

3.2 Problem Statement

3.2.1 Context

In this section, we introduce two major techniques used for DR. First, we explain the image inpainting technique. An image can be mathematically defined by a function χ giving the color intensities as:

$$\chi : \left\| \begin{array}{l} \mathcal{O} \subset \mathbb{R}^2 \rightarrow \mathbb{R}^n \\ \mathbf{p} \rightarrow \chi(\mathbf{p}), \end{array} \right. \quad (2)$$

where $\mathbf{p}$ represents a vector indicating the spatial coordinates of a pixel. For the RGB color space ($n = 3$), the image is described by three color intensity functions. So, χ can be written as $\chi^\top = (\chi^R \ \chi^G \ \chi^B)$. Image inpainting [14] was introduced as a term by Bertalmio *et al.* [4]. Since then, many real-time image inpainting techniques were proposed [1, 2, 3, 8, 10, 16, 20, 35]. In general, in the inpainting problem, the image described by χ (*i.e.* corresponding to each color channel of the image) is assumed to have gone through a degradation operation. As a result, the generic definition domain $\mathcal{O}$ of the input image χ can be seen as composed of two parts $\mathcal{O} = \mathcal{S} \cup \mathcal{T}$, $\mathcal{S}$ being the intact part of the image (the source region) and $\mathcal{T}$ the deleted part of the image which we search to recover (the target region). The goal of inpainting is to estimate the color intensities of the pixels $\mathbf{p}$ located in the target region $\mathcal{T}$. As a final result, this technique reconstructs the inpainted image described by $\hat{\chi}$. The objective in terms of quality is that the recovered region looks natural to the human eye, and is as physically plausible as possible. Typical inpainting artifacts are unconnected edges, blur and inconsistent pieces of texture (also called texture garbage).

The second technique we use is Simultaneous Localization and Mapping (SLAM). In our work, it is used to localize the camera and therefore, map the target region in all the frames of the video stream. We denote a 3D point as $\mathbf{x} \in \mathbb{R}^3$, the rotation of the camera as $\mathsf{R} \in \mathbb{SO}(3)$ and its translation as $\mathbf{t} \in \mathbb{R}^3$. At each frame f, SLAM determines the coefficients of R and $\mathbf{t}$ that coherently project a 3D point $\mathbf{x}$ to the camera's image plane. SLAM solves this problem in real time, and is available in mature software packages. We use the SLAM technique from [34].

3.2.2 DR as Spatio-Temporal Inpainting

We consider DR as a spatio-temporal inpainting problem. For the keyframe, spatial consistency is ensured by the inpainting technique. For the next frames of the video, SLAM propagates the spatially-consistent inpainting result while ensuring temporal consistency in the video. However, a specularity may appear around $\mathcal{T}$, which causes illumination variations. So, the spatial structure of the inpainting result should be properly modified in order to achieve spatio-temporal consistency. This modification is essential to obtain a realistic rendering result. We refer to the keyframe image by χ_K and the current video frame by χ_F. An inpainting technique is applied on χ_K to reconstruct the target region $\mathcal{T}_K$. We use a modified version of PatchMatch which is a real-time capable image inpainting approach initially proposed by Barnes *et al.* [2]. The inpainted image is then propagated to the next frames. We use SLAM to transform the current frame to the keyframe image plane. We therefore have dense pixel-wise correspondences between the target region in the keyframe and the one in the current frame. In other words, for each pixel $\mathbf{p}_K$ in the keyframe, we have a corresponding pixel $\mathbf{p}_F = \eta(\mathbf{p}_K)$, η being a homography function. So, we can transform the result of inpainting to all the frames of the video stream. We define the neighboring region $\mathcal{N}_K \subset \mathcal{S}$ centered around $\mathcal{T}_K$ with width $w_\mathcal{N} = z\ w_\mathcal{T}$ and height $h_\mathcal{N} = z\ h_\mathcal{T}$ (see figure 3). $\mathcal{N}$ is the set of neighboring pixels that are outside $\mathcal{T}$. $w_\mathcal{T}$ and $h_\mathcal{T}$ are, respectively, the width and the height of the target region selected by the user. $z > 1$ is set manually depending on the specularity's size to allow for an efficient observation of the specularity's isocontours. The larger the specularity, the greater z. In our experiments, we set $z = 2$. By observing the light variation in the current frame in $\mathcal{N}_F$, we aim to propagate this variation inside $\mathcal{T}_F$. For each pixel $\mathbf{p}_F$ in $\mathcal{N}_F$, the illumination variation coefficient is defined as:

$$v_{\mathbf{p}_K} = \chi_F(\mathbf{p}_F) - \chi_K(\mathbf{p}_K), \mathbf{p}_F \in \mathcal{N}_F. \quad (3)$$

For each frame f, knowing the variation coefficients of the pixels in $\mathcal{N}_F$, we aim to estimate the function ψ_F that returns the illumination variation for all pixels in $\mathcal{T}_F \cup \mathcal{N}_F$ and therefore, update their color intensities as:

$$\hat{\chi}_F(\mathbf{p}_F) = \psi_F(\mathbf{p}_K) + \hat{\chi}_K(\mathbf{p}_K),\ \mathbf{p}_F \in \mathcal{T}_F \cup \mathcal{N}_F. \quad (4)$$

$\psi_F(\mathbf{p}_K)$ can be seen as the estimated value of $v_{\mathbf{p}_F}$ if $\mathbf{p}_F \in \mathcal{T}_F$ and the real value of $v_{\mathbf{p}_F}$ if $\mathbf{p}_F \in \mathcal{N}_F$.

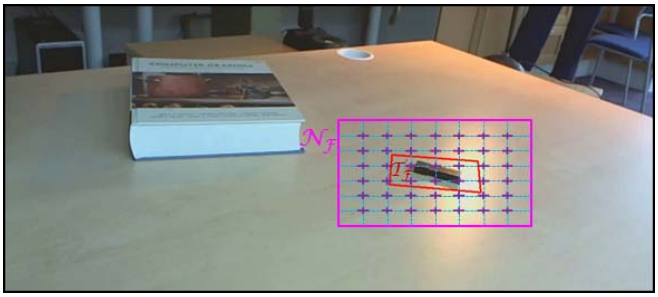
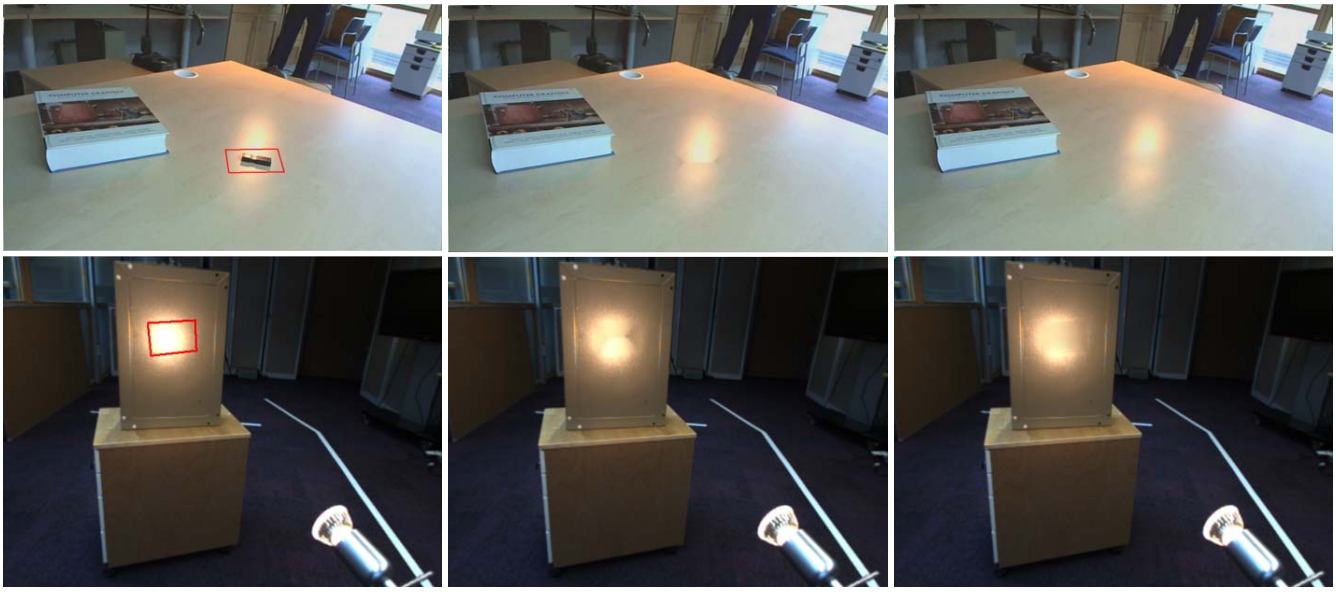


Fig. 3: Representing the region of interest in the current frame $\mathcal{F}$. The neighboring area $\mathcal{N}_\mathcal{F}$ is represented by the blue grid. The purple crosses represent the centers whose target values will be estimated from equation (12). The pixels inside the target region $\mathcal{T}_\mathcal{F}$ (delimited by a red contour) are then interpolated using the TPS.



(a) Original image (b) Herling *et al.* (TVCG 2014) [18] (c) Kawai *et al.* (TVCG 2015) [22]

Fig. 4: Illustration of the limitations of previous methods. The images in column (a) represent the original image and the target area (outlined in red). The results of DR by the methods of Herling *et al.* [18] and Kawai *et al.* [22] are respectively shown in columns (b) and (c). These results are not visually convincing in both cases.

4 State-of-the-Art

The literature has some real-time DR approaches. They use approximately the same pipeline as ours but with different image inpainting and camera tracking techniques [25, 26, 32]. However, only two of them consider the light change problem [18, 22]. They propose heuristic interpolation techniques to estimate the illumination variation in the target region. They use similar models, which suggest that the variation is continuous and smooth, and thus respect the first structural property of section 2. They however use different estimation approaches, explained in the next two sections.

4.1 Herling *et al.*

In the approach of Herling *et al.* [18], pixels from the boundary separating the target region from the rest of the image are monitored over time and the color difference at each of these pixels is computed. Then, a virtual grid $\mathcal{G}$ is defined, covering the target region. For each node of the grid inside the target region $\mathbf{c}$, the color correction function is determined by:

$$\psi_{\text{Herling}} : \left\| \begin{array}{l} \mathcal{G} \subset \mathcal{T}_F \rightarrow \mathbb{R} \\ \mathbf{c} \rightarrow \psi_{\text{Herling}}(\mathbf{c}), \end{array} \right. \tag{5}$$

where:

$$\psi_{\text{Herling}}(\mathbf{c}) = \frac{1}{\theta(\mathbf{c})} \sum_{j=1}^{|B_K|} (\chi_K(\mathbf{b}_{K,j}) - \chi_F(\mathbf{b}_{F,j})) e^{-\,|\mathbf{c}-\mathbf{b}_{K,j}|^{\frac{1}{2}}}, \tag{6}$$

with B_K representing the boundary contour in the keyframe containing the points $\mathbf{b}_{K,1}, \ldots, \mathbf{b}_{K,|B_K|}$ and B_F being the corresponding boundary contour in the current frame containing the corresponding points $\mathbf{b}_{F,1}, \cdots, \mathbf{b}_{F,|B_F|}$. $\theta(\mathbf{c})$ is a normalization factor defined as follows:

$$\theta(\mathbf{c}) = \sum_{j=1}^{|B_K|} e^{-\,|\mathbf{c}-\mathbf{b}_{K,j}|^{\frac{1}{2}}}. \tag{7}$$

Each pixel $\mathbf{p}$ of the target region is then corrected by a bi-linear interpolation considering the coefficients of the four closest grid nodes.

4.2 Kawai *et al.*

Kawai *et al.* [22] analyse the neighboring area to estimate the variation of illumination in the target region. A grid $\mathcal{G}$ is defined on $\mathcal{N}_F \cup \mathcal{T}_F$ where each node is placed in the center of a patch. Initially, they assign the mean illumination variation of each patch in $\mathcal{N}_F$ to its corresponding node. The illumination variation in a pixel $\mathbf{p} \in \mathcal{N}_F$ is computed as described in the problem statement in equation (3). Then, for the target region, the illumination variation of each grid node is computed separately under the assumption that the change in brightness between two adjacent nodes is minimal. We define the function:

$$\psi_{\text{Kawai}} : \left\| \begin{array}{l} \mathcal{G} \subset \mathcal{N}_F \cup \mathcal{T}_F \rightarrow \mathbb{R} \\ \mathbf{c} \rightarrow \psi_{\text{Kawai}}(\mathbf{c}). \end{array} \right. \tag{8}$$

This function is obtained by minimizing the following global cost:

$$\min_{\psi_{\text{Kawai}}} \sum_{(\mathbf{c}_i, \mathbf{c}_j) \in P} (\psi_{\text{Kawai}}(\mathbf{c}_i) - \psi_{\text{Kawai}}(\mathbf{c}_j))^2, \tag{9}$$

with, for $\mathbf{c}_i \in \mathcal{N}_F$:

$$\psi_{\text{Kawai}}(\mathbf{c}_i) = \frac{1}{|G_i|} \sum_{\mathbf{p} \in G_i} v_{\mathbf{p}}, \tag{10}$$

with $\mathbf{c}_i$ and $\mathbf{c}_j$ being the centers of two adjacent patches of the grid. P is a set of pairs of adjacent patches. G_i is the group of pixels in the patch centered around $\mathbf{c}_i$. Minimizing this cost allows one to retrieve the values of the grid nodes inside the target region $\mathcal{T}_F$. The coefficient of color variation $\psi_{\text{Kawai}}(\mathbf{p})$ for each pixel $\mathbf{p}$ inside the mask is then deduced by bi-linear interpolation.

4.3 Discussion

The two methods [18, 22] propose models that handle global image-level light changes well. However, only the smoothness property is considered by Kawai *et al.* (Property 1 in section 2). Herling *et al.*'s method [18] also respects the additivity property (property 5 in section 2). In other words, [22] uses a multiplicative model to update the illumination variation, [18] uses an additive model which is coherent with local illumination models. However, this is still insufficient in cases with specular surfaces and artificial lighting condition. Examples of DR show the limits of these methods in figure 4. Those demonstrate that even in basic scenarios including a specular planar surface under a point light source, state-of-the-art methods do not provide satisfying solutions.

5 Proposed Models and Methods

5.1 The Thin-Plate Spline

We propose two models based on our observations of the specularity's structural properties. Both models use the Thin-Plate Spline (TPS) as an interpolation function. It is a very suitable tool in this context because it enforces the smoothness constraint. Here, we briefly introduce the parameterization of the TPS. As inputs, we consider a set of l centers $\mathbf{c}_k \rightarrow u_k$ where $\mathbf{c}_k \in \mathbb{R}^2$ holds the coordinates of a center and $u_k \in \mathbb{R}$ is its corresponding unknown target value. We define the centers' coordinate matrix $\mathsf{C} = (\mathbf{c}_1 \cdots \mathbf{c}_l)$ and the centers' target vector $\mathbf{u}^\top = (u_1 \cdots u_l)$. The correspondence $(\mathbf{c}_1 \cdots \mathbf{c}_l) \rightarrow (u_1 \cdots u_l)$ represents the control points for the TPS. The TPS is a smooth function from $\mathbb{R}^2$ to $\mathbb{R}$ driven by these centers and given for any point $\mathbf{p} \in \mathbb{R}^2$ by:

$$\phi_{\text{tps}}(\mathbf{p}; \mathbf{u}) = \mathbf{l}_\mathbf{p}^\top \, \mathsf{E}_\lambda \, \mathbf{u} \, , \tag{11}$$

where $\mathbf{l}_\mathbf{p}^\top = \Big(\rho\big(d^2(\mathbf{c}_1, \mathbf{p})\big) \cdots \rho\big(d^2(\mathbf{c}_l, \mathbf{p})\big)\Big)$ with $\rho(d) = d \log(d)$ being the TPS kernel for the squared distance. E_λ is the feature-driven parameterization matrix which incorporates an internal regularization weight $\lambda \in \mathbb{R}^+$ [7, 11]. λ controls the sensitivity of the interpolation function to fine variations. We set it to a small value for small-size specularities and a larger value for large-size specularities (λ can be set from 10^{-3} and up to 2.10^{-1}).

In practice, we have arbitrary positioned centers with unknown target values $\mathbf{u}$. So, given a set of m data points $\mathbf{q}_i \rightarrow v_i$, we estimate the optimal target values by solving:

$$\min_{\mathbf{u}} \sum_{i=1}^{m} (\phi_{\text{tps}}(\mathbf{q}_i; \mathbf{u}) - v_i)^2. \tag{12}$$

This forms a linear least squares problem, which we solve with a simple matrix pseudo-inverse.

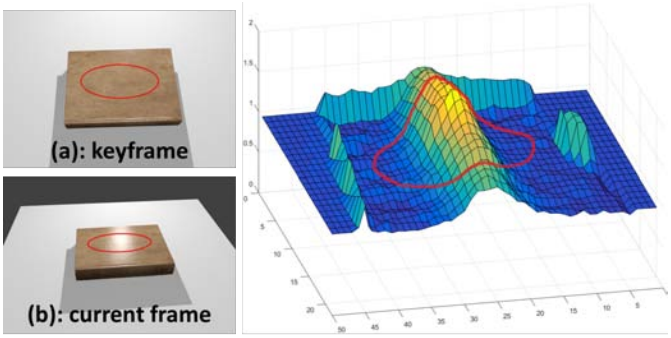


Fig. 5: The illumination variation in the image plane between the keyframe K and the current frame F is viewed as an elevation map. In this example, a specularity crosses the target region $\mathcal{T}_F$. This demonstrates the smoothness and continuity properties of a specularity.

5.2 Smooth Propagation Model

5.2.1 Description

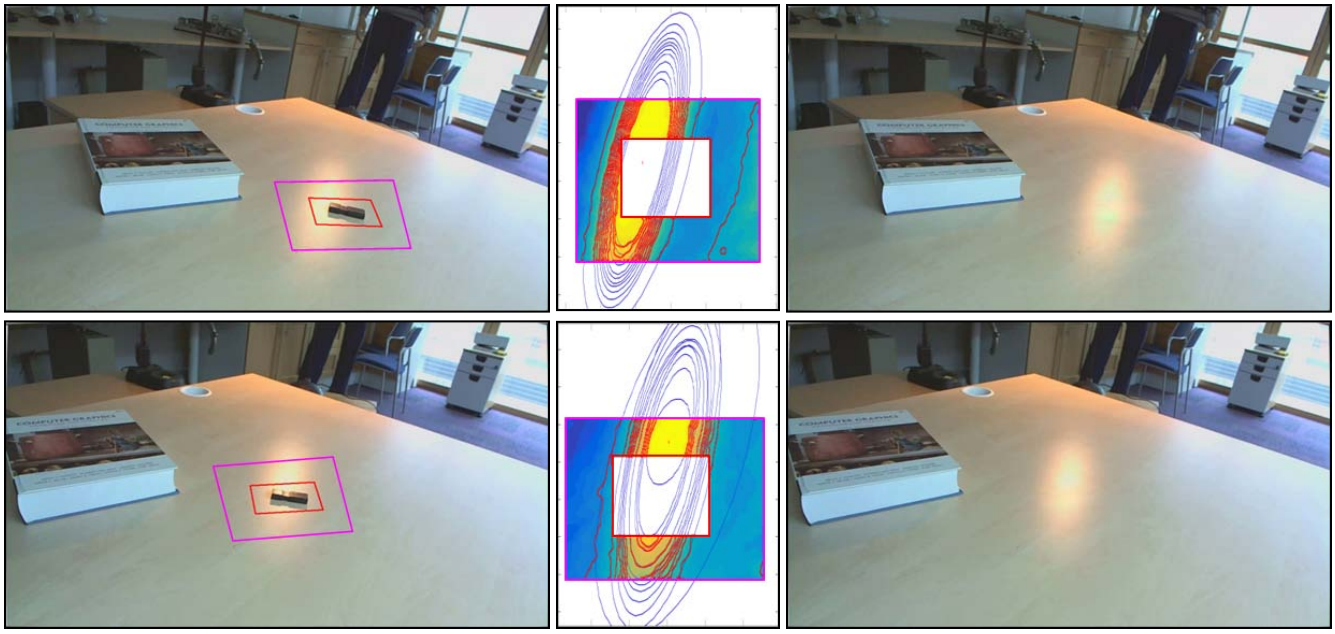
The illumination variation can be viewed as a time-varying elevation map, as shown in figure 5. The base represents the pixel coordinates in the image and the height gives the variation's value. We propose a first model that only incorporates the smoothness property. We model the illumination variation by a TPS. This model makes few assumptions on the scene so it works for general case scenarios. However, it may generate poor results in some cases. We call it Smooth Propagation Model (SPM).

5.2.2 Estimation

We use the TPS to represent the function ψ_F that returns the illumination variation, for all pixels in $\mathcal{N}_F \cup \mathcal{T}_F$:

$$\psi_F : \left\| \begin{array}{l} \mathcal{O} \subset \mathbb{R}^2 \rightarrow \mathbb{R} \\ \mathbf{p} \rightarrow \begin{array}{l} v_{\mathbf{p}_K} = \chi_F(\mathbf{p}_F) - \chi_K(\mathbf{p}_K), \text{ if } \mathbf{p}_F \in \mathcal{N}_F \\ \phi_{\text{tps}}(\mathbf{p}_K; \mathbf{u}), \text{ if } \mathbf{p}_F = \eta_F(\mathbf{p}_K) \in \mathcal{T}_F \end{array} \end{array} \right. \tag{13}$$

We consider a uniformly distributed grid $\mathcal{G}_\mathcal{F} \subset \mathcal{N}_\mathcal{F} \cup \mathcal{T}_\mathcal{F}$. We set the grid so as to have l nodes, with l a perfect square. Using the parameterization of the TPS introduced in section 5.1, we consider the grid nodes as the centers $\mathbf{c}_k$ and the pixel intensity variations between the keyframe and the current frame as the target values u_k. The points in $\mathcal{N}_F$ are considered as the data points used to estimate $\mathbf{u}$. Using the estimation method from section 5.1, we obtain the TPS function ϕ_{tps}. The number of centers is chosen as $l = 100$, and the number of data points m depends on how many pixels we have in the neighboring region. In terms of computation, this method requires a least squares fit at every frame to solve (12). However, the matrix E_λ is constant, meaning that it can be precomputed from the keyframe only. In other words, solving for ψ_F requires solving minimization (12) with a simple multiplication between a constant matrix and the measured vector of variations $\mathbf{v}^\top = (v_1 \cdots v_m)$. In the RGB color space, we need to estimate separately three intensity differences for each pixel $\mathbf{p}_F$, E_λ being the same for the three color



(a) Original images (b) Ellipse fitting (c) Results of model 2

Fig. 6: Isocontour estimation on the brightness map. In (b), we use red represent the points used to estimate the ellipses and blue for the final estimation. The interpolated ellipses may extend inside the target region, and this allows us to propagate the specularity with high accuracy.

channels. A TPS was already used in [33] to model image-based light changes in the context of registration.

5.3 Constrained Propagation Model

5.3.1 Description

We extend SPM by considering more structural properties of specularities in the case of a scene with a single point light source. In this case, we constrain the new model by all the five properties from section 2. We call the second model Constrained Propagation Model (CPM).

5.3.2 Estimation

We integrate the constraint of the elliptical isocontours (property 3) by fixing a number s of intensity levels in the specular highlight. An isocontour is a set of pixels with the same intensity level (see figure 2 (b)). For each isocountour with intensity level h, we estimate the ellipse E by solving:

$$\min_{\mathbf{e}} \sum_{j=1}^{r} \Big(h - \chi^L(E(j))\Big)^2, \tag{14}$$

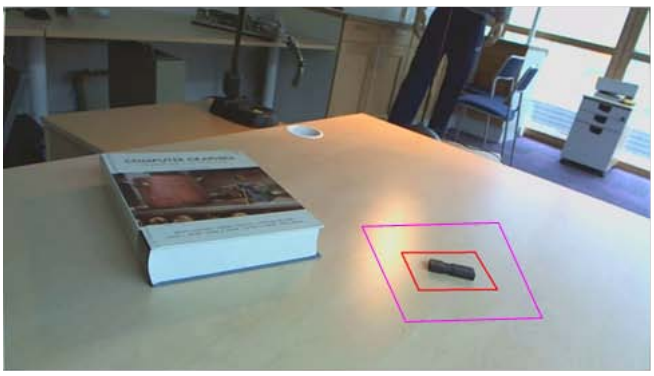
where χ^L returns the L color intensity values of a pixel in the Lab color space. The ellipse is represented by its five natural parameters $\mathbf{e}^\top = (o_x \ o_y \ a \ b \ w) \in \mathbb{R}^5$ with o_x and o_y as the center's coordinates, a as the semi-major axis, b as the semi-minor axis and w as the angle orienting the major axis. The ellipse is discretized in a group of points E of size $r = 100$ to evaluate the cost in (14), with $E(j) \in \mathbb{R}^2$ the j-th element in E. Further details on the fitting algorithm are given in section 5.3.2.2. The number of intensity levels s also represents the number of iso-contours considered, corresponding to the levels $h_{min} \cdots h_{max}$. Since the maximum intensity level h_{max} is constant, s is automatically adjusted depending on the minimum intensity level h_{min}. This threshold is fixed manually depending on the light exposure of the camera and light intensity. In particular, the brighter the light reflection, the higher the value of h_{min}.

5.3.2.1 Isocontour Detection

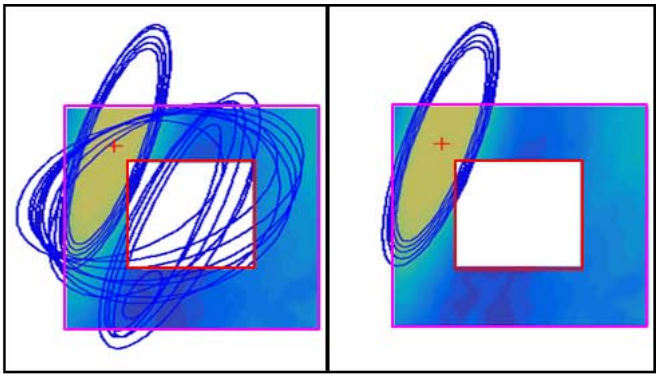
To evaluate the illumination variation, we convert the image to the Lab color space and consider only the L channel (Lightness). To reduce the computation cost, we only search for isocontours where a specularity is detected. To do so, we use a real-time algorithm for detecting specular reflections inspired from the methods in [24, 29]. To properly detect isocontours of intensity levels as ellipses, we begin by applying the Wiener filter [38] to segment the Lightness levels and reduce the noise generated by the roughness in the surface. Then we use a quantification histogram to segment the image into light intensity levels. The result is the brightness map. The detection of isocontours is carried out in this map. For a light level h, a point from the isocontour is detected when its corresponding intensity level is h and one of its neighboring points has an intensity level of h-1. Accordingly, we define s corresponding levels of intensity and detect their isocontour points.

5.3.2.2 Ellipse Fitting

Considering the brightness map obtained in the neighboring region $\mathcal{N}_F$, as shown in figure 6. The isocontour points



(a) Original image



(b) Before filtering (c) After filtering

(d) Result before filtering

(e) Result after filtering

Fig. 7: Comparing the ellipse fitting results on the same video frame before and after filtering out the inconsistent ellipses. In (b), before filtering, many false estimations of ellipses are seen. This generates visual artifacts on the final result seen in (d). In (c), we were able to filter the false estimated ellipses according to their confidence coefficients as described in section 5.4.2. In (e), the final result of CPM is visually more convincing than the one in (d). The brightness maps in (b) and (c) are rectified to the keyframe image plane and zoomed for better visualization.

that are outside $\mathcal{T}_F$ may be interpolated into ellipses extending within the target region. Our goal here is to estimate the extension of these isocontours assuming they have an elliptic shape. This is equivalent to minimizing the criterion in equation (14). To do so, we use the algorithm of Fitzgibbon *et al.* [13] for feature-based least squares fitting of ellipses. From a set of points from an isocontour, this method attempts to adjust the best ellipse which minimizes the algebraic distance. The result is used as an initialization. The parameters of the ellipse are fed to a simplex direct search algorithm to solve problem (14). This algorithm not only allows us to find all the points of the isocontour that are within the target area, but also the position of the ellipse's center. This information will be exploited to approximate the position of the brightest point. In this model we assume that the position of the brightest point is located at the center of the smallest detected isocontour. For an isocontour C, the difference of light variation for a color channel z is computed for points $\mathbf{p} \in C \cap \mathcal{T}_F$ as:

$$v^z_{\mathbf{p}} = \max_{\mathbf{q} \in C \cap \mathcal{N}_F} v^z_{\mathbf{q}}. \tag{15}$$

with C the set of points in the real isocontour and $v^z_{\mathbf{p}}$ the difference in intensity between $\mathcal{K}$ and $\mathcal{F}$ at pixel $\mathbf{p}$.

5.3.2.3 Consistency Filtering

In practice, due to estimation errors, there may be overlapping ellipses (see figure 7). So, we constrain the estimation of the ellipses by imposing property 4 from section 2, which is the monotonic decrease in intensity of the specular highlights. Based on our formulation of the problem, this property translates to the fact that *each ellipse of intensity level h_e should be totally inside any ellipse of intensity level $h < h_e$*. In order to respect this condition, we define a confidence coefficient for each estimated ellipse. The confidence coefficient for an estimated ellipse of intensity level h_e is determined with the function γ defined as:

$$\gamma(e) = \sum_{j=1, j \neq e}^{s} \delta(j, e), \tag{16}$$

where:

$$\delta(j,e) = \begin{cases} 1(E_j \subset E_e) \text{ if } h_j > h_e \\ 1(E_e \subset E_j) \text{ if } h_j < h_e \\ 0 \text{ otherwise.} \end{cases} \tag{17}$$

After the initialization with the feature-based fitting of ellipses, we compute the confidence coefficients of the different estimated contours and we retain the largest set of consistent ellipses (with the maximum confidence values). We refer to this operation by consistency filtering as in figure 7. We then refine the retrieved ellipses by solving problem (14).

5.3.2.4 Incorporating the Constraints to the TPS

Returning to the estimation of the TPS, we add a fixed number of pixel coordinates as data points. They belong to the interpolated ellipses $E_1, \ldots, E_s$ which are within the target region. Their respective target values are the differences in intensity of the corresponding isocontours.

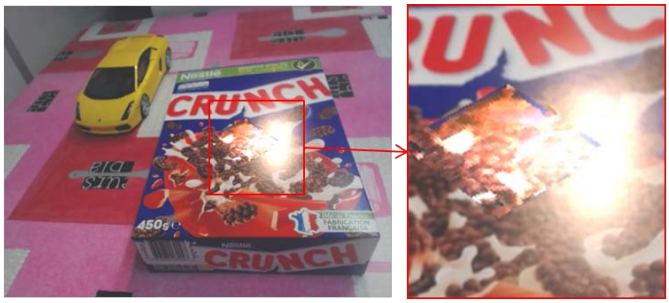
6 Results and Discussion

6.1 Datasets

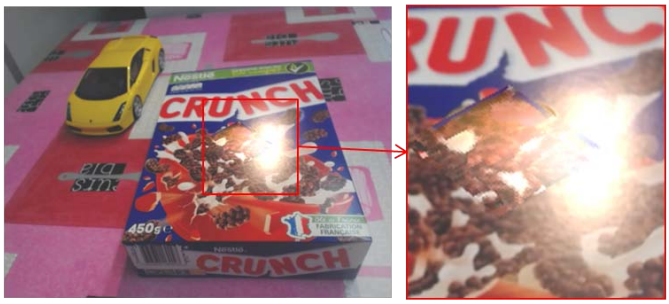
The video sequences used for comparison in this section are divided into two categories. The first category includes two videos: a synthetic one (video 1) which was generated from the rendering software Blender-3D illustrated in figure 10, and a real one (video 2) illustrated in figure 11. The synthetic environment chosen in video 1 allows us to take full control of the different parameters in the scene (the light source's position, its intensity value, the object's material, camera orientation, etc). The reflection model is Phong's. For both videos, no undesired object is placed in the target region in order to let us compare the rendered specularity to the real one. The second category includes two real videos, video 3 (figure 12) and video 4 (figure 13), with undesired objects. These videos show the case of a planar surface crossed by a specularity.

6.2 Comparing Results

We can see that CPM achieves the best results for all videos in terms of specularity propagation. For the first two videos, CPM is the closest to the ground truth video. For the two other videos, its results are more visually coherent than for the other methods. So, this model shows an improvement of the rendering quality compared to SPM and state-of-the-art. This shows the relevance of incorporating the specularity's structural properties. Even though it incorporates only the smoothness property, SPM still achieves better results than state-of-the-art. In fact, this demonstrates that the TPS is a well-adapted interpolation function for this problem. In particular, the method of Kawai *et al.* [22] works well with weak illumination variations when the brightest point is outside the target region. However, it clearly fails when a specularity enters the target region, with high intensity variations or with rich-texture surfaces. The method of Herling *et al.* [18] is very dependent on the boundary's shape. Therefore, the specularity rendering result is not visually convincing. It generates artifacts that seem unnatural. The interpolation of illumination variation in the previous methods does not respect the specularity's structure, which explains these results. Video 4 presents a very difficult case of rich texture surface (strong local variations of colors). In fact, the presence of such texture in the neighboring region along with the presence of the white color results in a non-smooth variation of pixel intensities in the three color channels. Since these pixels can be saturated in one or more of the RGB channels, the mean computed variation is usually insufficient to reproduce the specularity for SPM and previous methods. However, CPM can overcome this issue by imposing additional constraints. The fact that we consider the maximum difference of intensity for the isocontour points allows us to avoid these extreme cases. Although some artifacts can still occur, our model gives the best results by far, compared to previous methods (see figure 13).



(a) Multiplicative model



(b) Additive model

Fig. 8: Comparing the results of our second model CPM using an additive model versus a multiplicative model for computing the illumination variation coefficients.

6.3 Additive Versus Multiplicative Model

As mentioned in section 4, the previous methods express differently the illumination variation coefficients. Herling *et al.* [18] use an additive model while Kawai *et al.* [22] use a multiplicative model. In our proposed methods, our choice was guided by the local illumination models. This choice is also supported by the results on real data. In figure 8, we compare the results of CPM associated for the two different possibilities. This shows that the pixel colors are altered when using a multiplicative model (they become overly bright). This is particularly strong for rich texture surfaces. This is also confirmed by the results of [18] (using the additive model) which are more convincing than the ones of [22] (using the multiplicative model) on video 4.

6.4 Computation Time

In figure 9, we present the computation time for each frame of video 2 using the proposed models as well as the state-of-the-art methods. We ran these tests on an Intel i7 processor with a 2.70 GHz frequency. Video 2 has a frame size of 640x480 pixels. The inpainting step is launched at frame 160, which explains the peak in computation time for all models. SPM has a stable computation time at around 70 ms per frame which corresponds to a performance of 14 frames per second. CPM has a mean computation time of 160 ms per frame which corresponds to a performance of about 7 frames per second. Many optimizations could be done, including the parallelization of the algorithm and an optimized choice of the parameters. However, we can already state that the proposed models are well adapted for real-time applications.

6.5 Discussion and Limitations

Our proposed model SPM imposes the smoothness property of a specularity which is true for any type of specular surfaces. This property is observed on most types of surfaces and it is also imposed by state-of-the-art methods. However, our model uses a more adapted mathematical formalism based on the TPS which generates more plausible results. Our method CPM incorporates further properties based on the specularity's shape. However, the property of the uniqueness of the brightest point is particularly valid for a point light source. Under these circumstances, CPM outperforms by far previous methods as shown on the experimental results. The choice between the first and second models can be decided depending on the specifications of the observed scene. The methods presented in this paper consider a single plane in the target region and therefore compute a single homography corresponding to this plane. Kawai *et al.* [22] propose to consider multiple homographies in order to handle multiple planes. Similarly, our estimation method can also be extended by considering the geometry of the surface if known.

Our models along with state-of-the-art do not handle a rough surface because in this case the property of smoothness is no longer valid. Another specific case, which is not considered by our models as well as state-of-the-art, is when a small specularity fully enters the target region. For this case, a specularity prediction model using dynamics is needed. This implies that the light source position is known or can be estimated using a multiple-view geometric model as the one from [30].

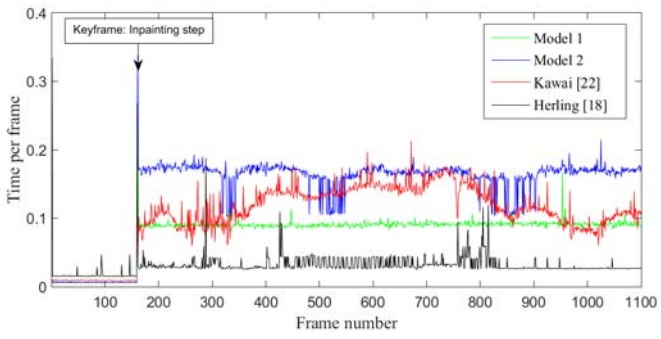


Fig. 9: The computation time per frame in seconds for the proposed models and previous models on video 2.

7 Conclusion

This paper deals with the illumination variation problem in the context of Diminished Reality. This complex problem can be transformed into a specularity propagation problem. From multiple empirical observations, we proposed a list of structural properties of a specularity. We then proposed two models that embed these properties to estimate the illumination variation. Our first model is generic, while our second model is more adapted to curvy surfaces with single point light sources. Our experimental results show the relevance of our approach compared to previous work which do not embed these properties. Particularly, the results of our second model CPM show a substantial improvement in rendering results with respect to the specularity's spatial structure compared to state-of-the-art. In future work, we plan to address other types of surfaces. We will introduce further constraints induced by the specularity's shape depending on the object's material as well as the surface's roughness and geometry. We will also handle multiple planes in the target region as explained in the discussion section.

Acknowledgements

This research has received funding from the EU's FP7 through the ERC research grant 307483 FLEXABLE.

References

[1] Coloma Ballester, Marcelo Bertalmio, Vicent Caselles, Guillermo Sapiro, and Joan Verdera. Filling-in by joint interpolation of vector fields and gray levels. *IEEE transactions on image processing*, 10(8):1200–1211, 2001.

[2] Connelly Barnes, Eli Shechtman, Adam Finkelstein, and Dan Goldman. Patchmatch: A randomized correspondence algorithm for structural image editing. *ACM Transactions on Graphics (TOG)*, 28(3):24, 2009.

[3] Marcelo Bertalmio, Andrea L Bertozzi, and Guillermo Sapiro. Navier-stokes, fluid dynamics, and image and video inpainting. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 1, pages I–355. IEEE, 2001.

[4] Marcelo Bertalmio, Guillermo Sapiro, Vincent Caselles, and Coloma Ballester. Image inpainting. In *ACM Special Interest Group on GRAPHics and Interactive Techniques (SIGGRAPH)*, pages 417–424. ACM, 2000.

[5] Andrew Blake and Heinrich Bülthoff. Does the brain know the physics of specular reflection? *Nature*, 343(6254):165–168, 1990.

[6] James F Blinn. Models of light reflection for computer synthesized pictures. In *ACM Special Interest Group on GRAPHics and Interactive Techniques (SIGGRAPH)*, volume 11, pages 192–198. ACM, 1977.

[7] Fred L Bookstein et al. Principal warps: Thin-plate splines and the decomposition of deformations. *IEEE Transactions on pattern analysis and machine intelligence (PAMI)*, 11(6):567–585, 1989.

[8] Tony F Chan and Jianhong Shen. Nontexture inpainting by curvature-driven diffusions. *Journal of Visual Communication and Image Representation (VCIR)*, 12(4):436–449, 2001.

[9] Robert L Cook and Kenneth E. Torrance. A reflectance model for computer graphics. *ACM Transactions on Graphics (TOG)*, 1(1):7–24, 1982.
[10] Antonio Criminisi, Patrick Perez, and Kentaro Toyama. Object removal by exemplar-based inpainting. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 2, pages II–721. IEEE, 2003.
[11] Jean Duchon. Splines minimizing rotation-invariant semi-norms in sobolev spaces. In *Constructive Theory of Functions of Several Variables*, pages 85–100. Springer, 1977.
[12] Akihito Enomoto and Hideo Saito. Diminished reality using multiple handheld cameras. In *Proc. ACCV*, volume 7, pages 130–135. Citeseer, 2007.
[13] Andrew Fitzgibbon, Maurizio Pilu, and Robert B Fisher. Direct least square fitting of ellipses. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 21(5):476–480, 1999.
[14] Christine Guillemot and Olivier Le Meur. Image inpainting: Overview and recent advances. *IEEE signal processing magazine*, 31(1):127–144, 2014.
[15] Kaiming He and Jian Sun. Statistics of patch offsets for image completion. In *European Conference on Computer Vision (ECCV)*, pages 16–29. Springer, 2012.
[16] Kaiming He and Jian Sun. Image completion approaches using the statistics of similar patches. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 36(12):2423–2435, 2014.
[17] Jan Herling and Wolfgang Broll. Pixmix: A real-time approach to high-quality diminished reality. In *IEEE and ACM International Symposium on Mixed and Augmented Reality (ISMAR)*, pages 141–150. IEEE, 2012.
[18] Jan Herling and Wolfgang Broll. High-quality real-time video inpainting with pixmix. *IEEE Transactions on Visualization and Computer Graphics (TVCG)*, page 1, 2014.
[19] Songkran Jarusirisawad, Takahide Hosokawa, and Hideo Saito. Diminished reality using plane-sweep algorithm with weakly-calibrated cameras. *Progress in Informatics*, 7:11–20, 2010.
[20] Narendra Ahuja Jia-Bin Huang, Sing Bing Kang and Johannes Kopf. Image completion using planar structure guidance. *ACM Special Interest Group on GRAPHics and Interactive Techniques (SIGGRAPH)*, 33(4):to appear, 2014.
[21] Norihiko Kawai, Tomokazu Sato, and Naokazu Yokoya. Diminished reality considering background structures. In *IEEE and ACM International Symposium on Mixed and Augmented Reality (ISMAR)*, pages 259–260. IEEE, 2013.
[22] Norihiko Kawai, Tomokazu Sato, and Naokazu Yokoya. Diminished reality based on image inpainting considering background geometry. *IEEE Transactions on Visualization and Computer Graphics (TVCG)*, 2015.
[23] Norihiko Kawai, Masayoshi Yamasaki, Tomokazu Sato, and Naokazu Yokoya. Ar marker hiding based on image inpainting and reflection of illumination changes. In *IEEE and ACM International Symposium on Mixed and Augmented Reality (ISMAR)*, pages 293–294. IEEE, 2012.
[24] Hyeongwoo Kim, Hailin Jin, Sunil Hadap, and Inso Kweon. Specular reflection separation using dark channel prior. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1460–1467, 2013.
[25] Otto Korkalo, Miika Aittala, and Sanni Siltanen. Light-weight marker hiding for augmented reality. In *IEEE and ACM International Symposium Mixed and Augmented Reality (ISMAR)*, pages 247–248. IEEE, 2010.
[26] Crystian Wendel M Leao, Joao Paulo Lima, Veronica Teichrieb, Eduardo S Albuquerque, and Judith Kelner. Altered reality: Augmenting and diminishing reality in real time. In *IEEE Conference on Virtual Reality (VR)*, pages 219–220. IEEE, 2011.
[27] Vincent Lepetit and Marie-Odile Berger. An intuitive tool for outlining objects in video sequences: Applications to augmented and diminished reality. *IEEE and ACM International Symposium Mixed Reality (ISMR) volume=2, pages=159-60, year=2001*.
[28] Siim Meerits and Hideo Saito. Real-time diminished reality for dynamic scenes. In *IEEE and ACM International Symposium on Mixed and Augmented Reality Workshops (ISMARW)*, pages 53–59. IEEE, 2015.
[29] Alexandre Morgand and Mohamed Tamaazousti. Generic and real-time detection of specular reflections in images. In *International Conference on Computer Vision Theory and Applications (VISAPP)*, volume 1, pages 274–282. IEEE, 2014.
[30] Alexandre Morgand, Mohamed Tamaazousti, and Adrien Bartoli. An empirical model for specularity prediction with application to dynamic retuxturing. In *IEEE and ACM International Symposium Mixed and Augmented Reality (ISMAR)*. IEEE, 2016.
[31] Bui Tuong Phong. Illumination for computer generated pictures. *Communications of the ACM*, 18(6):311–317, 1975.
[32] Sanni Siltanen. Diminished reality for augmented reality interior design. *The Visual Computer*, pages 1–16, 2015.
[33] Geraldo Silveira and Ezio Malis. Unified direct visual tracking of rigid and deformable surfaces under generic illumination changes in grayscale and color images. *International journal of computer vision*, 89(1):84–105, 2010.
[34] Mohamed Tamaazousti, Vincent Gay-Bellile, SN Collette, Steve Bourgeois, and Michel Dhome. Nonlinear refinement of structure from motion reconstruction by taking advantage of a partial knowledge of the environment. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3073–3080. IEEE, 2011.
[35] Alexandru Telea. An image inpainting technique based on the fast marching method. *Journal of graphics tools*, 9(1):23–34, 2004.
[36] Philipp Tiefenbacher, Michael Sirch, and Gerhard Rigoll. Mono camera multi-view diminished reality. In *IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 1–8. IEEE, 2016.
[37] Yonatan Wexler, Eli Shechtman, and Michal Irani. Space-time video completion. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 1, pages I–120. IEEE, 2004.
[38] Norbert Wiener. *Extrapolation, interpolation, and smoothing of stationary time series*, volume 2. MIT press Cambridge, MA, 1949.
[39] Edward Zhang, Michael F Cohen, and Brian Curless. Emptying, refurnishing, and relighting indoor spaces. *ACM Transactions on Graphics (TOG)*, 35(6):174, 2016.

Souheil HADJ SAID is a PhD candidate of the Université Clermont Auvergne, in Clermont-Ferrand, France in partnership with the ENCOV research group at the Institut Pascal and the LVIC laboratory at CEA-LIST under the supervision of Dr. Mohamed Tamaazousti and Pr. Adrien Bartoli. He received his Engineering Degree in computer science from the Tunisia Polytechnic School in La Marsa, Tunisia, in 2013. His research objectives are to improve the rendering of videos in Diminished Reality applications.

Mohamed Tamaazousti received his Master's Degree in applied mathematical from the University of Orleans, France in 2009. He received the Ph.D. degree in computer vision from the University Blaise Pascal, Clermont-Ferrand, France in 2013. He is currently a permanent researcher in the LVIC laboratory at CEA LIST. His main research interests include structure from motion for rigid scenes, real time vision-based localization and reconstruction (SLAM) for autonomous system. He is also interested in augmented and diminished reality applications.

Adrien Bartoli has held the position of Professor of Computer Science at Université Clermont Auvergne since fall 2009 and is a member of Institut Universitaire de France since fall 2016. He leads the Endoscopy and Computer Vision (ENCOV) research group at Institut Pascal, CNRS, UCA and CHU de Clermont-Ferrand. His main research interests include image registration and Shape-from-X for rigid and deformable environments, and their application in computer-aided endoscopy.

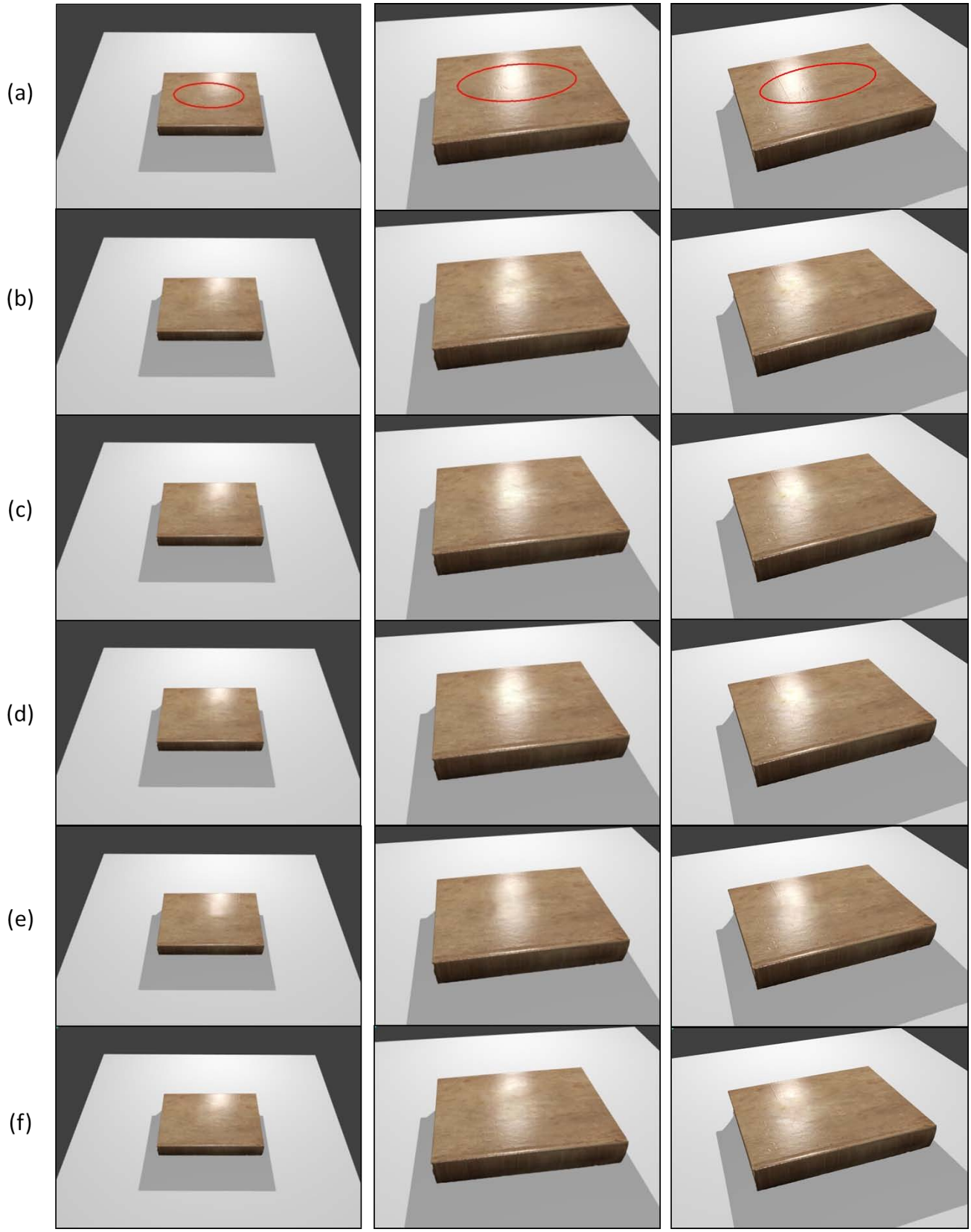


Fig. 10: Results of all the methods on frames 64, 71 and 130 of video 1. (a) corresponds to the original image with the target region in red. (b) SPM. (c) CPM before filtering inconsistent ellipses. (d) CPM after filtering inconsistent ellipses. (e) Method of Herling *et al.* [18]. (f) Method of Kawai *et al.* [22].

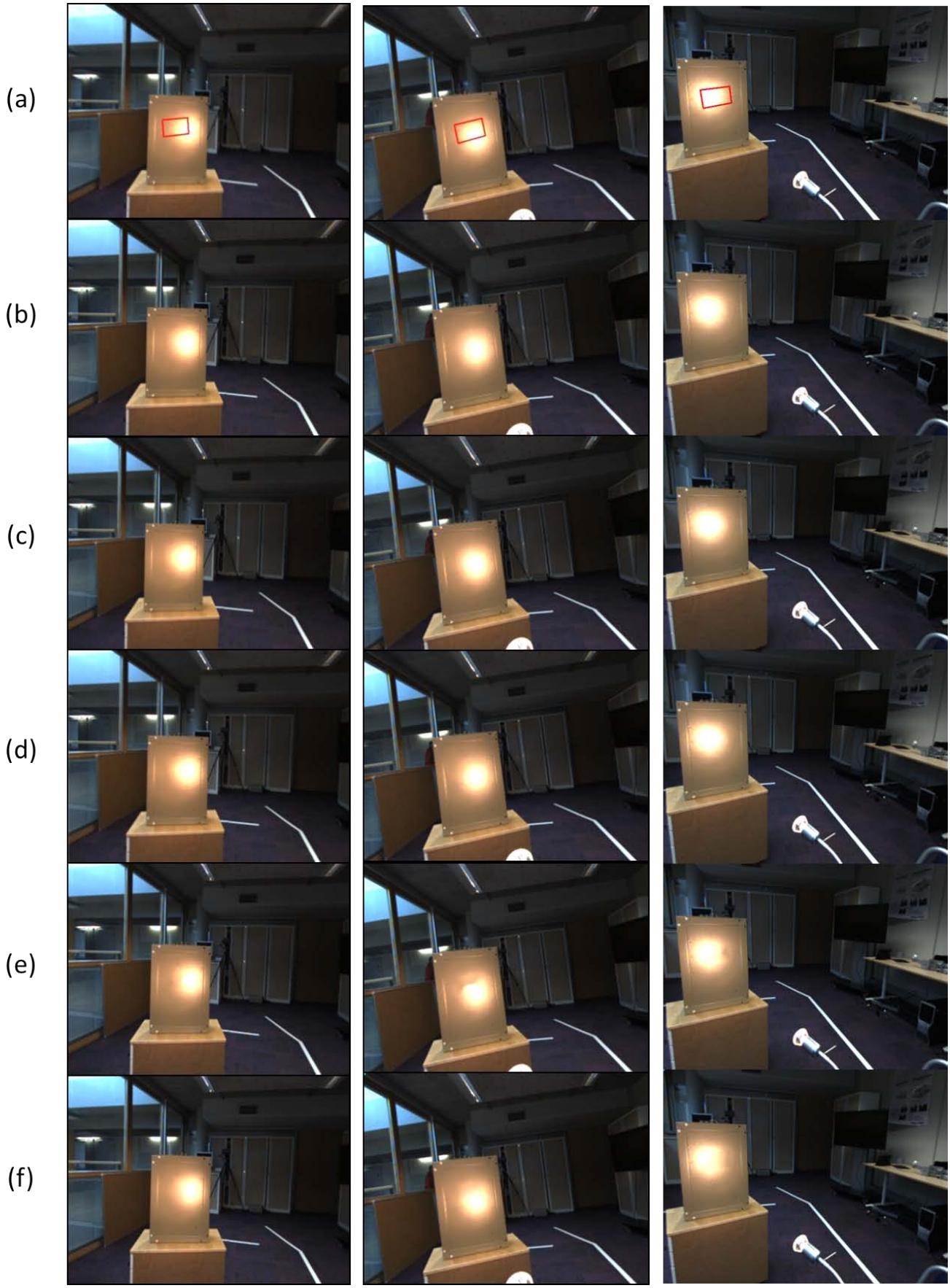


Fig. 11: Results of all the methods on frames 367, 541 and 1227 of video 2. (a) corresponds to the original image with the target region in red. (b) SPM. (c) CPM before filtering inconsistent ellipses. (d) CPM after filtering inconsistent ellipses. (e) Method of Herling *et al.* [18]. (f) Method of Kawai *et al.* [22].

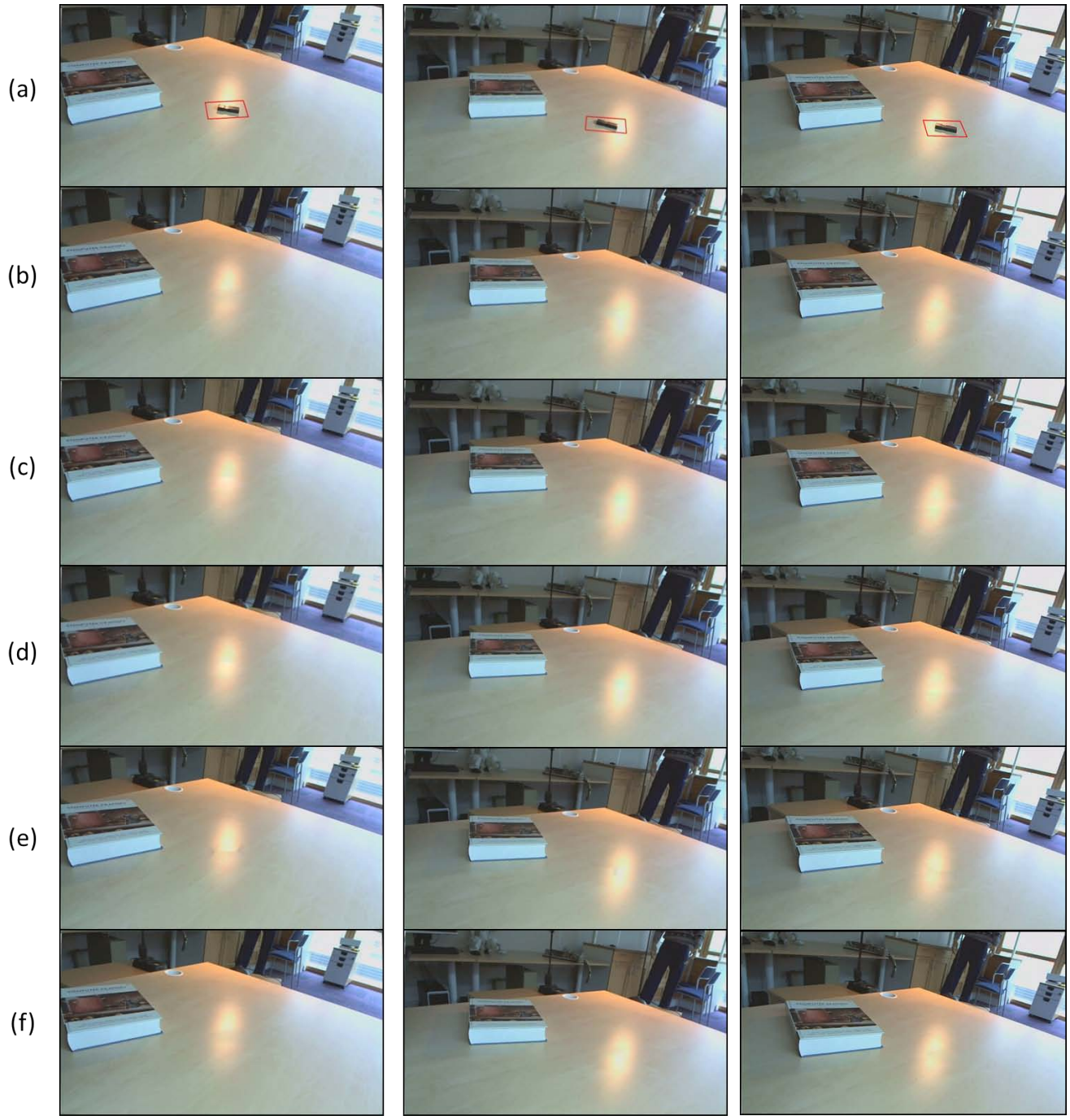


Fig. 12: Results of all the methods on frames 168, 202 and 328 of video 3. (a) corresponds to the original image with the target region in red. (b) SPM. (c) CPM before filtering inconsistent ellipses. (d) CPM after filtering inconsistent ellipses. (e) Method of Herling *et al.* [18]. (f) Method of Kawai *et al.* [22].

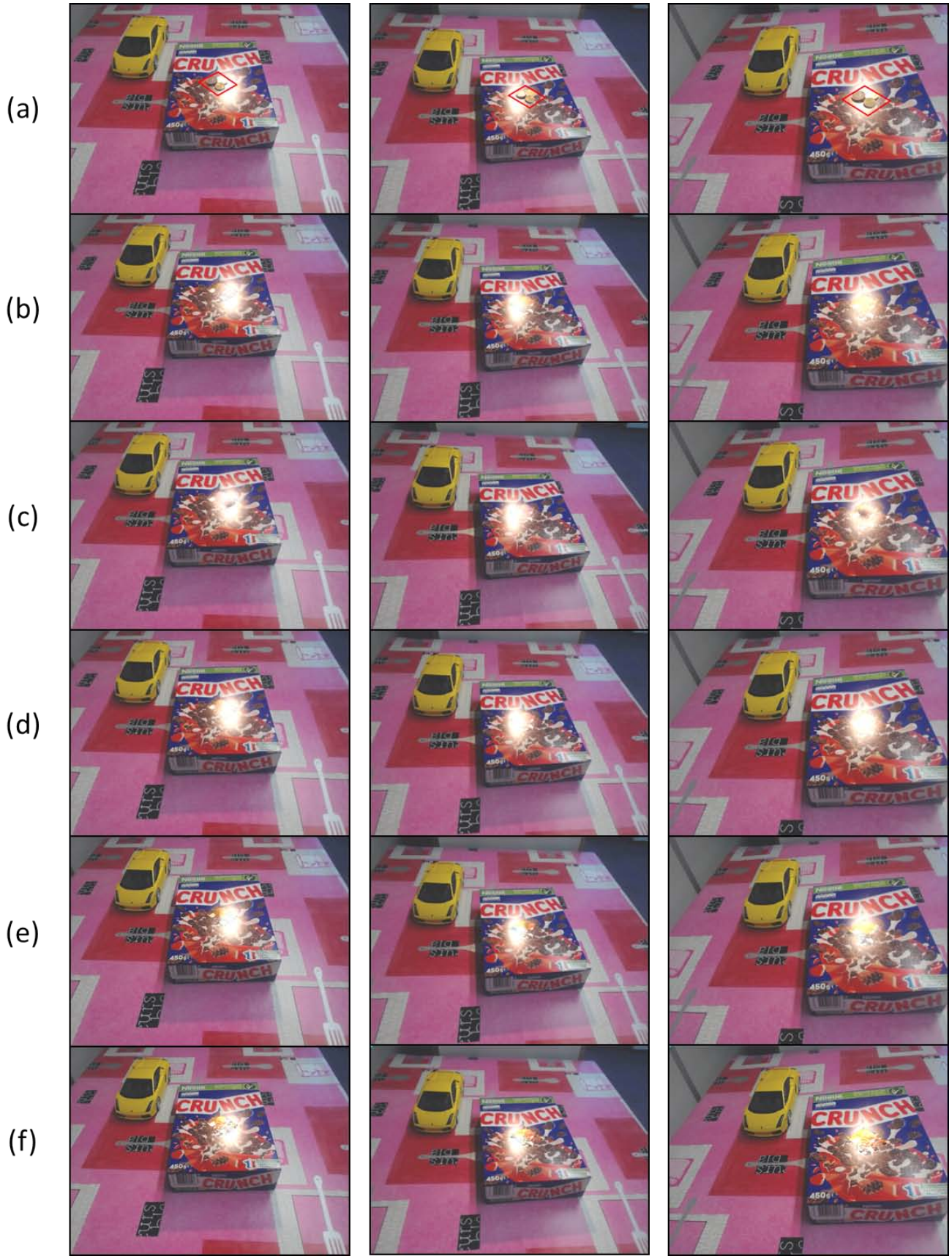


Fig. 13: Results of all methods on frames 637, 684 and 945 of video 4. (a) corresponds to the original image with the target region in red. (b) SPM. (c) CPM before filtering inconsistent ellipses. (d) CPM after filtering inconsistent ellipses. (e) Method of Herling *et al.* [18]. (f) Method of Kawai *et al.* [22].