Using Curvilinear Features in Focus for Registering a Single Image to a 3D Object

Hatem A. Rashwan, Sylvie Chambon, Pierre Gurdjos, Géraldine Morin and Vincent Charvillat

Abstract—In the context of 2D/3D registration, this paper introduces an approach that allows to match features detected in two different modalities: photographs and 3D models, by using a common 2D representation. More precisely, 2D images are matched with a set of depth images, representing the 3D model. After introducing the concept of curvilinear saliency, related to curvature estimation, we propose a new ridge and valley detector for depth images rendered from 3D model. A variant of this detector is adapted to photographs, in particular by applying it in multi-scale and by combining this feature detector with the principle of focus curves. Finally, a registration algorithm for determining the correct viewpoint of the 3D model and thus the pose is proposed. It is based on using histogram of gradients features adapted to the features manipulated in 2D and in 3D, and the introduction of repeatability scores. The results presented highlight the quality of the features detected, in terms of repeatability, and also the interest of the approach for registration and pose estimation.

I. INTRODUCTION

Many computer vision and robotic applications are used to take 2D contents as input, but, recently 3D contents are simultaneously available and popular. In order to benefit from both modalities, 2D/3D matching is necessary. For medical imaging, registration of pre-operative 3D volume data with intra-operative 2D images becomes more and more necessary to assist physicians in diagnosing complicated diseases easily and quickly [1]. For robotic, the 2D/3D matching is very important for many tasks that need to determine the 3D pose of an object of interest: 3D navigation or object grasping [2]. The main goal of 2D/3D registration is to find the transformation of the 3D model that defines the pose for a query 2D image. Thus, a typical 2D/3D registration problem consists of two mutually interlocked subproblems, point correspondence and pose estimation.

To match 2D photographs directly to 3D models or points clouds, most systems rely on detecting and describing features on both 2D/3D data and then on matching these features [6,7]. Recently, some approaches are based on learning by specific supervision classifier [3,4]. These methods produce very interesting results, however, they require huge amount of viewpoint-annotated images to learn the classifiers. What makes difficulty to the problem of matching 3D features of an object to 2D features of one of its photographs is that the appearance of the object dramatically depends on intrinsic characteristics of the object, like texture and color/albedo, as well as extrinsic characteristics related to the acquisition, like the camera pose and the lighting conditions. Consequently, some approaches manually define correspondences between the query image and the 3D model, such as [5]. These manual methods can be robust but it can easily become hard to apply this manual selection to large image sets. Moreover, in this paper, we focus on automated approaches. Note that some systems are able to generate a simultaneous acquisition of photographs and scanning of a 3D model but using this kind of systems induces limited applications. Other methods solve the problem by distinguishing two subproblems: to choose the common representation of the data and, then, to find the correspondences. These methods transforms the initial 2D/3D registration problem to a 2D/2D matching problem by rendering multiple 2D images of 3D models from different viewpoints, such as [8,9,10].

Consequently, the first task of 2D/3D registration is to find an appropriate representation of 3D models in which reliable features can be extracted in 2D and 3D data. In [8], synthetic images of the 3D model are rendered, while depth images are rendered in [9]. More recently, [10] proposes average shading gradients. This rendering technique for a 3D model averages the gradient normals over all lighting directions to cope with the unknown lighting of the query image. The advantage of representing the 3D model by a set of depth images is that it can express the model shape independently to color and texture information. Therefore, representing the 3D model by a set of depth images is the best option for this work, see Fig. 1. In this case, features extracted from depth images are only related to shape information.

Then, the second difficulty of 2D/3D registration consists in proposing how to match entities between the two modalities in this common representation. It can be partial [11] or dense matching, based on local or global characteristics [12]. In [8], silhouettes extracted from synthetic images are matched to ones extracted from the color images. However, this method is not able to take into account most of the occluding contours that are useful for accurate pose estimation. In turn, in [10], image gradients are matched with their 3D representation. Since image gradients are still affected by image textures and background, this technique can fail to estimate the correct correspondences. A key requirement on these features, as in classic 2D matching between real images, is to be computed with a high degree of repeatability. In our case, similar to the definition in [13], the repeatability of a feature is defined as the frequency with which one detected in the depth image is found within ε pixels around the same location in the corresponding intensity image (if it is supposed that the features are not moving or are following a small displacement). Then, since we suppose that an individual photograph of an object of
To compare 2D images with 3D models, we use a collection of rendered images of the 3D models from different viewpoints, and then we detect points of interest (curvilinear saliency) with common basis definitions between depth images and intensity images. For that evaluation, each depth image is compared with the original 2D image, based on these point of interest detection, and the proposed algorithm gives as output the depth image with the most similar point of view with the point of view of the 2D image.

interest is acquired, in a textured environment, we will focus on comparing pre-processed features of color images with features of a set of rendered images of 3D models, more precisely, a set of depth images, see Fig. 1.

More precisely, the 3D object will be given by a set of 3D depth surfaces, which describe how the original object surface is shortened by a perspective viewing and the image is given by the 3D intensity surface. Since the depth and the intensity surfaces have a different order of representation, the two surfaces cannot be directly matched. Thus, bringing both rendered depth images and photographs into a common representation, such as gradient and edge representation, allows to establish a robust sparse 2D-to-3D matching [10]. We propose to extract gradient-based features corresponding to object’s shapes in both depth and intensity images regardless of illumination and texture changes. In other words, as 2D photographs (intensity images) are affected by background, textures and lighting changes, we take into account these difficulties by reducing the influence of non-redundant information (i.e., color and texture) on features extracted from photographs. It means that we extract features in depth images that highlight geometric characteristics of an object. For photographs, we need to refine detected features by selecting salient points acquired by a camera in focus. These points are a function of the degree of blur (blurriness) in an image. Thus, the detected points are analyzed based on measuring the blur amount of every feature point. Finally, what we call focus points should be able to detect the approximate shape and to discard the other components such as textures.

To summary, the contributions of this paper, as shown in Fig. 1 are:

1) A ridge and valley detector, for depth images rendered from 3D model. We name it curvilinear saliency (CS) as it is related to the curvature estimation (a function of the eigenvalues derived of the Hessian matrix).

This representation directly relates to the discontinuities of the object’s geometry, and, by nature, the extracted features should be robust to texture and light changes.

2) A variant of this detector adapted to photographs. This curvilinear saliency detector is applied in multi-scale by searching over all scales and all image locations in order to identify scale-invariant interest points. To reduce the influence of structures due to texture and background regions, we introduce the extraction of focus curvilinear saliency features. It corresponds to ridges that are not affected by blur.

3) A registration algorithm for determining the correct viewpoint of the 3D model and thus the pose. This method is based on using histogram of gradients, HOG, features [14], adapted to the features manipulated in 2D and in 3D, and the introduction of repeatability scores. More precisely, the HOG descriptor is computed on both depth images (i.e., curvilinear features extracted with curvilinear saliency detection) and photographs (i.e., curvilinear features in focus extracted with multiscales curvilinear saliency detection) and it combines the curvilinear saliency value with the orientation of the curvature. The repeatability score measures the set of repeatable points detected both in a photograph and in the rendered depth images.

After presenting the related work and reminder on differential geometry, sections II and III we introduce the 3D model representation, section IV and, then, the image representation used for 2D/3D matching, section V. Then, we describe how we try to be robust to background and texture by using the same principle used in the detection of focus curves, section VI. We illustrate how this new global approach for 2D/3D matching allows to obtain more repeatable features, compared to state of the art, section VII. Finally, we explain how we obtained 2D/3D registration results, section VIII and
pose estimation, section [IX], by highlighting the interest of the proposed approach in these applications, before conclusion, section [X].

II. RELATED WORK

As mentioned earlier, a typical 2D/3D registration problem consists of two subproblems: feature correspondence and pose estimation (i.e., alignment). Thus, the related work is divided into three parts related to these subproblems: 1) detect features in 2D photography 2) detect features in a 3D model and finally 3) match 2D features to 3D features to estimate the 3D pose.

A. Classic 2D feature detection

In this section, we try to identify if, in the literature, it exists a 2D classical detector interesting in order to obtain points comparable as points detected, with the same principle or same tool, in 3D. We suppose that, for this purpose, it is necessary to detect features that are related to points of interest on the structure of the object and not on the texture or the light changes on the object. In 2D, edge detection [15] based on the first-order derivative information is the initial technique. It can detect any kind of edges, even low contrasted edges that are not due to the structure but more due to texture. The second technique is to detect the interest points [16] that refer to point-like features in an image by analyzing the eigenvalues of the structure tensor (i.e., the first-order derivative) at each point—the two eigenvalues have to be maximal to highlight a point of interest. Again, this technique does not take into account difficulties due to textures, light changes or scale changes. In another way, blob detection [18, 20, 19, 21] provides a complementary description of image structures in terms of regions, as opposed to point-like algorithms. These methods are based on the Hessian matrix (i.e., the second-order derivative) because of its independence to zero- and first-order changes and its good performance in computation time and accuracy. More recently, multi-scale approaches have been introduced, like a generalization of Harris or Laplacian detectors [22] or the well known approach of SIFT, Scale Invariant Features Transform [19]. In [23], SURF, Speeded Up Robust Features, a detector also based on Hessian matrix analysis, is introduced to be faster than SIFT and other multi-scale techniques by using approximation of Laplacian and algorithmic tricks. All these techniques are robust to light changes, rotations and translations. It makes the features detection invariant to viewpoint changes. However, they totally rely on texture and/or intensity changes to find the features.

Curvature detection is one of the most important techniques of second-order derivative-based approaches used for extracting the structure properties. Recently, [17] has proposed a detector based on curvature $\kappa$ expressed as the change of the image gradient along the tangent to obtain a scalar $\eta$ approximating $\kappa$. In addition, [24] presented PCBR, Principal Curvature-Based Regions, detector that uses the maximum or minimum eigenvalue of the Hessian matrix to find the principal curvature in a multi-scale space. As mentioned in [24], the maximum eigenvalue yields a high value only for the dark side of edges, i.e. the minimum eigenvalue detects light lines on a dark background. By definition, it is a restrictive way to select features and it does not guarantee to select the maximal number of reliable features.

In conclusion of this review, curvature features have several advantages over more traditional intensity-based features [17], especially with extracting local structure of the points of interest. In addition, particularly, curvature features are invariant to viewpoint changes and to transformations that do not change the shape of the surface.

B. Classic 3D feature detection

In this subsection, our goal is to find if, in the literature, some 3D detector can be directly adapted to 2D, in order to obtain comparable points of interest. Feature extraction of 3D models/scenes can be classified into point-based and image-based approaches. Most of point-based methods are based on using SIFT in 3D by proposing an adaptation of the initial SIFT [25, 26]. These approaches are interesting but they are not dedicated to 2D/3D registration and, so, they do not consider to detect similar features in 2D and in 3D. Other methods proposed to find curves that have special properties in terms of differential geometry of the surface. For example, in [27], curves are supposed to be located at paraboloic lines which occur at points of extremal curvature. These curves capture important object properties closed to the object surface, but they do not vanish along the surface when the viewpoint changes.

In image-based approaches, the 3D model is first rendered to form images or geometric buffers. Then image processing methods are applied such as edge detection [28]. In [29], 2D SIFT is applied on images of a rendered 3D mesh because this multi-scale representation extracts features that are supposed to be related to local extrema of the surface variation. The apparent ridges (AR), which are a set of curves whose points are local maxima on a surface, are introduced in [30]. In this paper, a view-dependent curvature corresponds to the variation of the surface normal with respect to a viewing screen plane. However, apparent ridges often produce false edges that are not related to occluding contours, which are important for pose estimation of most of objects. Mesh saliency measures the region importance of 3D models using Gaussian-weighted mean curvatures in multi-scales [31]. However, mesh saliency is based on mesh images that are also affected by lighting conditions. Average Shading Gradients, ASG, was proposed in [10]. This rendering technique is based on averaging gradients over all lighting directions to cope with the unknown lighting conditions.

In conclusion, only [10] can be used in a 2D/3D registration context and we will describe, in section [V], and compare our work with this method, in section [VII].

C. 2D/3D registration and pose estimation

In the computer vision literature, the problem of automatically aligning 2D photographs with an existing 3D model of the scene has been investigated in depth over the past fifteen years. In the general case, the proposed solution will be an image-to-model registration to estimate the 3D pose of the
object. The 2D-to-3D registration problem is approached in the literature through indirect and direct methods.\cite{32}

For indirect registration, these methods are performed either by 3D-to-2D registration or by finding some appropriate registration parameters, such as the standard Iterative Closest Point, ICP algorithm.\cite{65} This kind of techniques are more global and do not really provide points to points correspondences.

For direct registration methods, in\cite{26}, correspondences are obtained by matching SIFT feature descriptors between SIFT points extracted in the images and in the 3D models. However, establishing reliable correspondences may be difficult due to the fact that the set of points in 2D and in 3D are not always similar, in particular because of the variability of the illumination conditions during the 2D and 3D acquisitions. In the same context, in\cite{45}, the authors assume that the object in the input image has no or poor internal texture. Methods relying on higher level features, such as lines\cite{33}, planes\cite{34} and building bounding boxes\cite{35}, are generally suitable for Manhattan World scenes and hence applicable only in such environments. Skyline-based methods\cite{36} as well as methods relying on a predefined 3D model\cite{39} are, likewise, of limited applicability. Recently, the histogram of gradients, HOG detector\cite{44} or a fast version of HOG\cite{41} have been also used to extract the features from rendering views and real images. All of these approaches give interesting results, however, they do not evaluate the repeatability between the set of points detected in an intensity image and those detected in an image rendered from the 3D model. Finally, in\cite{10}, 3D corner points are detected using the 3D Harris detector and the rendering average shading gradients images on each point. For a query image, similarly, corner points are detected in multiscale. Then, the gradients computed for patches around each point is matched with the database containing average shading gradient images using HOG descriptor. This method still relies on extracting gradients of photographs affected by textures and background and it can give erroneous correspondences. Consequently, they propose a refine stage based on RANdom SAmple Consensus, RANSAC\cite{55} to improve the final pose estimation.

In this paper, structural cues (e.g., curvilinear shapes) based on curvilinear saliency are extracted instead of only considering silhouettes, since they are more robust to intensity, color, and pose variations. In fact, they have the advantage to both represent outer and inner (self-occluding) contours that also characterize the object and that are useful for estimating the pose. In order to merge in the same descriptor curvilinear saliency values and curvature orientation, the HOG descriptor, widely used in the literature and that properly describes the object shape, is employed. Finally, HOG features and repeatability scores are used to match the query image with a set of depth images rendered from a 3D model.

In the rest of the paper, after the reminder on differential geometry, we will describe the 3D model representation and the image representation that are introduced in order to compare and to match the 3D and the 2D data. In particular, we will explain how these representations allow to be robust to background details and to texture before illustrating the interest of the proposition with experimental results on feature detection, registration between 2D images and 3D models and pose estimation.

III. REMINDER ON DIFFERENTIAL GEOMETRY

Notations

In the sequel, these notations are used:

- \( \nabla f \): the gradient vector of a scalar-valued function \( f \).
- \( F_x \): the partial first-order derivative \( \frac{\partial F}{\partial x} \) of a vector-valued function \( F \) w.r.t. variable \( x \).
- Similarly, \( F_{xy} \): the partial second-order derivatives \( \frac{\partial^2 F}{\partial x \partial y} \) of \( F \) w.r.t. variables \( x \) and \( y \).

Moreover, we assume a calibrated perspective camera where the image point coordinates are given with respect to the normalized image frame i.e., as \( x = (x, y) \) obtained from the equation \( [x, y, 1] = K^{-1} [u, v, 1] \), where \( (u, v) \) are pixel coordinates and \( K \) is the usual calibration upper triangular matrix.\cite{63}

A. Definitions: Differential of a Map – Tangent Plane – Gauss Map

Let \( F \subset \mathbb{R}^3 \) be a regular surface\cite{49} whose parameterization is given by the differentiable map \( F: U \subset \mathbb{R}^2 \to \mathbb{R}^3 \) with

\[
F(x, y) = [X(x, y), Y(x, y), Z(x, y)]^T 
\]

To each \( x = [x, y]^T \in U \) is associated a map \( dF_x : \mathbb{R}^2 \to \mathbb{R}^3 \), called the differential of \( F \) at \( x \) and defined as follows\cite{62, p. 128}. Let \( v \in \mathbb{R}^2 \) be a vector and let \( \alpha : (-\epsilon, \epsilon) \to U \) be a differentiable curve satisfying \( \alpha(0) = x \) and \( \alpha'(0) = v \). By the chain rule, the curve \( F \circ \alpha \) in \( \mathbb{R}^3 \) is also differentiable. We then define

\[
dF_x(v) = (F \circ \alpha)'(0)
\]

It provides a linear (i.e., first-order) approximation to \( F(x + v) \) when the increment \( v \) is small enough. This is illustrated in Fig.\ref{fig:2}(a). The vector subspace \( dF_x(U) \subset \mathbb{R}^3 \) has dimension 2 and is a plane consisting of all tangent vectors of \( F \) at \( P = F(x) \). It is called the tangent plane of \( F \) at \( P \) and denoted by \( T_P(F) \).

It can be proved\cite{62, p129} that the above definition does not depend on the choice of \( \alpha \). Furthermore, the fact that \( (F \circ \alpha)'(0) = F'(\alpha(0))\alpha'(0) \) entails that \( dF_x(v) \) is linear in \( v \). In particular, in the canonical bases of \( \mathbb{R}^2 \) and \( \mathbb{R}^3 \), we have

\[
dF_x(v) = \begin{bmatrix} J_F(x) & F_y(x) \\ F_x & 0 \end{bmatrix} v
\]

involving the \( 3 \times 2 \) Jacobian matrix of \( F \) at \( x \)

\[
J_F(x) = \begin{bmatrix} F_x(x) & F_y(x) \end{bmatrix}
\]

with \( F_x \) and \( F_y \) as columns. This also shows that the vector subspace \( dF_x(U) \) has indeed dimension 2.

Let \( P = F(x) \) be a point of \( F \). Let \( N : F \subset \mathbb{R}^3 \to \Sigma \subset \mathbb{R}^3 \) be the differentiable map that assigns to \( P \) the coordinate

\[\text{See the definition of a regular surface in } \mathbb{R}^n \text{ in }\cite{60, p. 281-286].\]
vector $\mathbf{N}(\mathbf{P})$ on the unit sphere $\Sigma$ representing the unit normal of $\mathcal{F}$ at $\mathbf{P}$ and computed as

$$\mathbf{N}(\mathbf{P}) = \frac{\mathbf{F}_x(x) \times \mathbf{F}_y(x)}{\|\mathbf{F}_x(x) \times \mathbf{F}_y(x)\|} \text{ with } x = \mathbf{F}^{-1}(\mathbf{P}) \quad (3)$$

This map is called the Gauss map of $\mathcal{F}$.

The Gauss map is a mapping between the two surfaces $\mathcal{F}$ and $\Sigma$ and the definition of differential is extended to that case. The differential of the Gauss map of $\mathcal{F}$ at point $\mathbf{P}$ is the map $\mathrm{d}\mathbf{N}_\mathbf{P} : T_{\mathbf{P}}(\mathcal{F}) \subset \mathbb{R}^3 \rightarrow \mathbb{R}^3$ defined as follows. Let $\mathbf{V} \in T_{\mathbf{P}}(\mathcal{F})$ be a vector on the tangent plane of $\mathcal{F}$ at $\mathbf{P}$ and let $\beta : (-\epsilon, \epsilon) \rightarrow \mathcal{F}$ be a differentiable curve on the surface $\mathcal{F}$ satisfying $\beta(0) = \mathbf{P}$ and $\beta'(0) = \mathbf{V}$. By the chain rule, the curve $\mathbf{N} \circ \beta$ in $\mathbb{R}^3$ is also differentiable; we then define

$$\mathrm{d}\mathbf{N}_\mathbf{P}(\mathbf{V}) = (\mathbf{N} \circ \beta)'(0)$$

It expresses how $\mathbf{N}$ behaves — how $\mathcal{F}$ curves — in the vicinity of $\mathbf{P}$. This is illustrated in Fig. 2(b). Again, it can be proved [62, p129] that the above definition of $\mathrm{d}\mathbf{N}_\mathbf{P}$ does not depend on the choice of one possible curve $\beta$.

Fig. 2: (a) A map $\mathbf{F}$ parameterizing a regular surface $\mathcal{F}$ and its differential at point $\mathbf{x}$ along direction $\mathbf{w}$. (b) The map $\mathbf{N}$ is the Gauss map (which is a mapping between the surface $\mathcal{F}$ and the unit sphere $\Sigma$) and differential at $\mathbf{P}$ along $\mathbf{T}$.

Similar to the differential of a map seen above, the fact that $(\mathbf{N} \circ \beta)'(0) = \mathbf{N}'(\beta(0))\beta'(0)$ entails that $\mathrm{d}\mathbf{N}_\mathbf{P}(\mathbf{V})$ is linear in $\mathbf{V} = \beta'(0)$. The vector subspace $\mathrm{d}\mathbf{N}_\mathbf{P}(T_{\mathbf{P}}(\mathcal{F})) \subset \mathbb{R}^3$ has dimension 2: it is the plane $T_{\mathbf{N}(\mathbf{P})}(\Sigma)$ consisting of all tangent vectors to the unit sphere at point $\mathbf{N}(\mathbf{P})$ called the tangent plane of $\Sigma$ at $\mathbf{N}(\mathbf{P})$. Therefore the domain of values of $\mathrm{d}\mathbf{N}_\mathbf{P}$ is $\mathrm{d}\mathbf{N}_\mathbf{P}(T_{\mathbf{P}}(\mathcal{F})) = T_{\mathbf{N}(\mathbf{P})}(\Sigma)$. Actually, it can readily seen that $T_{\mathbf{N}(\mathbf{P})}(\Sigma)$ and $T_{\mathbf{P}}(\mathcal{F})$ are parallel planes so the differential of $\mathbf{N}$ is usually defined as $\mathrm{d}\mathbf{N}_\mathbf{P} : T_{\mathbf{P}}(\mathcal{F}) \rightarrow T_{\mathbf{P}}(\mathcal{F})$.

### B. Curvatures. Fundamental forms of a surface

Let $\mathbf{T} \in T_{\mathbf{P}}(\mathcal{F})$ be a unit vector representing a direction on the tangent plane of $\mathcal{F}$ at $\mathbf{P}$. Let $\mathcal{C}$ be the curve obtained by slicing $\mathcal{F}$ with the normal section of $\mathcal{F}$ at $\mathbf{P}$ along $\mathbf{T}$ i.e., the plane through $\mathbf{P}$ parallel to both $\mathbf{N}(\mathbf{P})$ and $\mathbf{T}$. The normal curvature of $\mathcal{C}$ in (unit) direction $\mathbf{T}$ is the curvature of $\mathcal{C}$ at $\mathbf{P}$ which can be given by [62, p. 144]:

$$\kappa_N(\mathbf{T}) = -\mathbf{T} \cdot \mathrm{d}\mathbf{N}_\mathbf{P}(\mathbf{T}) \quad (4)$$

It expresses how $\mathbf{N}$ behaves — how $\mathcal{F}$ curves — in the vicinity of $\mathbf{P}$. An important remark is that the radius of curvature of $\mathcal{C}$ at $\mathbf{P}$ is equal to $1/|\kappa_N|$.

The principal curvatures $\kappa_1, \kappa_2$ of $\mathcal{F}$ at $\mathbf{P}$ can be defined as the extrema of function (4) with respect to directions $\mathbf{T} \in T_{\mathbf{P}}(\mathcal{F})$, subject to the constraint $||\mathbf{T}|| = 1$. The corresponding directions are called principal directions of $\mathcal{F}$ at $\mathbf{P}$. It is well-known that “the coefficients $\kappa_1, \kappa_2$ are decisive parameters that fully describe local surface shape up to the second order modulo a rigid movement.” [64] This means that the principal curvatures are invariant to the surface parameterization.

Now consider a new 3D coordinate system with $\mathbf{P} = \mathbf{F}(\mathbf{x})$ as origin. Any vector $\mathbf{T}$ on the tangent plane $T_{\mathbf{P}}(\mathcal{F})$ can be written as

$$\mathbf{T} = u\mathbf{F}_x(\mathbf{x}) + v\mathbf{F}_y(\mathbf{x}) = \mathbf{J}_\mathbf{F}(\mathbf{x}) \begin{bmatrix} u \\ v \end{bmatrix} \quad (5)$$

where $\mathbf{J}_\mathbf{F}(\mathbf{x})$ is the Jacobian matrix of $\mathbf{F}$ defined in (2) and $(u,v)$ are so-called local coordinates of $\mathbf{T}$. From now on, we will put $uv$ as subscript to relate a vector to its local coordinates, e.g., $\mathbf{T}_{uv}$.

1) **First fundamental form of a surface**: Given any $(u,v)$, the norm of any vector $\mathbf{T}_{uv}$ on $T_{\mathbf{P}}(\mathcal{F})$ writes

$$||\mathbf{T}_{uv}|| = \sqrt{[u,v] \mathbf{I}_\mathbf{P} [u,v]^T} \quad (6)$$

where

$$\mathbf{I}_\mathbf{P} = \begin{bmatrix} \mathbf{F}_x(x) \cdot \mathbf{F}_x(x) & \mathbf{F}_x(x) \cdot \mathbf{F}_y(x) \\ \mathbf{F}_x(x) \cdot \mathbf{F}_y(x) & \mathbf{F}_y(y) \cdot \mathbf{F}_y(y) \end{bmatrix}$$

The quadratic form on $T_{\mathbf{P}}(\mathcal{F})$

$$\mathbf{J}_\mathbf{F}(u,v) = [u,v] \mathbf{I}_\mathbf{P} [u,v]^T \quad (7)$$

is called the first fundamental form of $\mathcal{F}$ [62, p94].

2) **Second fundamental form of a surface**: Let $\mathbf{T}_{uv}$ be a direction of the tangent plane $T_{\mathbf{P}}(\mathcal{F})$, given in local 2D coordinates. On the one hand, it can be shown [62, p156] that the differential of the Gauss map of $\mathcal{F}$ at $\mathbf{P} = \mathbf{F}(\mathbf{x})$ along $\mathbf{T}_{uv}$ writes in standard 3D coordinates

$$\mathrm{d}\mathbf{N}_\mathbf{P}(\mathbf{T}_{uv}) = [\mathbf{N}_x(x) \mid \mathbf{N}_y(y)] \begin{bmatrix} u \\ v \end{bmatrix}$$

On the other hand, let denote by $\mathrm{d}\mathbf{N}_\mathbf{P}(u,v)$ the differential of $\mathbf{N}$ at $\mathbf{P} = \mathbf{F}(\mathbf{x})$ along $\mathbf{T}_{uv}$ expressed in local 2D coordinates i.e., such that $\mathrm{d}\mathbf{N}_\mathbf{P}(\mathbf{T}_{uv}) = \mathbf{J}_\mathbf{F}(\mathbf{x})[\mathrm{d}\mathbf{N}_\mathbf{P}(u,v)]$. Then we have

$$\mathrm{d}\mathbf{N}_\mathbf{P}(u,v) = \mathbf{I}_\mathbf{P}^{-1} \mathbf{J}_\mathbf{F} \begin{bmatrix} u \\ v \end{bmatrix} \quad (8)$$

\footnote{This holds for any plane through $\mathbf{P}$ parallel to $\mathbf{T}$. This is due to Meusnier’s theorem [58, p482] “All curves lying on a surface $\mathcal{S}$ and having at a given point $\mathbf{P} \in \mathcal{S}$ the same tangent line have at this point the same normal curvatures.”}
The proof can be found in [62, p156].

The quadratic form
\[ \mathcal{D}(u,v) = [u,v] \Pi_P[u,v]^\top \] (10)
is called the second fundamental form of \( F \) [62, p143]. It directly follows from this that Eq. (11) can be expressed in local coordinates and writes
\[ \kappa_N(u,v) = [u,v] \Pi_P[u,v]^\top \] (11)
For any (non unit) vector \( \mathbf{v} \) on \( T_p(S) \), in order to set it as a unit direction, it is needed to divide \( \Pi_P \) by the square of expression (6). Hence, the normal curvature in the unit direction \( \mathbf{v}/\|\mathbf{v}\| \) is can be given by
\[
\kappa_N(u,v) = \frac{-\nabla u \cdot \nabla N_P(\mathbf{v})}{\|\nabla u\|^2} = \mathcal{D}(u,v)\mathbf{v} \] (12)

3) Closed-form solutions for principal curvatures: The principal curvatures \( \kappa_1, \kappa_2 \) of \( S \) at \( P \) can be defined as the extrema of function (12) with \( (u,v) \)-coordinates as variables.

Seeing (12) as a generalized Rayleigh quotient, it is known [27, p18] that \( \kappa_N \) has an extremum at \((\mathbf{v},\mathbf{v})\) only if \( \kappa_N(\mathbf{v},\mathbf{v}) \) is a root of \( \det(\Pi_P - \kappa N_P) \) or, equivalently, only if \( \kappa_N(\mathbf{v},\mathbf{v}) \) is an eigenvalue of the \( 2 \times 2 \) matrix \( \Pi_P^{-1} \Pi P \), which is not symmetric but always has real eigenvalues [38] p500.

As a result, the principal curvatures are the two eigenvalues \( \kappa_\alpha \) \((\alpha = 1,2)\) of the matrix \( \Pi_P^{-1} \Pi P \). The principal 3D directions are \( \mathbf{T}\alpha = J_P(x)e_\alpha \) where \( e_\alpha \) are the corresponding eigenvectors.

Now we state a proposition that we derive from the above results, which will be used in our work.

Proposition 1: The principal curvature \( \kappa_\alpha \) \((\alpha = 1,2)\) at \( P \) associated to the unit principal 3D direction \( \mathbf{T}\alpha \) is equal to the absolute magnitude of the differential of the Gaussian map at this point i.e.,
\[
|\kappa_\alpha| = \|\nabla N_P(\mathbf{T}\alpha)\| \] (13)

Proof. Since the Euclidean norm is invariant to changes of Euclidean coordinates, without loss of generality, let choose a new parameterization \( \mathbf{S}(\bar{x},\bar{y}) = [\bar{x},\bar{y},\mathbf{Z}(\bar{x},\bar{y})]^\top \) for some new height function \( \mathbf{Z} \), w.r.t. 3D orthonormal frame whose origin is \( P \) and \( \bar{x}\bar{y}\)-plane coincides with the tangent plane \( T_P(F) \). More generally, we will add the symbol \( \bar{\cdots} \) to distinguish the new representations from the old ones, except for the principal curvatures which are irrespective of parameterizations. Let us remind that \( \mathbf{T}\alpha = J_P(x)e_\alpha \), where \( e_\alpha \) is the associated eigenvector, and note that the first new fundamental matrix, \( \bar{\Pi}_P \), is then the identity. As a result, starting from the fact that \( \kappa_\alpha \) is an eigenvalue of the \( 2 \times 2 \) matrix \( \Pi_P^{-1} \Pi P \), we have:
\[
(\bar{\Pi}_P^{-1} \bar{\Pi}P)\bar{e}_\alpha = \kappa_\alpha \bar{e}_\alpha
\]
\[
\Leftrightarrow \nabla[\bar{S}_x | \bar{S}_y]d\bar{N}_P(\bar{e}_\alpha) = \kappa_\alpha [\bar{S}_x | \bar{S}_y]d\bar{e}
\]
\[
\Leftrightarrow \nabla[\bar{T}\alpha]d\bar{N}_P(\bar{e}_\alpha) = \kappa_\alpha \bar{T}\alpha
\]
\[
\Rightarrow \|d\bar{N}_P(\bar{T}\alpha)\|^2 = \kappa_\alpha^2
\]

IV. 3D MODEL REPRESENTATION

The work the most related to what is proposed in this paper is the Average Shading Gradient (ASG) approach, proposed in [10]. After introducing how object surface can be represented, we highlight the differences between these two approaches.

Object surface: Denote by \( \mathcal{M} \) the surface of some observed object associated to a parameterization \( \mathbf{M}(x,y) \triangleq [X(x,y), Y(x,y), Z(x,y)]^\top \), where \((x,y)\) varies over the restricted image domain of a given camera which is delimited by the occluding contour of the object. Under perspective projection, every visible 3D point of \( \mathcal{M} \) (seen from the camera viewpoint), with vector \( \mathbf{M}(x,y) \), is assumed to be in one-to-one correspondence with the 2D image point with vector \( \mathbf{x} = [x, y]^\top \), such that \[x = X(x,y)/Z(x,y) \text{ and } y = Y(x,y)/Z(x,y) \] as a result, we get
\[
\mathbf{M}(x,y) = Z(x,y)[x, y, 1]^\top \] (14)

Let \( \mathbf{N}(x,y) \) denotes the Gaussian map of \( \mathcal{M} \) which assigns, on the unit sphere, to each point point \( \mathbf{M}(x,y) \) of \( \mathcal{M} \) the unit normal of \( \mathcal{M} \) defined by \[
\mathbf{N}(x,y) = \frac{\bar{N}(x,y)}{\|\bar{N}(x,y)\|} \] where, using \[14\], \( \bar{N} \) writes
\[
\bar{N} = \mathbf{M}_x \times \mathbf{M}_y = Z[-Z_x, -Z_y, xZ_x + yZ_y + Z]^\top \] (15)

It can be shown that the Jacobian \( 3 \times 2 \) matrix of \( \mathbf{N} \) writes
\[
\mathbf{J}_N = [\mathbf{N}_x \quad \mathbf{N}_y] = (I - \mathbf{NN}^\top) \mathbf{J}_N \] (16)
where the columns of \( \mathbf{J}_N = [\mathbf{N}_x \quad \mathbf{N}_y] \) have the form
\[
\mathbf{N}_\mathbf{x} = \begin{bmatrix}
Z_xZ_x - Z_{xx}Z \\
Z_yZ_y - Z_{yy}Z
\end{bmatrix} + \begin{bmatrix}
xZ_xZ + yZ_{xy}Z + Z \times (xZ_x + yZ_y + 3Z)
\end{bmatrix}
\] (17)

A. Average Shading Gradient (ASG) Feature [10]

Plötz et al. assumed in [10] that the image intensity function obeys the Lambertian shading function
\[
I(x,y) \propto \max(0, -\mathbf{N}(x,y) \cdot \mathbf{s}) \] (18)
for a parallel light source \( \mathbf{s} \in \mathbb{R}^3 \). This means that the reflectance describing the object material is assumed to be Lambertian with constant albedo. In addition, the background is assumed to be constant (e.g., a plane).

The authors propose as feature in the intensity image the magnitude of the gradient of the shading function. To register the intensity image to the 3D (untextured) model, the idea is to generate virtual images viewing the object from different camera pose candidates. Nevertheless, it is clearly impossible to render any such virtual image obeying the shading function (18) without prior information about the lighting direction and so about \( \mathbf{s} \). Therefore, the authors propose to replace the gradient magnitude feature, in the virtual images, by a feature

\footnote{A general shading function is \( I(x,y) = \rho(\mathbf{M}(x,y)) \max(0, -\mathbf{N}(x,y) \cdot \mathbf{s}) \) where \( \rho(\mathbf{M}(x,y)) \) is the albedo at object point \( \mathbf{M}(x,y) \).}
corresponding to the average value of the gradient magnitude computed over all light directions, so-called average shading gradient magnitude. Denoting \( \| \nabla I \|^2 = I_x^2 + I_y^2 \) the magnitude of the gradient of the shading function \( I \), then the magnitude of the average shading gradient is:
\[
\| \nabla I \| = \int_S \| \nabla I \| \, ds \quad (19)
\]
where the vector \( s \), cf. \( \text{(18)} \), varies over the unit sphere \( S \) in \( \mathbb{R}^3 \) and \( ds \) is the volume element.

The nice contribution of Plötz et al. is, by applying Jensen’s inequality, to derive the following closed-form bound on \( \| \nabla I \| \):
\[
\| \nabla I \| \leq \sqrt{\int_S \| \nabla I \|^2 \, ds} = \gamma \sqrt{\left( \| N_{x} \|^2 + \| N_{y} \|^2 \right)} \quad (20)
\]
with \( \gamma = \sqrt{\frac{\pi}{2}}. \) It is reported by the authors to behave like a very good approximation of \( \| \nabla I \| \). This is the elegant way the authors get rid of the unknown lighting direction \( s \) in \( \text{(18)} \).

### B. Curvilinear Saliency Features (CS)

As already mentioned, our goal is to find a common representation between the 3D model and the 2D image in order to be able to match them. For that purpose, we first show how the 3D model can be represented or studied from different points of view and how these different viewpoints can be analyzed and compared to a 2D image. For that purpose, we represent the observed 3D object by a set of synthetic depth maps generated from camera locations distributed on concentric spheres encapsulating, by sampling elevation and azimuth angles, as well as distances from the camera to the object. A depth map (or depth image) \( Z(x, y) \) associates to every image point \((x, y)\) the \( Z\)-coordinate, w.r.t. the camera frame, of the object 3D point \((\tilde{x}, \tilde{y})\) that projects onto \((x, y)\).

Let \( D \) denote the depth surface that is the 3D surface whose graph parameterization is:
\[
D(x, y) = [x, y, Z(x, y)]^\top
\]

Which features should be extracted in the depth map? We aim at detecting depth “discontinuities” by searching points on \( D \) having high principal curvature in one direction and low principal curvature in the orthogonal direction. We call Curvilinear Saliency features of a surface loci of such points. Basically, they correspond to the ridges and valleys of this surface. In this work, we use the difference of the principal curvatures \( \kappa_1 - \kappa_2 \) to describe the ridges and valleys.

**Principal curvatures and directions:** Consider a point \( P = D(x, y) \). Let \( N'(x, y) \) denote the Gaussian map of \( D \) assigning to \( P \) the unit normal of \( D \) at \( P \), such that
\[
N' = \frac{N'}{\| N' \|} \quad \text{where} \quad N' = D_x \times D_y = \alpha \left[ -\nabla Z \quad 1 \right]^\top \quad (21)
\]
with \( \nabla Z = [Z_x, Z_y]^\top \) and \( \alpha = 1/\sqrt{1 + \| \nabla Z \|^2}. \)

As the two columns of the Jacobian matrix \( J_D \) of \( D \) are \( D_x = [1, 0, Z_x]^\top \) and \( D_y = [0, 1, Z_y]^\top \), the first fundamental form of \( D \) can be computed as
\[
I_P = I_3 + \nabla_Z \nabla_Z^\top
\]
and the second fundamental form of \( D \) can be computed as
\[
\Pi_P = \alpha H_Z
\]
where \( H_Z \) is the Hessian matrix of \( Z \) i.e., with the second-order partial derivatives of \( Z \) w.r.t. \( x \) and \( y \) as elements.

The **principal curvatures** of \( D \) at \( P \) coincide with the eigenvalues \( \kappa_\alpha (\alpha = 1, 2) \) of \( \Pi_P^{-1} I_P \), which are always real. In the tangent plane \( T_P(D) \), the local coordinates of the principal directions of \( D \) at \( P \) are given by the eigenvectors \( e_\alpha \) of \( \Pi_P^{-1} I_P \) so the 3D principal directions in 3D wrote \( J_P e_\alpha \).

As Koenderink writes in \[64\], “it is perhaps not superfluous to remark here that the simple (eigen-)interpretation in terms of \[1\] of \( \Pi_P = \alpha H_Z \) is only valid in representations where \( \nabla Z = 0 \), which is the condition for the point to be local extremum.

Thanks to proposition [1] we know that the principal curvature \( \kappa_\alpha \) at \( P \) associated to the principal 3D direction \( T_\alpha = J_P e_\alpha \) is equal to the absolute magnitude of the change of the normal
\[
|\kappa_\alpha| = \| dN_P(T_\alpha) \| \quad (23)
\]
where \( dN_P(T) \) denotes the differential of \( N' \) at \( P \) in direction \( T \). We will make us of this result for the image representation, cf. [1].

Now let us explain why we propose as feature the difference \( \kappa_1 - \kappa_2 \) where \( \kappa_1 \geq \kappa_2 \).

**Curvilinear feature:** Without losing generality, let \( \kappa_1 \) and \( \kappa_2 \) be the principal curvatures computed as ordered eigenvalues of \( \Pi_P^{-1} I_P \) so that \( \kappa_1 \geq \kappa_2 \). We aim at detecting points lying on “elongated” surface parts. In this work, we detect points at which this difference is high:
\[
CS(x, y) = \kappa_1(x, y) - \kappa_2(x, y)
\]

We call \[24\] the *curvilinear saliency* (CS) feature. Curvilinear means a feature that belongs to a curved line. The rest of this paragraph justifies such a choice.

Given a point \( P \) on \( D \), let \((\tilde{x}, \tilde{y})\) be the Cartesian coordinates on the tangent plane \( T_P(D) \) w.r.t. the 2D frame whose origin is \( P \) and the orthonormal basis is formed by the principal directions \( \{e_1, e_2\} \). As a result, \( D \) can now locally be associated to the new parameterization \( F(\tilde{x}, \tilde{y}) = [\tilde{x}, \tilde{y}, F(\tilde{x}, \tilde{y})]^\top \), for some height function \( F \). In that case, it can be readily seen that \( I_P \) is the identity matrix, and so \( I_P \Pi_P I_P = \Pi_P = \text{diag}(\kappa_1, \kappa_2) \) is exactly the Hessian matrix of \( F \). For some \( \epsilon > 0 \) small enough, consider on the two planes parallel to \( T_P(D) \) at distances \( \pm \epsilon \) from \( T_P(D) \), the curves \( \mathcal{C}_\pm = \{(\tilde{x}, \tilde{y}), F(\tilde{x}, \tilde{y}) \in T_P(D) \mid F(\tilde{x}, \tilde{y}) = \pm \epsilon \} \). It can be shown \[38\] p500 that the first-order approximation of the intersections of \( D \) with the two parallel planes is the union of two conics (one real and one virtual) with equations \( \Pi_P(\tilde{x}, \tilde{y}) = \pm 2 \epsilon \). This union is known as the *Dupin indicatrix* when written in canonical form (i.e., by replacing \( 2 \epsilon \) by \( 1 \)). The real Dupin conic characterizes the local shape of \( D \) and gives

\footnote{Note the difference with \[14\].}

\footnote{By neglecting \( I_P \).}
local information on the first-order geometry of the surface, at least at points where the conic is non degenerate. It specializes to a parabola if the Gauss curvature vanishes i.e., \( \kappa_1 \kappa_2 = 0 \), to an ellipse if \( \kappa_1 \kappa_2 > 0 \), and to a hyperbola if \( \kappa_1 \kappa_2 < 0 \), see Fig. \[3\]. Points are said to be elliptic, hyperbolic or parabolic respectively.

A simple way to compute the curvilinear feature: After algebraic manipulations, it can be shown that \( \Gamma_p^{-1} \Pi_p = \frac{1}{\alpha} \mathcal{M} \) where

\[
\mathcal{M} = \begin{bmatrix}
(Z_x^2 + 1)Z_{xx} - Z_x Z_y Z_{xy} & (Z_y^2 + 1)Z_{xy} - Z_x Z_y Z_{yy} \\
(Z_y^2 + 1)Z_{xy} - Z_x Z_y Z_{xx} & (Z_x^2 + 1)Z_{yy} - Z_x Z_y Z_{xy}
\end{bmatrix}
\]

(26)

**Proposition 3:** The squared curvilinear feature can be computed as

\[
\text{CS}^2 \triangleq \| \nabla_Z \|^2 \left( (\text{trace } \mathcal{M})^2 - 4 \det \mathcal{M} \right)
\]

(27)

\[
= 4 \| \nabla_Z \|^2 \left( \bar{\kappa}^2 - K \right)
\]

(28)

where \( \bar{\kappa} \) is the mean curvature of \( D \) and \( K \) its Gaussian curvature.

**Proof.** Let \( \lambda_1 \) and \( \lambda_2 \) be the ordered eigenvalues (\( \lambda_1 \geq \lambda_2 \)) computed from \(-\mathcal{M}\) as defined in \( \mathcal{M} \), to have \( \kappa_i = \| \nabla_Z \| \lambda_i \), \( i = 1, 2 \). As the two eigenvalues of \(-\mathcal{M}\) write

\[
\lambda_{\pm} = \frac{1}{2} \left( -\text{trace} (\mathcal{M}) \pm \sqrt{\left( \text{trace} (\mathcal{M}) \right)^2 - 4 \det \mathcal{M}} \right)
\]

(29)

we have \( \lambda_1 = \lambda_+ \) and \( \lambda_2 = \lambda_- \). Since as well as

\[
\kappa_1 - \kappa_2 = \| \nabla_Z \| (\lambda_1 - \lambda_2)
\]

(30)

the squared curvilinear saliency is then defined.

However, the rely on the highest or smallest principal curvature alone is not adequate for defining accurate ridges \([42]\). In Fig. \[4\] we show the different detections obtained using the minimum or the maximum principal curvature, as proposed by \([24]\). The maximum principal curvature provides a high response only for dark lines on a light background, while the minimum gives the higher answers for the light lines on a dark background. The difference of the principal curvatures, \( \kappa_1 - \kappa_2 \), improves robustness as it responds in both settings.

V. IMAGE REPRESENTATION

A. Proposed Curvilinear Features for Images

Let \( I(x, y) \) denote the value of the image intensity function \( I : U \subset \mathbb{R}^2 \rightarrow \mathbb{R} \) at image point \( (x, y) \). Similar to the work of \([10]\), we assume the Lambertian shading model \([14]\). Let the intensity image be treated as an intensity surface \( I \) defined by the vector function

\[
I(x, y) = [x, y, I(x, y)]^T
\]

(31)

Remind that the unit normal is \( \mathbf{N}(x, y) = \mathbf{N}(x, y)/\| \mathbf{N}(x, y) \| \) where \( \mathbf{N} \) is defined in \([14]\) and so, only depends on the depth \( Z(x, y) \) and its derivatives up to order-1.

We now want to detect features in the intensity surface \( I \) and check whether they are good candidates to be matched to curvilinear features detected in the depth surface \( D \), w.r.t. a given camera pose. The key issue here is that detected features in \( I \) can be matched to features detected in \( D \) on the condition that both are based on measurements with the same order of derivation in \( Z(x, y) \), in order to yield a “compatible” matching that ensures repeatability. The fact that \( I \) depends on \( Z(x, y) \) and its derivatives up to order-1, entails that the detection of features in \( I \) must rely on order-1 variations of the surface \( I(x, y) \), e.g., on its differential along some adequate direction.
Proof. The fundamental form matrix $\nabla \mu$ and $J$ in (20) is actually the Frobenius norm of the Jacobian matrix $J$ mentioned in [10]. The AVG feature defined in the difference of the eigenvalues of the first fundamental form matrix given in (34).

Let $\mu_1$ be the two eigenvalues of the first fundamental form matrix in the image plane. It can be easily seen that $\nabla I/\|\nabla I\|$ is the eigenvector of

$$
\left[\begin{array}{cc}
I_x & I_x \\
I_y & 1 + (I_y)^2
\end{array}\right]
= \nabla I
$$

(34)

associated with the largest eigenvalue $\mu_1$. It is worthy to note that the similarity between the expression of the principal values of the quadratic form $\|J\|$ and is compatible with our curvilinear measurements of the depth surface (i.e., with same order of the derivatives of $Z$).

In order to get a scalar measurement, we define the unit vectors $T_1 = \frac{\nabla I}{\|\nabla I\|}$ and $T_2$ by rotating $T_1$ by $\frac{\pi}{2}$. For $\alpha = 1, 2$, we also define

$$
|\mu_\alpha| = \|\text{d}I_Q(T_\alpha)\| 
$$

(33)

which is the differential of $I$ along unit direction $T_\alpha$ in the image plane. It can be easily seen that $\nabla I/\|\nabla I\|$ is the eigenvector of

$$
\frac{\|\text{d}I_Q(T_\alpha)\|}{\|\nabla I\|}
$$

(34)

this describes the second-order behaviour of the surface $M$ relative to the normal at one of its points in the immediate vicinity of this point. Using the result in (15), (16) and (17), we can claim that the extracted feature in the virtual image only depends on $X, Y, Z$ and their derivatives up to order-2. It is consistent (regarding the considered orders of the derivatives of $X, Y, Z$) with the feature $\|\nabla I\|^2 = I_x^2 + I_y^2$ detected in the intensity image, where $I_{\alpha}, \alpha \in \{1, 2\}$ is given in (32).

B. Multi-Curvilinear Saliency (MCS)

Multi-scale helps to detect important structures as well as small details. In consequence, in this paper, we compute the curvilinear saliency images in a multi-scale space. To build the scale pyramid, an edge-preserving smoothing approach, named anisotropic diffusion filter [47], is used in order to avoid oversmoothing. In fact, this filter tries to separate the low frequency components (i.e, sharp edges) from the high frequency components (i.e., textures) by preserving the largest edges in an image.

Contrary to depth images which represent textureless 3D shapes, intensity images are composed of shape and texture components. Consequently, the curvilinear saliency (CS) estimated from intensity images is affected by the textured regions. Our idea is to put forward the assumption that multi-scale analysis can discriminate between keypoints (those with high CS value in the image) due to shape and keypoints due to texture. At a coarse level, edges detected are reliable but with a poor localization and they miss small details. At a fine level, details are preserved, but detection suffers greatly from clutters in textured regions. In addition, the CS values of small details and textures are high in the coarse level, whereas these values become lower in the finest levels. To combine the strengths of each scale, the CS value of each pixel over $n$ scales is analyzed. If this value in all scales is higher than a threshold $T$, the maximum curvilinear saliency (MCS) value of this pixel over all scales is then kept. This threshold is a function of the number of the smoothed images, $n$, (i.e., $T = e^{-n}$; when $n$ is small, then $T$ is a big value and vice versa). However, if the CS value is lower than $T$ in one level, it is considered as a point that belongs to a texture (or a small detail) point, thus it is removed from the final multi-scale curvilinear saliency, MCS, image. Adding this multiscale step should help to reduce the impact of the texture on the point of
interest detection. However, in the next section, we propose to introduce the principle used for estimating focus map in order to increase the robustness to the background and to the presence of the texture.

VI. ROBUSTNESS TO BACKGROUND AND TO TEXTURE

Before introducing the proposed improvement, we briefly present existing works about texture detection and, in particular, about focus curve estimation.

A. Extraction of texture: state of the art

Various methods, such as [57, 58, 59, 60] have been proposed for extracting the texture from a natural image. In these approaches, a given image is separated into two components while preserving edges. In [57], Aujol et al. proposed a variational model based on total-variation (TV) energy for extracting the structural part. In [58], the authors proposed an algorithm in the field of scale space theory. This technique is a rolling guidance method based on an associated depth image to automatically refine the scale of the filtering in order to preserve edges. A structure-preserving image smoothing approach is introduced in [59]. This method locally analyzes the second order feature statistics of a patch around a pixel. The algorithm used a 7-dimensional feature vector that contains intensity, orientation and pixel coordinates. Finally, under a condition that the images contain smooth illumination changes and Lambertian surfaces, [60] proposed an intrinsic image decomposition model which explicitly determines a separate texture layer, as well as the shading layer and the reflectance layer. The method is based on surface normal vectors generated from an RGB-D image. All these works first smooth the intensity image as a pre-processing stage and then extracting the shape from that image relying on prior knowledge. And, to sum up, most of these methods for structure-texture decomposition are analogous to the classical signal processing low pass-high pass filter decomposition. However, even if it is correct to consider that the structure part of an image contains strong edges, the texture can also contain medium and high frequencies and the texture can only be partially removed. Another possibility is to consider focusness, related to the degree of focus.

Usually focusness is defined as inversely proportional to the degree of blur (blurriness) [52]. It is a very valuable tool for depth recovery [50] but also for blur magnification, or for image quality assessment. Blur is usually measured in regions containing edges, since edges would appear in images as blurred luminance transitions of unknown blur scale [43]. Then the estimation of the blur can be propagated to the rest of image. Since blur occurs for many different causes, this task is challenging and, in the literature, many methods on focus map estimation have been proposed. In [43], the authors identify blur as focal blur, induced by finite depth of field, as penumbral blur or shading blur and then estimate the blur scale. In the context of matting method [50], the blur ratio for every pixel corresponds to the ratio between the gradients of the input image and the re-blurred images. In [55], the authors use the K nearest neighbors (KNN) matting interpolation under the assumption that depth is locally uniform. However, blurring can also appear with edges caused by shadows and glossy highlights can also produce error in focus estimation. To remove errors induced by these other sources of blur, [50] used a cross bilateral filtering and estimation of sharpness bias and [52] used multi-scale. In the field of saliency detection, another way to estimate the amount of blur consists in computing the defocus blur between two Difference of Gaussian (DoG) images in multi-scale levels [52]. An optimization algorithm is used by minimizing the difference between the blurriness of a pixel and the weighted average of blurriness of neighboring pixels. In a Markov Random Field (MRF) formulation, a local contrast prior based on comparing local contrast and local gradient is also introduced in [51]. In [53], the authors propose to use the ratio between principle components, that is a weighted mixture of the spectral components. Moreover, the weights are proportional to the energy in the spectral component. Some algorithms also use the analysis of localized Fourier (Gabor filtering) spectrum [54]. In addition, smoothness constraints and image color edge information are taken into account to generate a defocus map for trying to preserve discontinuities on the transitions between objects.

Following all these aforementioned approaches, we can find that most of the existing algorithms [50, 52, 53] depend on measuring the blur amount using the ratio between the edges in two different scale levels (i.e., the original image and the re-blurred image). In consequence, we propose to use the ratio between the two curvilinear saliency images that contain robust edges in different scales to determine the blur amount based on the methods developed in [50]. For the multi-scale aspect, our approach is inspired by the principles explained in [52].

B. Removing background with focus curves: state of the art

Based on the mapping between the depth of a point light source and the focus level of its image, Shape From Defocus (SFD) approaches recover the 3D shape of a scene from focused images that represent the focus level of each point in the scene [48]. We can also notice that the focus (defocus) maps can be also used as an alternative for depth map, like in existing Adobe tools [50, 52, 54]. Consequently, it seems interesting to introduce what we call the detection of “focus curves” that capture blurriness in images. More precisely, focus curves mean that we estimate the scale of blur at the curvilinear saliency feature of the original image and we suppose that these features should be only related to discontinuities.

Focal blur occurs when a point is out of focus, as illustrated in Fig 5. When the point is at the focus distance, $d_f$, from the lens, all the rays from it converge to a sharp single sensor point. Otherwise, when $d \neq d_f$, these rays generate a blurred region in the sensor area. The blur pattern generated by this way is called the circle of confusion (CoC) whose diameter is denoted $c$.

In [49, 50, 52], the defocus blur can be modeled as a convolution of a sharp image with the point spread function (PSF) as shown in Fig. 6. The PSF is usually approximated...
by a Gaussian function \( g(x, \sigma) \), where the standard deviation \( \sigma \propto c \) measures the blur amount and is proportional to the diameter of the CoC:

\[
c = \frac{|d - df|}{d - f},
\]

where \( d, df, f \) are focus distance, defocus distance and focal length respectively as shown in Fig. 5. A blurred edge \( i(x) \) is then given by

\[
i(x) = f(x) \otimes g(x, \sigma)
\]

where \( f(x) = Au(x) + B \) is an ideal edge where \( u(x) \) is the step function. The terms \( A \) and \( B \) corresponds to the amplitude and the offset of the edge, respectively. Note that the edge is located at \( x = 0 \).

In [50], the blur estimation method was described for 1D case. The gradient of the re-blurred edge is:

\[
\nabla i_1(x) = \nabla (i(x) \otimes g(x, \sigma_0)) = \nabla (Au(x) + B) \otimes g(x, \sigma_0)
\]

\[
= \frac{A}{\sqrt{2\pi(\sigma + \sigma_0)}} \exp \left( -\frac{x^2}{2(\sigma^2 + \sigma_0^2)} \right)
\]

where \( \sigma_0 \) is the standard deviation of the blur Gaussian kernel. Thus, the gradient magnitude ratio between the original and the re-blurred edges is:

\[
R = \left| \frac{\nabla i_1(x)}{\nabla i_1(x)} \right| = \sqrt{\frac{\sigma^2 + \sigma_0^2}{\sigma^2}} \exp \left( -\frac{x^2}{2(\sigma^2 + \sigma_0^2)} \right) - \frac{x^2}{2(\sigma^2 + \sigma_0^2)} \right).
\]

It can be proved that the ratio is maximum at the edge location \( x = 0 \) and the maximum value is given by:

\[
R = \sqrt{\frac{\sigma^2 + \sigma_0^2}{\sigma^2}}
\]

Finally, given the maximum value \( R \) at the edge locations, the unknown blur amount \( s \) can be calculated using:

\[
s = \frac{\sigma_0}{\sqrt{R^2 - 1}}
\]

C. Focus curves based on Curvilinear Saliency – Multi Focus Curves (MFC)

We suppose that using focus can help to remove the background and using multiscale can help to reduce the influence of the texture in the same way as in section [V-B]. So, we propose to use the curvilinear saliency computation instead of the edge response to estimate the focus curves of an input image. In addition, we propose to estimate focus curves in multi-scales rather than in one scale as proposed in [50] to have scale invariant focus salient curves. In addition, we combine all information gotten from different blurring scales.

Assume the original pixel in an image is blurred as \( F(x, y) = I(x, y) \otimes g(x, y, \sigma) \). Thus to get the curvilinear saliency, the structure tensor is calculated as:

\[
S_T = f(\nabla ((I(x, y) \otimes g(x, y, \sigma))))
\]

\[
= \left( \begin{array}{cc}
I_x^2 & I_x I_y \\
I_x I_y & I_y^2
\end{array} \right) \otimes g(x, y, \sigma)
\]

(43)

If the Hessian matrix is expressed with eigenvectors and eigenvalues, we obtained:

\[
H = \left( \begin{array}{cc}
e_1 & e_2 \\
0 & \lambda_1
\end{array} \right) \left( \begin{array}{c}
e_1^T \\
\lambda_2\end{array} \right) \otimes g(x, y, \sigma)
\]

(44)

The curvilinear saliency can be described as:

\[
CS = (\lambda_1 - \lambda_2) \otimes g(x, y, \sigma)
\]

(45)

In particular, the curvilinear saliency can be directly computed as:

\[
CS = \alpha((I_x^2 + I_y^2)) \otimes g(x, y, \sigma)
\]

(46)

The re-blurred curvilinear saliency image, named \( CS_i \), in multi-scale can then be defined as:

\[
CS_i = \alpha((I_x^2 + I_y^2)) \otimes g(x, y, \sigma) \otimes g(x, y, \sigma_i), \; i = 1, 2, ..., n
\]

(47)

where \( n \) is the number of scales.

Consequently, the ratio between the original and re-blurred curvilinear saliency is:

\[
R_i = \frac{CS_i}{CS} = \frac{\sigma^2 + \sigma_i^2}{\sigma^2} \exp \left( -\frac{x^2 + y^2}{2(\sigma^2)} - \frac{x^2 + y^2}{2(\sigma^2 + \sigma_i^2)} \right)
\]

(48)

Within the neighborhood of an pixel, the response reaches its maximum when \( x = 0 \) and \( y = 0 \), thus:

\[
R_i |_{0} = \frac{CS_i}{CS} = \frac{\sigma^2 + \sigma_i^2}{\sigma^2} = 1 + \frac{\sigma_i^2}{\sigma^2}
\]

(49)

Finally, given the maximum value \( R_i \) in each scale level, the unknown blur amount \( s_i \) can be calculated using:

\[
s_i = \frac{\sigma_i}{\sqrt{R_i |_{0} - 1}},
\]

(50)

For \( n \) scales, we compute \( n - 1 \) focus curve scales by using the ratio between curvilinear saliency of the coarse level (i.e., the original image) and the next scale levels. By following
the same remarks as in section \[V-B\] we define Multi Focus Curves (MFC) that correspond to the fusion of all the focus curves into one map by keeping only the pixels that have a high focus value in all the \(n-1\) scales (i.e., a high value means a value bigger than \(T = e^{-n}\), chosen in the same way as section \(V-B\)). If the pixel has a high value at all scales, the maximum value of the scale of blur is taken into account to build the final multi-scale curve map:

\[
MFC = \frac{1}{\arg \max_i (s_i)}.
\]  

In conclusion, the highest values of the estimated \(MFC\) indicate edges that have low blurring (i.e., sharp edges). On the contrary, low values indicate edges that have high blurring. Consequently, we expect that focus curves highlight curvilinear saliency in images that are approximately similar to the detected curvilinear saliency features in depth images.

**VII. EXPERIMENTS FOR FEATURE DETECTION**

**A. Comparison with existing methods**

One of our most important objectives in this work was to introduce a detector that is more repeatable between 2D images and 3D models than classical detectors in the literature. In consequence, we compare the features detected on 3D models with the proposed curvilinear saliency detector with features detected on real images with these three 2D detectors: Image Gradient (IG), Multi-scale Curvilinear Saliency (MCS) and Multi-scale Focus Curves (MFC). In addition, we measure the repeatability between the two others 3D model detectors, i.e., Average Shading Gradient (ASG) \([10]\) and Hessian Frobenius Norm (HFN), and the same three 2D detectors. And then, we compared MFC and MCS with nine classical 2D detectors:

1) **Edge detectors**: (i) Sobel, (ii) Laplacian of Gaussian (Log), (iii) Canny [15] and (iv) Fuzzy logic technique [40];

2) **Corner detectors**: (v) Harris detector based on autocorrelation analysis and (vi) Minimum Eigenvalues detector based on analysis of the Hessian matrix [46];

3) **Multi-scale detectors**: (vii) SIFT, Scale Invariant Feature Transform [19], that uses the analysis of difference of Gaussian, (viii) SURF, Speeded Up Robust Features, a multi-scale technique based on the Hessian matrix [23] and (ix) a multi-scale Principal Curvature Image (PCI) detector [24].

**B. Evaluation criteria**

The eleven 2D detectors are evaluated with two measures:

1) Intersection percentage (IP): the probability that a 2D intensity-based key feature can be found close to those extracted in a depth image [42].

2) Hausdorff Distance (HD): the classical measurement is defined for two point sets \(A\) and \(B\) by:

\[
HD(A, B) = \max(h(A, B), h(B, A)),
\]

where \(h(A, B) = \max_{a \in A} \min_{b \in B} \| a - b \|\). The lowest the distance, the most similar the two sets.

**C. Datasets**

Two datasets are evaluated:

1) **Web collection**: we have collected 10 objects and 15 real images of each object on the web by choosing views as close as possible to the views used for the generation of the depth images. Moreover, to highlight the robustness of the approach to different acquisition conditions, many real images of a similar model are taken.

2) **PASCAL3D+ dataset [61]**: it is used in order to assess scalability, it contains real images corresponding to 12 rigid objects categories. We have computed average results for all non occluded objects in each category, i.e. around 1000 real images and 3 or more reference models per category. The real images are acquired under different acquisition conditions (e.g., lighting, complex background, low contrast). We have rendered the depth images of the corresponding 3D CAD model using the viewpoint information from the dataset. Only non-occluded and non-truncated objects in the real images were used. Furthermore, we choose 3D textureless objects (available online [3]).

For all the tested 3D models, we have rendered depth images using MATLAB 3D Model Renderer [1].

**D. Analysis of the results**

As shown in tables [3] and [4] and as expected, the proposed approach using focus curves based on curvilinear saliency,
is presented. These results highlight that image gradients are
happier) of results obtained for the PASCAL3D+ dataset [61] are
greater than the state of the art.

In the rest of this section, we illustrate the results for
the most significant dataset, the PASCAL3D+, in order
to avoid redundancy in the explanations. In Fig. 7, two examples
of results obtained for the PASCAL3D+ dataset [61] are
given. More precisely, the repeatability percentage between
the three comparable 3D detectors, i.e. MFC, MSC and
Image Gradient (IG), and the three comparable 2D detectors,
Hessian Frobenius Norm, Average Shading Gradient and CS,
is presented. These results highlight that image gradients are
affected by texture. Moreover, MCS improves the repeatability

| Methods | Car | Shoe | Plane | T-Rex | Elephant | Phydant | Jeep | Mug | Teddy | Pistol |
|---------|-----|------|-------|-------|----------|---------|------|-----|-------|--------|
| MFC     | 59  | 38   | 35    | 66    | 37       | 56      | 69   | 57  | 44    | 69     |
| MCS     | 50  | 31   | 58    | 59    | 32       | 42      | 57   | 54  | 39    | 67     |
| PCI     | 46  | 51   | 35    | 19    | 26       | 42      | 66   | 50  | 32    | 63     |
| MinEig  | 08  | 02   | 06    | 09    | 03       | 06      | 05   | 03  | 04    | 09     |
| Harris  | 04  | 10   | 04    | 06    | 02       | 04      | 05   | 05  | 05    | 05     |
| SIFT    | 03  | 01   | 03    | 05    | 03       | 02      | 05   | 03  | 04    | 09     |
| SURF    | 10  | 04   | 13    | 12    | 14       | 13      | 15   | 11  | 10    | 14     |
| Log     | 18  | 10   | 11    | 20    | 13       | 10      | 15   | 12  | 08    | 08     |
| Fuzzy   | 11  | 05   | 11    | 20    | 13       | 11      | 14   | 11  | 08    | 08     |

| Methods | Car | Shoe | Plane | T-Rex | Elephant | Phydant | Jeep | Mug | Teddy | Pistol |
|---------|-----|------|-------|-------|----------|---------|------|-----|-------|--------|
| MFC     | 55  | 69   | 48    | 47    | 71       | 84      | 101  | 85  | 85    | 85     |
| MCS     | 37  | 51   | 59    | 58    | 50       | 48      | 99   | 54  | 99    | 54     |
| PCI     | 15  | 15   | 15    | 15    | 15       | 15      | 15   | 15  | 15    | 15     |
| MinEig  | 13  | 13   | 13    | 13    | 13       | 13      | 13   | 13  | 13    | 13     |
| Harris  | 15  | 15   | 15    | 15    | 15       | 15      | 15   | 15  | 15    | 15     |
| SIFT    | 13  | 13   | 13    | 13    | 13       | 13      | 13   | 13  | 13    | 13     |
| SURF    | 13  | 13   | 13    | 13    | 13       | 13      | 13   | 13  | 13    | 13     |
| Log     | 13  | 13   | 13    | 13    | 13       | 13      | 13   | 13  | 13    | 13     |
| Fuzzy   | 13  | 13   | 13    | 13    | 13       | 13      | 13   | 13  | 13    | 13     |

Table III: Mean Hausdorff Distance (HD) (lower is better) of all depth images rendered from different viewpoints and all real images captured under different textures and lighting for the web collection with the proposed method (MFC), the method (MCS) and 9 tested detectors.

Table IV: Mean Hausdorff Distance (HD) (lower is better) of all depth images rendered from different viewpoints and all real images captured under different textures and lighting for the PASCAL3D+ with the proposed method (MFC), the method (MCS) and 9 tested detectors.

In addition, as shown in [III and [IV the average Hausdorff Distance (HD) with MFC is less than 24 and with MCS, less than 32. On the contrary, the other detectors do not reach high repeatability scores.

All these quantitative results support that MFC is able to
detect curvilinear saliency features that are more repeatable
between an intensity image and its corresponding depth image
than the state of the art.

In this section, we register a 2D query image to a 3D
model by finding the closest view \( d \) between all the rendered
images of the 3D model \( d_k, k = 1 \ldots N \), where \( N \) the
number of rendered views (i.e., depth images). We suppose
that the object to recognize is contained in a bounding box
and we want to estimate the 3D pose. Estimating the pose
consists in estimating the elevation and the azimuth angles,
respectively \( (h) \) and \( (r) \), and the distance between the model
and the camera, \( (v) \). For each 3D model, we generate depth
images from near uniformly distributed viewing angles around
a sphere by changing \( h, a \) and \( v \) to have \( N \) views per model.

The choices for these terms are explained in paragraph [IX-A
To describe an object in a photograph and in all the rendered depth images, we naturally expand the famous classical HOG, Histogram Of Gradient, widely used in the literature [44, 10], to work on curvilinear saliency. HOG is used in a sliding window fashion in an image to generate dense features based on binning the gradient orientation over a spatial region. Indeed, both in rendered depth images and in photographs, the orientation of the curvature and the magnitude of the curvilinear saliency is used for building the descriptors. For depth image representation, we multiply $CS$ by the eigenvector $e_{H_1}$ corresponding to the largest eigenvalue of the matrix $M$ in (26):

$$\overrightarrow{CS} = CS, e_{H_1}.$$

In turn, for photographs, for $MCS$, we multiply $MCS$ values by the eigenvector $e_{S_1}$ corresponding to the curvilinear saliency $\lambda_1 - \lambda_2$, as shown in (56):

$$\overrightarrow{MCS} = MCS, e_{H_1}.$$

In addition, for $MFC$, we also multiply $MFC$ values with the eigenvector $e_{S_1}$:

$$\overrightarrow{MFC} = MFC, e_{S_1}.$$

Using the same principle of HOG presented in [14], we propose a descriptor that contains the orientation of the curvature and the value of curvilinear saliency and the magnitude of $CS$, $MCS$ and $MFC$ for an image are binned into sparse per-pixel histograms.

Given the HOG descriptor from a 2D query image $D_q$, we compute the HOG descriptors of the rendered images $D_{dN}$, with $N$ rendering depth images. In order to compare $D_q$ to every $D_{dN}$, the similarity scores are computed as proposed in [44]:

$$S_{hog}(k, h, a, v) = \left( D_{dN} - \mu_s \right)^T \Sigma^{-1} D_q,$$

where, $k = 1...N$, $\Sigma$ and $\mu_s$ are, respectively, the covariance matrix and the mean over all descriptors of the rendered images. For the registration process, evaluating $S_{hog}(k, h, a, v)$ can be done by computing the probability of the inverse of the inner product between $D_q$ and a transformed set of descriptors. The $S_{hog}(k, h, a, v)$ probability is then maximized in order to find the closest corresponding views of the query image.

We also evaluate a global similarity by measuring how well each individual detected point in an image is able to be matched with a corresponding detected point in the depth map, i.e., how well each image detected points are repeatable. More precisely, this repeatability scores, $Rep$, normalized between 0 and 1, is the probability that key features in the intensity image are found close to those extracted in the depth image $Rep_{dN}$. Since the closest view should have a high repeatability scores in comparison to other views, the dissimilarity based on repeatability scores is defined by $R_{d_i} = 1 - Rep_{d_i}$. If we denote $R_{d_i}$ the repeatability...
scores of $N$ rendered views of a model and a given image,  
the similarity $S_{\text{rep}}$ is defined by:  
\[ S_{\text{rep}}(k, h, a, v) = \exp\left(-\frac{(\mathbf{R}_{d_k} - \mu_r)^2}{2\sigma_r^2}\right). \]  
(54)  
where $\mu_r$ is the mean value of $\mathbf{R}_{d_k}$ and $\sigma_r$ is the standard deviation (i.e., in this work $\sigma_r = 0.1$).

Finally, by combining all HOG feature similarities and the similarity based on the repeatability, the probability of the final similarity is given by:
\[ S(m, h, a, v) = S_{\text{hog}}(k, h, a, v) \odot S_{\text{rep}}(k, h, a, v). \]  
(55)  
where $\odot$ is the Hadamard product. Based on calculating $S(k, h, a, v)$, we select at least the highest three correspondences to estimate the full pose. From the selected three views, the logically ordered or connected views (i.e., coherent views) are firstly selected. We then find minimum and maximum values of $h$, $a$ and $v$ of the corresponding views. Additional views are then generated in the vicinity of the selected views that are between the minimum and maximum values of the three parameters with small steps (e.g., $\delta h = 5^\circ$, $\delta a = 5^\circ$ and $\delta v = 5cm$). The process is again repeated for these ranges to find the closest view to the object in a query image until convergence. Assume the ground-truth transformation matrix ($\mathbf{T}_g$) containing rotation ($\mathbf{R}_g$) and translation ($\mathbf{t}_g$) matrices:
\[ \mathbf{T}_g = \begin{bmatrix} \mathbf{R}_g & \mathbf{t}_g \\ 0 & 1 \end{bmatrix}. \]  
In addition, the estimated transformation matrix ($\mathbf{T}_e$) containing rotation ($\mathbf{R}_e$) and translation ($\mathbf{t}_e$) matrices:
\[ \mathbf{T}_e = \begin{bmatrix} \mathbf{R}_e & \mathbf{t}_e \\ 0 & 1 \end{bmatrix}. \]  
The error between the two transformation matrix is:
\[ E = \|M_i - M_j\|, \]  
where $i \neq j$. Thus, the convergence criterion is $E \leq \varepsilon$, $\varepsilon$ is a very small value (in this work, $\varepsilon = 0.05$).

IX. EXPERIMENTS FOR POSE ESTIMATION

A. 3D models representation and alignment

Matching photographs and rendered depth images requires a complete 3D model representation. Each depth image represents a 3D model from different viewpoints. Actually, we need to have a large number of depth images to completely represent a 3D model. However, this yields a massive execution time. Consequently, we have orthographically rendered $N$ depth images (around 700 in our experiments) from approximately uniformly distributed viewing angles $h$ and $a$ and the distance $v$ (i.e., in these experiments, $h$ is increased by a step of $50^\circ$, the azimuth angle, $20^\circ$, and the distance, $0.3$ m, for a range between $0$ and $2$ m).

In addition, we need a parametrization of the alignment of the model view in a depth image with the object detected in a color image. Consequently, when we compare two models, we need to compute the optimal measure of similarity, over all possible poses. To do this, each model is placed into a canonical coordinate frame, normalizing for translation and rotation. Since we know the centroid of them, the models are normalized for translation by shifting them so that the center of mass is aligned with the origin. Next, the two models are normalized for rotation by aligning the principal axes of the model with the x- and y-axes. It defines the ellipsoid that best fits the model. By rotating the two sets of points so that the major axis of the ellipsoid is aligned with the x-axis, and the second major axis aligns with the y-axes, we obtain a model in a normalized coordinate frame. We use Principle Components Analysis, PCA, to find the orientation of the major axis of the ellipsoid. The set of points of the model is rotated by the difference of the direction of the two major axes. After normalization, the two models are (near) optimally aligned and can be directly compared in their normalized poses.

In addition, in this paper, the HOG descriptor is quantized into 9 bins, exactly as proposed in [14]. The photograph and each depth image is divided into a grid of square cells (i.e., in this work, the image is divided into $8 \times 8$). For each cell, the curvilinear saliency, focus curves or image gradients, histograms are aggregated by weighting them with their respective magnitude.

B. Analysis of the results

![Fig. 9: Repeatability scores of an image with 40 rendered views around the correct view. The correct view is the one in the middle of the axis.](image)

For pose estimation or even for object recognition, it is needed that the probability that key features in the photograph are found close to those extracted in the depth image must be high when the photograph and the depth image come from the same viewpoint. In the first experiment that is presented, we try to illustrate that the correct view provides the highest repeatability score, in particular with the proposed detector. In consequence, the repeatability scores of features extracted on photograph and depth images with MCS, SIFT and Sobel edge detector, for the correct view and the other views around it, were calculated and shown in Fig. 9 with one example of the dataset. As shown, the three detectors yield the highest repeatability score with the correct viewpoint (even if the difference between views is small, like with Sobel). In addition, different grids were tested: $4 \times 4$, $8 \times 8$ and $16 \times 16$. The grid with $8 \times 8$ size yields the best precision rate.
as expected, the score is gradually diminished whenever it is far from the correct viewpoint. The most important result is that MCS gives the highest differences between the correct view and all the other views, so, it illustrates that it is the most adapted detector for pose estimation based on 2D/3D registration. This result is quite coherent because SIFT was designed to be robust against many changes, many difficulties and it induces that the differences should be lower than MCS that is designed to be efficient in the case of 2D/3D matching.

In addition, the second experiment was performed with the Pascal+3D dataset. For each category of objects, we compute the precision rate for detecting the correct view after using the three aforementioned methods for 3D model representations, i.e., Curvilinear Saliency (CS), Average Shading Gradient (ASG) and Apparent Ridges (AR) [30], against the three techniques for intensity image representation, i.e., Image Gradient(IG), Multi Curvilinear Saliency (MCS) and Multi Focus Curves (MFC). As shown in table VI, the registration between our Curvilinear Saliency (CS) representation of the 3D model and the multiscale focus curves (MFC) extracted on corresponding images outperforms all other variations of the tested methods. This confirms that curvilinear saliency representation computed from the depth images of a 3D model can capture the discontinuities of surfaces. In addition, MFC can reduce the influence of texture and background components by extracting the edges related to the object shape in intensity images rather than MCS. Furthermore, the precision rate is reduced by more than 25% compared with ASG and IG to represent 3D models and intensity any other image representation. Apparent ridge rendering yields the smallest registration accuracy with the three image representations among all the 3D model representation techniques. However, using ASG with untextured 3D models against MFC and MCS also increases correct pose estimation rate. All these results indicate that average shading gradients computed from the normal map of an untextured geometry is a good rendering technique for the untextured geometry. However, image gradients are not the good representation of intensity images in order to match with rendering images because it is affected by image textures. All these results are confirmed by the third experiment, in table VI, where the details are given about the precision of the pose estimation in terms of elevation, azimuth, yaw angles and distance.

![Table VI: Average Error of the estimated (Est.) (a) elevation, (b) azimuth and (c) yaw angles and (d) distance, in centimeter, of the pose of the camera. The term Clo. indicates the closest view to the correct pose. These quantitative results demonstrate that the best combination is MFC/CS.](image)

In the next experiment, in Fig. 10, we show the precision of the registration of images among the top r similarities, i.e. we sort all the similarity scores obtained for all the views, and we analyse the r first highest similarities (more precisely, the 1, 3, 5, 10 and 20 first ranks). The correct pose is searched within this set of views. As shown, the precision rate increased when the number of views is increased, for any combination of 3D model representation and image representation. However, MFC yields the highest precision with the three tested methods of representing 3D models (i.e., CS, ASG, and AR). In addition, MCS yields good precision values. In fact, IG yields the smallest precision values because the edges detected with texture information have a bad influence on estimating the successful registration.

Finally, the Fig. [11] shows some examples of the Pascal+3D dataset of correct registrations with the top-ranked pose estimation. It can be seen that our system is able to register an image with a great variety of texture and viewing angle. In addition, the proposed algorithm can register images regardless of light changes in images.

**X. CONCLUSION AND PERSPECTIVE**

After an analysis of existing tools for 2D/3D registration, the major goal of this paper was to propose an approach for 2D/3D matching more adapted, and in particular more justified, than existing approaches. For that purpose, we also proposed an evaluation protocol based on repeatability study. More precisely, for doing this matching, we have studied these two important aspects: how to represent the studied data, in 2D and in 3D, and then, how to compare them. In this context, we introduce a 3D detector based on curvilinear saliency and a 2D detector based on the same principle but adapted in multiscale and combined with the principle of focus curves. The interest of this new method was also illustrated by quantitative evaluation on pose estimation and 2D/3D registration. All the results are very encouraging and the next step of this work is to use this registration to identify defaults on objects. For that purpose, we need to study the robustness of this work to missing part of objects and to adapt the registration process in consequence.

**REFERENCES**

[1] P. Markelj and D. Tomazevič and B. Likar and F. Permuš. A review of 3D/2D registration methods for image-guided
Fig. 10: Precision values with different ranks with image representation using (a) Image Gradient (IG), (b) Multiscale Curvilinear Saliency (MCS) and (c) Multiscale Focus Curves (MFC).

interventions. Medical Image Analysis, 16(3):642–661, 2012.
[2] F. Pomerleau and F. Colas and S., Roland. A Review of Point Cloud Registration Algorithms for Mobile Robotics. Found. Trends Robot, 4(1):1–104, 2015.
[3] A. Krizhevsky and I. Sutskever and G. E. Hinton. ImageNet Classification with Deep Convolutional Neural Networks. In International Conference on Neural Information Processing Systems. 2012.
[4] H. Su and C. R. Qi and Y. Li and L. J. Guibas. Render for CNN: Viewpoint Estimation in Images Using CNNs Trained With Rendered 3D Model Views. In IEEE International Conference on Computer Vision. 2015.
[5] M. Dellepiane and M. Callieri and F. Ponchio and R. Scopigno. Mapping highly detailed color information on extremely dense 3D models: the case of David’s restoration, In Eurographics Conference – Cultural Heritage Papers.
[6] C. Wu and B. Clipp and X. Li and J. M. Frahm and M. Pollefeys. 3D model matching with Viewpoint-Invariant Patches (VIP). In IEEE Conference on Computer Vision and Pattern Recognition. 2008.
[7] S. Agarwal and Y. Furukawa and N. Snavely and I. Simon and B. Curless and S. Seitz and R. Szeliski. Building Rome in a Day. Communications of the ACM, 54(10):105–112, 2011.
[8] R. J. Campbell and P. J. Flynn. A Survey Of Free-Form Object Representation and Recognition Techniques. Computer Vision and Image Understanding, 81(2):166–210, 2001.
[9] C. B. Choy and M. Stark and S. Corbett-Davies and S. Savarese. Enriching object detection with 2D-3D registration and continuous viewpoint estimation. In IEEE Conference on Computer Vision and Pattern Recognition. 2015.
[10] T. Plötz and S. Roth. Automatic Registration of Images to Untextured Geometry Using Average Shading Gradients. International Journal of Computer Vision, 125(1):65–81, 2017.
[11] A. Irschara and C. Zach and J. M. Frahm and H. Bischof. From structure-from-motion point clouds to fast location recognition. In IEEE International Conference on Computer Vision. 2009.
[12] D. Scharstein and R. Szeliski, Richard. A Taxonomy and Evaluation of Dense Two-Frame Stereo Correspondence Algorithms. International Journal of Computer Vision, 47(1):7–42,2002.
[13] R. Szeliski. Computer Vision – Algorithms and Applications. Springer, 2011.
[14] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In IEEE Conference on Computer Vision and Pattern Recognition. 2005.
[15] J. Canny. A Computational Approach to Edge Detection. IEEE Transactions on Pattern Analysis and Machine Intelligence, 8(6):679–698, 1986.
[16] C. Harris and M. Stephens. A combined corner and edge detector. In Alvey Vision Conference, 1988.
[17] P. Fischer and T. Brox. Image Descriptors Based on Curvature Histograms. In German Conference on Pattern Recognition. 2014
[18] S. Smith and J. Brady. SUSAN - A New Approach to Low Level Image Processing. International Journal of Computer Vision, 23(1):45–78, 1997.
[19] D. G. Lowe. Distinctive image features from scale-invariant keypoints. International Journal of Computer Vision, 60(2):91–110, 2004.
[20] J. Matas and O. Chum and U. Martin and T. Pajdla. Robust wide baseline stereo from maximally stable extremal regions. In British Machine Vision Conference. 2002.
[21] T. Tuytelaars and L. Van Gool. Matching Widely Separated Views Based on Affine Invariant Regions. International Journal of Computer Vision, 59(1):61–85, 2004.
[22] K. Mikolajczyk and C. Schmid. Scale & Affine Invariant Interest Point Detectors. International Journal of Computer Vision, 60(1):63–86, 2004.
[23] H. Bay and A. Ess and T. Tuytelaars and L. Van Gool. Speeded-Up Robust Features (SURF). IEEE Computer Vision and Image Understanding, 110(3):346–359, 2008.
[24] H. Deng and W. Zhang and E. Mortensen and T. Dietterich and L. Shapiro. Principal Curvature-Based Region Detector for Object Recognition. IEEE Computer Vision and Image Understanding, 2007.
Fig. 11: Some correct registration examples with the Pascal+3D dataset. We show the query image (column 1), the corresponding 3D model (column 2) and the first ranked pose estimation (column 3). It illustrates that even if the 3D model does not have the same detailed shape, the registration can be correctly done.

[25] C. H. Lee and A. Varshney and D. W. Jacobs. Mesh Saliency. ACM transactions on graphics, 24(3):659–666, 2005.
[26] T. Sattler and B. Leibe and L. Kobbelt. Fast image-based localization using direct 2D-to-3D matching. In IEEE International Conference on Computer Vision. 2011.
[27] Y. Ohtake and A. Belyaev and H.-P. Seidel. Ridge-valley Lines on Meshes via Implicit Surface Fitting. ACM transactions on graphics, 23(3):609–612, 2004.
[28] Y. Lee and L. Markosian and S. Lee and J. F. Hughes. Line Drawings via Abstracted Shading. ACM transactions on graphics, 26(3), 2007.
[29] Y. Y. Lee and K. Park and J. D. Yoo and K. H. Lee. Multi-scale Feature Matching Between 2D Image and 3D Model. In SIGGRAPH Asia. 2013.
[30] T. Judd, F. Durand, and E. H. Adelson. Apparent ridges for line drawing. ACM transactions on graphics, 26(3), 2007.
[31] A. Godil and A. I. Wagan. Salient local 3D features for 3D shape retrieval. In SPIE. 2011.
[32] D. P. Paudel and C. Demonceaux and A. Habed and P. Vasseur. Localization of 2D Cameras in a Known Environment Using Direct 2D-3D Registration. In IEEE Conference on Pattern Recognition, 2014.
[33] C. Xu and L. Zhang and L. Cheng and R. Koch. Pose Estimation from Line Correspondences: A Complete Analysis and a Series of Solutions. IEEE Transactions on Pattern Analysis and Machine Intelligence, 39(6):1209–1222, 2017.
[34] M. Tamaazousti and V. Gay-Bellile and S. N. Collette and S. Bourgeois and M. 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. 2011.
[35] L. Liu and I. Stamos. Automatic 3D to 2D registration for the photorealistic rendering of urban scenes. In IEEE Conference on Computer Vision and Pattern Recognition. 2005.
[36] S. Ramalingam and S. Bouaziz and P. Sturm and M. Brand. Geolocalization using skylines from omni-images. In IEEE International Conference on Computer Vision Workshops, 2009.
[37] B. Parlett. The Symmetric Eigenvalue ProblemSociety for Industrial and Applied Mathematics (SIAM), 1998.
[38] J.-H. Gallier. Geometric Methods and Applications: For Computer Science and Engineering, (Texts in applied mathematics). Springer. 2001.
By computing $\mathbf{I}_x = (1, 0, I_x)^\top$ and $\mathbf{I}_y = (0, 1, I_y)^\top$, it can be easily shown that

$$J_1^\top J_1 = \begin{bmatrix} \mathbf{I}_x \cdot \mathbf{I}_x & \mathbf{I}_x \cdot \mathbf{I}_y \\ \mathbf{I}_x \cdot \mathbf{I}_y & \mathbf{I}_y \cdot \mathbf{I}_y \end{bmatrix} = \begin{bmatrix} 1 + (I_x)^2 & I_x I_y \\ I_x I_y & 1 + (I_y)^2 \end{bmatrix} = I + \nabla_I \nabla_J$$

(56)