Available online at www.sciencedirect.com
SCIENCE DIRECT®

ELSEVIER

Computer Vision and Image Understanding 97 (2005) 1–29

Computer Vision and Image Understanding

www.elsevier.com/locate/cviu

Scene-consistent detection of feature points in video sequences

Ariel Tankus*, Yehezkel Yeshurun

School of Computer Science, Tel-Aviv University, Tel-Aviv 69978, Israel

Received 21 January 2003; accepted 3 May 2004
Available online 18 October 2004

Abstract

Detection of feature points in images is an important preprocessing stage for many algorithms in Computer Vision. We address the problem of detection of feature points in video sequences of 3D scenes, which could be mainly used for obtaining scene correspondence. The main feature we use is the zero crossing of the intensity gradient argument. We analytically show that this local feature corresponds to specific constraints on the local 3D geometry of the scene, thus ensuring that the detected points are based on real 3D features. We present a robust algorithm that tracks the detected points along a video sequence, and suggest some criteria for quantitative evaluation of such algorithms. These criteria serve in a comparison of the suggested operator with four other feature trackers. The suggested criteria are generic and could serve other researchers as well for performance evaluation of stable point detectors.

© 2004 Elsevier Inc. All rights reserved.

Index Terms: Feature point detection; Scene-consistent detection; Stable point tracking; Tracking evaluation

1. Introduction

Context-free detection of specific image points ("features") is being addressed in Computer Vision for a long time, as it is the basis of many higher level algorithms of

* Corresponding author.
E-mail addresses: arielt@post.tau.ac.il (A. Tankus), hezy@post.tau.ac.il (Y. Yeshurun).

1077-3142/$ - see front matter © 2004 Elsevier Inc. All rights reserved.
doi:10.1016/j.cviu.2004.05.003

visual information processing. Works in this field divide up into two categories: the first category is "Attentional," defining the detected points as those attracting computational resources; this is apparently the case in biological systems. Other names for this category are detection of "Interest Points" or "Regions of Interest" (RoI). The second approach defines the task as consistent selection of a subset of image pixels, regardless of their "attentional" value. This approach is known also as: "Anchor Points" or "Stable Points" detection. Such points could either be used for object recognition [1], or as correspondence points for recovering 3D characteristics of the scene.

The main goal of this paper is *robust detection of scene-consistent feature points in video sequences*, and as such it takes the second approach. By robust we mean that the algorithm should consistently detect points in noisy images, and by scene-consistent (or: stable), that the algorithm should consistently detect the same 3D point over multiple video frames, regardless of illumination changes, pose variations, camera motion, or parallax. This implies that detection should depend merely on local geometry of scene objects.

The literature of feature detection has a wide variety of detectors based on edges. Edges in many cases are not intrinsic to one subject, but rather delineate a boundary between a subject and the background. Therefore, edge-based features in many cases depend on the background and viewing direction (e.g., silhouette of a person in upright position vs. profile). These inherent flaws of edge-based features raise a need for non-edge-based feature detectors.

This paper presents an operator for non-edge-based feature detection. The features are extrema of the intensity function, but only in smooth image domains. Pay attention, that an edge (e.g., a black line on a white paper) would *not* yield a strong response of the suggested operator, because the intensity function is discontinuous there (i.e., not a smooth domain). The uniqueness of the suggested approach lies in its ability to detect local extrema of smooth image domains in a reliable and robust way, while avoiding local extrema created by edges. As the paper would present, this robustness has a solid theoretic basis; a large number of images would demonstrate that the robustness is attained in real-life scenes as well.

The contribution of this paper divides into two:

1. The theoretic point of view: The paper maps the image domains which the operator detects, by their differential–geometric features. Using this mapping, we characterize the 3D scene locations projected onto the detected image domains. The theorems we present explain why the detection of the operator is robust, and why it consistently follows 3D points in the scene.
2. The practical viewpoint: Our paper focuses on the *detection* of features in video sequences. However, when evaluating fitness of features to real-life tasks, it is highly nontrivial to decouple detection from tracking. Hence, we examine the detection–tracking system as a whole ([2] takes a similar approach), and define measures of evaluation of such systems. This approach can thus evaluate how appropriate a selected group of features is for higher levels of processing. Using

these measures, we show that our method favorably compares with four other methods (Kanade-Lucas-Tomasi [2], Junction Detection [3], ImpHarris [4] and Harris-Laplacian [5]). In our comparison we have used a common tracker for all detection methods except for that of Kanade-Lucas-Tomasi, where the detector has a built-in tracker.

The paper is structured as follows: Section 2 surveys the literature of feature detectors and trackers as well as that of evaluation criteria. Section 3 sketches the operator for detection of RoI in static images, that was suggested in [6]. Section 4 then analytically characterizes the specific features of the image intensity function $I(x,y)$ which the suggested method detects, and proves that these image-space features correspond to the local 3D geometry of objects in the scene. This section establishes the theoretical basis explaining why the operator is very robust. Section 5 presents a simple algorithm, based on Kalman filter, that robustly tracks these features in video sequences. The usage of video sequences confronts the operator with new effects which do not exist in static images: parallax, camera motion and 3D object transformations. The operator copes well with these effects, because it responds to intrinsic properties of 3D objects (as is proved in Section 4). In Section 6, we rigorously define two measures for evaluating detection–tracking systems: completeness (with respect to correct tracking of 3D points) and stability. These measures are generic and could be of use for other researchers as well. The measures serve in a comparison between the suggested tracker and four other trackers (Section 7). Section 8 suggests a way to further increase the aptness of evaluation measures to higher level tasks. Following concluding remarks in Section 9, Appendices A and B supply the complete proofs of the theorems of Section 4.

2. Literature survey

This section briefly surveys the recent work in the fields of feature point detection, point tracking, and evaluation of feature points. We do not present a complete survey of feature point detection or tracking, because the literature of these fields is very rich and its description is beyond the scope of this paper. In addition, two such surveys were recently published: [4,7].

2.1. Feature detection

Typical examples for the "Attentional" approach to feature detection can be found in [7–9]. One type of scene-consistent methods is corner detectors: [10–13]. Lowe [14] uses extrema of a Difference of Gaussian function applied to scale space to detect stable points.

Three other feature detectors are of particular interest for our discussion, as we compare our method with them. The first is Junction Detection [3], where automatic scale selection associates a Region of Interest with each detected junction. Junctions are detected according to the curvature of the level curves of the intensity function, multiplied by the gradient magnitude raised to the power of three. The second is the Harris operator [15] (also known as the Plessey feature point detector) which is based on the variation of local auto-correlation over different orientations. It is calculated by functions related to the principle curvatures of the local auto-correlation. Schmid et al. [4] uses an improved version of the Harris operator (marked: ImpHarris) in an extensive comparison of feature detectors. The improvement is replacement of the method of differentiation from mask-based to Gaussian-based. The third algorithm is the Harris-Laplacian operator [5], which combines the Harris function with a scale-space representation. It first utilizes the Harris detector for 2D localization of interest points at each level of the scale-space. It then uses the Laplacian to test whether an interest point forms a maximum in the scale direction.

2.2. Feature tracking

One approach to feature tracking is estimation of motion and deformations of features in the image [16] or egomotion [17]. A different approach is independent detection of features in each frame, and association of features in successive frames. Features used by this approach include color [18] or corners: [19] and [20] use the SUSAN [21] and Frstner [22] corner detectors, respectively. [23,24] and also [19] use the Harris [15] corner detector. Feature tracking based on Kalman filter appears in [19,20,23,24]. Bretzner and Lindeberg [25] presents a view-based image representation, the qualitative multi-scale feature hierarchy.

The KLT (Kanade-Lucas-Tomasi) tracking algorithm [26] is based on a model of affine image change. Features are selected to maximize tracking quality. A feature is present if the eigenvalues of the auto-correlation matrix are significant. Monitoring tracking quality is based on a measure of dissimilarity that uses the affine motion as the underlying image change model. Tommasini et al. [27] improves the original KLT based on an outlier rejection rule.

2.3. Evaluation of feature detectors and trackers

Several different approaches to evaluation of feature detectors and trackers were suggested. Zheng and Chellapa [17] and Verestóy and Chetverikov [28] use the idea that the beginning and end of a track should follow the same real-world point, but they ignore the interior of the tracks. Ground truth information was utilized in [29,30] and [31]. This approach is more appropriate for laboratory tests than for real-life sequences.

Schmid et al. [4] introduces two evaluation criteria for interest points: repeatability rate and information content. Repeatability rate is the percentage of total observed points that are detected in both images. It evaluates the geometric stability under different transformations. Information content measures the distinctiveness of features. Based on these criteria, six interest point detectors are compared, finding the improved version of the Harris operator (ImpHarris) superior to the others.

3. Operator for feature detection

To accomplish scene-consistent detection of feature points in video sequences, we first present an operator that has been suggested [6] for detecting points in static images.

3.1. Intuition

The suggested operator detects local extrema of the intensity function, but only in domains where the intensity function is smooth. Intuitively, one may think of the detected domains as hilltops (or equivalently, bottoms of valleys) of the intensity function.

A property of local extrema of a smooth function is that its gradient on local closed curves containing the extremum, points outward *along the whole closed curve*. The operator does *not* look for these closed curves *explicitly*, but rather, it takes advantage of the discontinuity of the 2D arctan function for fast and robust detection of such domains, as the next subsection would show.

3.2. Definition

The gradient map of an image in Cartesian coordinates is estimated by:

$$\begin{aligned}\nabla \mathrm{I}(x,y) &= \left(\frac{\partial}{\partial x}\mathrm{I}(x,y), \frac{\partial}{\partial y}\mathrm{I}(x,y)\right)\\ &\approx ([D_\sigma(x)G_\sigma(y)] * \mathrm{I}(x,y), [G_\sigma(x)D_\sigma(y)] * \mathrm{I}(x,y)),\end{aligned}$$

where $G_\sigma(t)$ is a 1D Gaussian with mean 0, and standard deviation σ, and $D_\sigma(t) = \frac{\mathrm{d}}{\mathrm{d}t}G_\sigma(t)$. We convert to polar coordinates and compute the gradient argument:

$$\theta(x,y) = \arg(\nabla \mathrm{I}(x,y)) = \arctan\left(\frac{\partial}{\partial y}\mathrm{I}(x,y), \frac{\partial}{\partial x}\mathrm{I}(x,y)\right),$$

where the 2D arctan function is defined by:

$$\arctan(y,x) = \begin{cases} \arctan(\frac{y}{x}) & \text{if } x > 0\\ \arctan(\frac{y}{x}) + \pi & \text{if } x < 0,\ y \geq 0\\ \arctan(\frac{y}{x}) - \pi & \text{if } x < 0,\ y < 0\\ 0 & \text{if } x = 0,\ y = 0\\ \frac{\pi}{2} & \text{if } x = 0,\ y > 0\\ -\frac{\pi}{2} & \text{if } x = 0,\ y < 0. \end{cases} \tag{1}$$

Notice the well known discontinuity at the negative part of the x-axis (Fig. 1 (Left)), which is the basis for our method.

3.3. Detecting presence of a certain range of directions of intensity normal

The discontinuity of the 2D arctan occurs at the negative x-axis, so it corresponds to angles of π or $-\pi$ radians. Because $\theta(x,y)$ is the azimuth of the normal to the

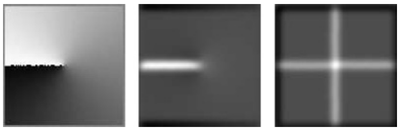

Fig. 1. (Left) The 2D arctan. Pay attention to the discontinuity at the negative part of the x-axis, which is the basis of the method. (Middle) The y-derivative of the 2D arctan. (Right) Rotate the 2D arctan by 0°, 90°, 180°, and 270°, differentiate along the y-direction, rotate back, and sum the responses. (An isotropic operator.) Note that the center has the strongest response, as it is a point surrounded by local closed curves, where the intensity normal points outward in *all* directions.

intensity function, the presence of such a discontinuity implies that the azimuth of the normal there is π or $-\pi$ radians. Therefore, the presence of the discontinuity implies that part of our goal, intensity normal which points outward along the whole closed curve, has been accomplished: we know of the presence of a certain range of directions of the intensity normal (near π or $-\pi$ radians).

But how can we detect the discontinuity of the 2D arctan? To do so, we define an operator:

$$\mathbf{Y}_{\arg} \stackrel{\text{def}}{=} \frac{\partial}{\partial y}\theta(x,y) \approx [G_\sigma(x)D_\sigma(y)] * \theta(x,y). \tag{2}$$

When one differentiates $\theta(x,y)$ with respect to y, the derivative approaches infinity ($\frac{\partial}{\partial y}\theta(x,y) \to \infty$) at the discontinuity ray. In practice, this appears as a very strong response at the discontinuity ray, as can be clearly seen in Fig. 1(Middle). Obviously, it is easy to isolate such a response, e.g., by thresholding. In other words, because of the strong (theoretically infinite) response of the derivative of $\theta(x,y)$, we can isolate the discontinuity of $\theta(x,y)$, which implies the presence of a certain range of values (near π or $-\pi$) of the angle of the intensity normal.

3.4. Detecting locations surrounded by intensity normals in all orientations

To attain our objective and detect the presence of an outward normal along the whole closed curve, we have to define an operator whose strongest response occurs when the closed curve contains an outward normal in every possible orientation (an isotropic operator). Ideally, to define this operator based on $Y_{\arg}$, one has to rotate the original image in all possible angles α, operate $Y_{\arg}$, rotate the results back to their original pose, and integrate the responses over all α.

In practice, we define an isotropic version of the operator, called: $D_{\arg}$, to be the result of rotating the original image by 0°, 90°, 180°, and 270°, operating $Y_{\arg}$ on the rotated images, rotating back to the original pose, and summing the four responses (Fig. 1(Right)). The infinite response of $Y_{\arg}$ is the reason why only four angles are enough for the isotropic operator $D_{\arg}$: differentiating $\theta(x,y)$ in any direction which is *not* exactly parallel to the x-axis yields an infinite response at the negative x-axis due to the discontinuity of the 2D arctan. Consequently, even differentiation in merely two perpendicular directions would yield an infinite response at the disconti-

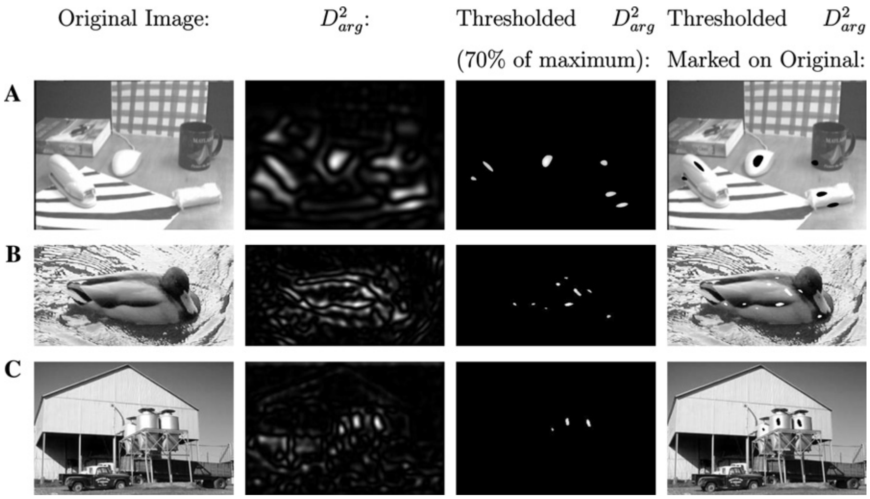
Original Image:
D^2_{arg}:
Thresholded D^2_{arg} (70% of maximum):
Thresholded D^2_{arg} Marked on Original:
A
B
C

Fig. 2. Images with domains of maximal D^2_{arg} marked. Pay attention that domains rich with edges or with flat objects were not detected (low D^2_{arg}). (A) D^2_{arg} detects all four convex 3D objects, but not edge-dense regions. (B) D^2_{arg} detects highly convex domains of the duck. The water is not detected, despite sharp intensity changes. (C) D^2_{arg} detects the three tanks, but not the barn, car or truck, which are flat objects.

nuity ray for *any* orientation. Experiments demonstrating the robustness of D_{arg} to orientation changes were reported in [6].

All in all, the operator locates extrema of smooth intensity patches by first isolating a certain range of values of the gradient argument $\theta(x,y)$, and then applying the method to the rotated images. The specific range of azimuths of the intensity normal is detected by the discontinuity ray of the 2D arctan, whose presence indicates an angle near π or $-\pi$ radians. To detect the discontinuity ray, we differentiate $\theta(x,y)$ with respect to y. We then receive infinite responses at locations where the discontinuity ray is crossed. Examples of domains where strong D^2_{arg} responses occur appears in Fig. 2.

Since we are looking for a qualitative shape description, the Y_{arg} operator is very robust, in contrast with classic methods of shape-from-shading (e.g., [32]; see a survey at [33]). Many intensive robustness tests of the operator are described in [6]: robustness in illumination strength changes, and in variations of orientation or scale. The robustness of the operator in dominating textures has lead to its usage for camouflage breaking [34], thus relaxing the original demand of constant albedo of the subject.

4. Response of Y_{arg} to the intensity surface and scene geometry

This section presents the mathematical basis of our claim that the response of Y_{arg} is stable. The complete proofs of the theorems are included in Appendices A and B. By definition, the theorems hold for D_{arg}, too.

4.1. Response to the intensity surface

First, we qualitatively characterize the behavior of Y_{arg} in continuous ("well-behaving") domains. We analyze the case where, the original intensity function $I(x,y)$ is twice continuously differentiable. Let (x_0,y_0) denote an image point, where $\frac{\partial}{\partial y}\theta(x,y)$ approaches infinity. In other words, (x_0,y_0) is a point of high Y_{arg} response (recall the strong response ray in Fig. 1 (Middle)). Our basic observation is that at such a point (x_0,y_0), $\theta(x,y)$ has a jump-discontinuity (with respect to the y direction):

1. Because $I(x,y)$, $\frac{\partial I(x,y)}{\partial x}$ and $\frac{\partial I(x,y)}{\partial y}$ are differentiable and continuous, and for all points (x_0,y_0) the left- and right-hand limits: $\lim_{y\to y_0}\pm\arctan(y,x_0)$ exist, it follows that $\theta(x,y)$ has left- and right-hand limits in the y direction, anywhere.
2. If at point (x_0,y_0) the left- and right-hand side limits are equal, $\theta(x_0,y)$ is continuous or has a removable singularity.
   (a) If $\theta(x,y)$ is continuous: because $\frac{\partial I(x,y)}{\partial x}$ and $\frac{\partial I(x,y)}{\partial y}$ are assumed differentiable anywhere, point (x_0,y_0) must be a point where $\arctan(y,x_0)$ is continuous. Since $\arctan(y,x_0)$ is differentiable at all points where it is continuous, it follows that $\theta(x,y)$ is also differentiable.
   (b) If $\theta(x,y)$ has a removable singularity: the estimation of Y_{arg} is achieved using a convolution Eq. (2). By definition, the convolution is an integral. The integral of a function with a removable singularity is identical to that of the fixed function (i.e., if one sets the value of the function at the singular point to the value of the left- and right-hand side limits of the function at that point). Therefore, this case is similar to that of a continuous $\theta(x,y)$, and $\frac{\partial}{\partial y}\theta(x,y)$ does not approaches infinity in this case, too.
3. If the left- and right-hand limits are different, the derivative would approach infinity. This is the jump-discontinuity case.

We are interested in domains, where Y_{arg} approaches infinity; they are the stable points of the response. Formally,

Theorem 1. *Let* $I{:}R\times R\mapsto R\in C^2$ *(i.e.,* $I(x,y)$ *is twice continuously differentiable w.r.t both* x *and* y*) be an intensity function. If* (x_0,y_0) *is a point, where:* $\lim_{y\to y_0}\frac{\partial}{\partial y}\theta(x,y)|_{x=x_0}=\pm\infty$ *(i.e., a point, where* $Y_{\text{arg}}\to\pm\infty$*), then there exists* $\varepsilon>0$ *so that for all* y*, for which* $|y-y_0|<\varepsilon$*, one of the following cases holds:*

1. $\frac{\partial I(x,y)}{\partial y}|_{x=x_0}=0$ *for all* y *and* $\forall y<<y_0$, $\frac{\partial I(x,y)}{\partial x}|_{x=x_0}\geqslant 0$, *and* $\forall y>y_0$, $\frac{\partial I(x,y)}{\partial x}|_{x=x_0}<0$.*
2. $\frac{\partial I(x,y)}{\partial y}|_{x=x_0}>0$ *for* $y<y_0$ *and* $\frac{\partial I(x,y)}{\partial y}|_{x=x_0}=0$ *for* $y>y_0$, *and**
   (a) $\forall y>y_0$, $\frac{\partial I(x,y)}{\partial x}|_{x=x_0}=0$ *or*
   (b) $\forall y<y_0$, $\frac{\partial I(x,y)}{\partial x}|_{x=x_0}=0$ *or*
   (c) $\forall y<y_0$, $\frac{\partial I(x,y)}{\partial x}|_{x=x_0}>0$, *and* $\forall y>y_0$, $\frac{\partial I(x,y)}{\partial x}|_{x=x_0}<0$.*
3. $\frac{\partial I(x,y)}{\partial y}|_{x=x_0}<0$ *for* $y<y_0$ *and* $\frac{\partial I(x,y)}{\partial y}|_{x=x_0}=0$ *for* $y>y_0$,*
   except for the case, where $\forall y:y\neq y_0$, $\frac{\partial I(x,y)}{\partial x}|_{x=x_0}>0$.

4. *(x_0, y_0) is a local extremum of $I(x_0, y)$,*
except for the case, where $\forall y : y \neq y_0$, $\frac{\partial I(x,y)}{\partial x}|_{x=x_0} > 0$.

* *The case where the conditions for $y < y_0$ are swapped with those for $y > y_0$ is also valid; it is an equivalent case and was therefore omitted.*

The complete proof of Theorem 1 appears in Appendix A. The rest of this subsection illustrates the response of Y_{arg} to the different features described by Theorem 1.

Fig. 3 demonstrates Case 1 using the intensity function: $I(x, y) = -x \cdot y$. It follows that: $\frac{\partial I(x,y)}{\partial x} = -y$, $\frac{\partial I(x,y)}{\partial y} = -x$. The feature point is $(x_0, y_0) = (0, 0)$. To see that indeed this intensity surface belongs to Case 1: $\frac{\partial I(x,y)}{\partial y}|_{x=x_0} = x_0 = 0$. $\forall y < y_0 = 0$, $\frac{\partial I(x,y)}{\partial x} = -y > 0$, and $\forall y > y_0 = 0$, $\frac{\partial I(x,y)}{\partial x} = -y < 0$. We can see the strong response of Y_{arg} at (0,0) in this case (Fig. 3 (Case 1, right)).

Fig. 3 exhibits Case 2 using the intensity function:

$$I(x, y) = \begin{cases} -y^2 & \text{if } y < 0 \\ 0 & \text{if } y \geqslant 0 \end{cases}$$

whose derivatives are:

$$\frac{\partial}{\partial y} I(x, y) = \begin{cases} -2y & \text{if } y < 0 \\ 0 & \text{if } y \geqslant 0 \end{cases}$$

and $\frac{\partial}{\partial x} I(x, y) = 0$ for all x. This is an example of Cases 2a or 2b (it is appropriate for both), where the feature points are $y = 0$. The strong negative response (black strip) can be seen in Fig. 3 (Case 2, right).

Similarly, in Fig. 3 the intensity function:

$$I(x, y) = \begin{cases} y^2 & \text{if } y < 0 \\ 0 & \text{if } y \geqslant 0 \end{cases}$$

demonstrates Case 3. The response of Y_{arg} in this case is positive.

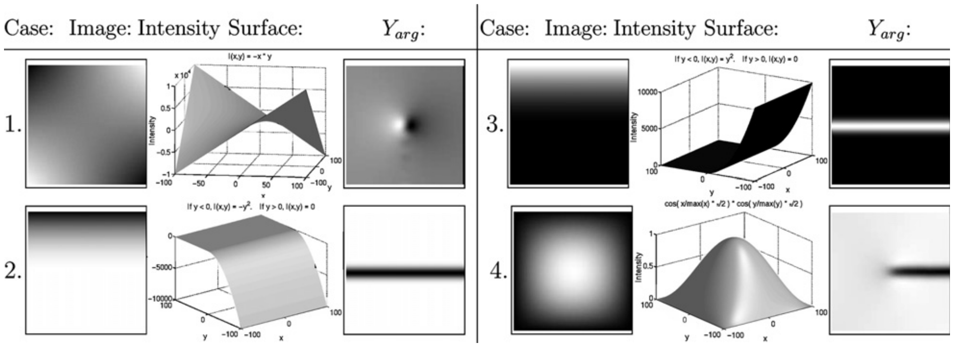

Fig. 3. Demonstration of Theorem 1: *Case 1:* $I(x,y) = -x \cdot y$. The feature is at (0,0), and indeed Y_{arg} has a very strong response there. *Case 2:* For $y > 0$, $I(x,y) = -y^2$, for $y \leqslant 0$, $I(x,y) = 0$. The features are at $y = 0$. Y_{arg} has a very strong negative response there—the black stripe. *Case 3:* For $y > 0$, $I(x,y) = y^2$, for $y \leqslant 0$, $I(x,y) = 0$. The features are at $y = 0$. Y_{arg} has a very strong positive response there. *Case 4:* $I(x,y) = \cos(\frac{\pi x}{2\max(x)})\cos(\frac{\pi y}{2\max(y)})$. The features are at the positive part of the x-axis, and indeed Y_{arg} responds to them. The negative part of the x-axis is the case excluded from Case 4.

The demonstration of Case 4 in Fig. 3 presents a local maximum of the intensity function in the y direction. The intensity function in the example is:

$$I(x,y) = \cos\left(\frac{\pi x}{2\max(x)}\right)\cos\left(\frac{\pi y}{2\max(y)}\right)$$

whose maximum w.r.t y is at the x-axis. A strong negative response of Y_{arg} (Y_{arg} approaches $-\infty$) occurs at the positive part of the x-axis. There is no strong response at the negative part of the x-axis, because there $I(x,y)$ is monotonically increasing as a function of x, and this is exactly the case excluded from Case 4 ($\frac{\partial I(x,y)}{\partial x}|_{x=x_0} > 0$). As this example shows, when an intensity function has a local extremum (in the strong sense), the typical response of Y_{arg} would be to one side of the x-axis, either the negative or the positive, and to the center (i.e., the local extremum itself). By using the isotropic operator D_{arg}, one receives a strong response in all axes, but the strongest response is at the extremum itself (as any orientation of the image contributes to the center, but not to all other parts of the axes). This enables the isolation of the point of extremum of the intensity function.

4.2. Response to local 3D scene structure

By now, our analysis related Y_{arg} merely to the intensity function. This section would relate the domains, where Y_{arg} approaches infinity to the 3D scene object.

4.2.1. Basic assumptions

Let us assume an ideal Lambertian surface of constant albedo is illuminated by a point light source at infinity and is photographed by a projective camera. We assume a camera model, where the radiance of the surface (L) in the direction of the camera is proportional to the irradiance of the image (E) at the corresponding patch (i.e., $E \propto L$). Such a system is described at [35] (a lens system, where the off-axis decay is compensated).

4.2.2. The detected domains—case 4

Case 4 of Theorem 1 is detection of local intensity extrema in the y direction. This subsection characterizes the geometry of scenes where these extrema occur.

Assuming a Lambertian surface and an orthographic projection, the intensity function would be:

$$I_{\mathrm{orth}}(x,y) = \rho\overrightarrow{N}(x,y)\cdot\overrightarrow{L} = \rho\cos(\alpha(x,y)), \tag{3}$$

where ρ is (constant) albedo; $\overrightarrow{N}(x,y)$, normal to the 3D surface $z(x,y)$; $\overrightarrow{L}$, light source direction (assumed constant); and $\alpha(x,y) = \angle(\overrightarrow{N}(x,y),\overrightarrow{L})$, the angle between the light source direction and the normal to the scene surface. Obviously, $\alpha(x,y) \in [0,\frac{\pi}{2}]$. In this domain, the cosine decreases monotonically. Therefore, if $\alpha(x,y)$ has a maximum (minimum) at point (x_0,y_0), then $\cos(\alpha(x,y))$ has a minimum (maximum) there, and vice versa. It follows that if $I_{\mathrm{orth}}(x,y)$ attains a local extrema at (x_0, y_0), then the angle $\alpha(x, y)$ also attains a local extrema there (but of the opposite type). Thus, local extrema of the intensity function I_{orth} are local extrema (of the opposite type) of the angle $\alpha(x, y)$. Appendix B proves that this remains valid under a perspective projection, too.

Therefore, detections by Y_{arg} in Case 4 (i.e., extrema of the intensity) correspond to extrema of the angle between light source direction and surface normal. As long as the light source direction is constant, detection by Y_{arg} (Case 4) depends merely on scene geometry.

This result explains (to a certain extent) why a *locally* constant albedo is sufficient for Y_{arg} to remain stable: If $\alpha(x, y)$ attains an extremum at a certain location in a vicinity of constant albedo, then $Y_{\mathrm{arg}} \to \infty$ there. As Y_{arg} detects no edges, sharp changes in the albedo yield no additional feature points.

An alternative way to relate the detected extrema points and scene geometry is via level sets. Several level set methods for Shape from Shading [35–37] use intensity extrema as singular points from which reconstruction begins. Local extrema of a smooth intensity domain are local extrema of $z(x, y)$ for a vertical light source ($\overrightarrow{L} = (0, 0, 1)$). The case of an oblique light source, can be reduced to the vertical one by rotation of the coordinate system [38]. Thus, both cases relate detection by Y_{arg} and scene geometry.

To summarize Case 4: in twice continuously differentiable domains, Y_{arg} detects intensity extrema, which correspond to extrema of angle $\alpha(x, y)$. Assuming a constant light source direction, the detected points relate to scene geometry. What distinguishes Y_{arg} (or D_{arg}) from other methods of isolating intensity extrema is its high robustness, and the consistency with which it detects its features.

4.2.3. The detected domains—Cases 1–3

A common property of Cases 1–3 of Theorem 1 is that a first-order derivative, either $\frac{\partial}{\partial x} I(x, y)$ or $\frac{\partial}{\partial y} I(x, y)$, vanishes at one side of point (x_0, y_0), but at least one side of either of them does not. We call this behavior a semi-weak change of sign.

Under the assumptions of Section 4.2.1, a 3D plane of constant albedo produces a constant intensity function (i.e., vanishing image derivatives), due to its constant normal Eq. (3). If this plane turns smoothly into a convex (or concave) surface, the points where the change begins project onto image points with a semi-weak change of sign of the derivatives, as in Cases 1–3. Thus, the characterization of these points relies on the local geometry of the scene surface.

In addition, Cases 1–3 should be considered in a broader context: the complete mathematical analysis of the behavior of Computer Vision algorithms including extreme cases is highly important, but it must take into account the distribution of images in a natural environment; even the human vision system may fail on rare images. In such an analysis, Cases 1–3 may be considered outliers due to their low frequency. This results from the requirement of a smooth transition between the planar and convex (concave) parts of an object, whose probability in natural images is relatively low. Higher level processing (e.g., tracking) can be used to further decrease the level of noise.

4.2.4. The stability of the detected intensity extrema

Isolation of extrema of the intensity function is certainly not new, for example [39]. Local extrema divide into stable and unstable according to the determinant of the Hessian matrix: critical points with a nonvanishing determinant are generic (see [39]). Some cases of Theorem 1 (e.g., Case 1) necessarily represent stable extrema, but some cases may include unstable extrema as well. The use of the isotropic operator D_{arg}, i.e., differentiation in all orientations, reduces the likelihood of semi-weak derivatives, or in other words, of unstable extrema.

Another explanation for the stability of the detected points lies in their aforementioned relation to scene geometry. Instability of the intensity extrema implies an inherent instability of the scene: either a critical lighting direction (cf. [40], there discussing stability of viewpoints; a similar analysis can be applied to lighting directions) or an object deformation.

This section shows that under the Lambertian model Y_{arg} responds to certain properties of the surface normal. This establishes a connection between Y_{arg} and *geometric* features of the 3D object, leading to stability of the detected points. The discussion is incomplete without referring to specular reflection: Specular reflection indeed distracts Y_{arg}, being [to a certain extent] a virtual image of the light source.

4.3. Theoretic basis of feature detectors

The majority of studies of feature point detectors is based on edges or corners (e.g., Junction Detection [3], ImpHarris [4], Harris-Laplacian [5]). Nevertheless, edges and corners are not inherent properties of objects and may occur on their boundaries. As such, they are subjected to effects of the background or neighboring objects. Consequently, these feature points cannot be related to scene geometry.

The KLT [2] detector is based on monitoring image changes assuming an affine change model. These changes need not be related to scene geometry.

We see that many existing feature detectors are based on empiric results or a priori assessment of the informativeness of certain features, but there is no theoretic model to relate them to the scene. D_{arg}, on the other hand, is related to scene geometry. This ensures the theoretical correctness of its selected feature with respect to the scene.

Similarly to D_{arg}, there exist other feature detectors which isolate intensity maxima, and as such their features relate to scene geometry as well. There, however, care must be taken to isolate only those maxima which appear inside smooth intensity domains to avoid edge detection (as done, for example in [41]). D_{arg} has the advantage of built-in ignoration of (or low response to) edges, and the robustness of its computation.

5. The algorithm

As stated in the introduction, our method focuses on feature *detection*. We evaluate its fitness to higher level tasks using a complete detection–tracking system. This section presents the system. As our goal is evaluation of the detection part of the system, we use a simple tracker (Fig. 4). Intuitively, if a simple tracker is sufficient for a

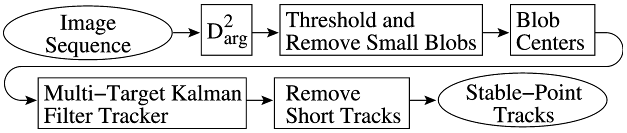

Fig. 4. Block diagram of the stable points detection and tracking algorithm. (Upper row blocks:) The stable-points detector. (Lower row blocks:) The tracking facility. The input to the tracking facility consists of point locations only (blob centers). The tracking facility has no other knowledge of the image.

certain detector while other detectors require more sophisticated trackers, then this detector should be more stable. A more sophisticated tracker may compensate for instabilities of detectors.

The *stable-point detector* is based on the D_{arg} operator. As the input is discrete and bounded, the algorithm isolates maxima of the dynamic range of D^2_{arg} (by thresholding). The stable points are the centers of gravity of the blobs produced by thresholding D^2_{arg}. These stable points are the only input to the point tracker: the tracker has no knowledge about the mechanism producing its input points, and it obtains no other knowledge of the image.

The *point tracker* is a classic multi-target 2D Kalman filter tracker. The Kalman filter [23,24,42] is used for point tracking, assuming a constant velocity motion model, and doing the update by setting the position components of the state vector of the filter to the measurement itself (i.e., to the point produced by the stable-point detector), each time a point is associated with a track. The velocity components remain unchanged. This reinforces the claim that stability is due to the stable-point detector, rather than the filtering process. We remove tracks which are too short, for further stabilization of tracking.

5.1. Demonstration by video sequences

Let us demonstrate the stability of the algorithm by three video sequences taken by a hand-held camera. The sequences contain complex camera motion: rotation, acceleration, changes of motion direction, vibrations, and zooms. In all sequences, tracking parameters were identical, except for two cases: the maximal idle time (i.e., time when only Kalman-based approximation is maintained, without any new measurement associated to that track) and minimal track length. These parameters depend on the variability and length of the video sequence, respectively.

In the following examples of video sequences (Figs. 5, 6), only the interior of the marked black frame participates in the tracking (to avoid boundary conditions). The tracks resulting from the stable-point extraction algorithm are marked on the images. The exact feature point is the center of each square (or: center of X mark). A label to the right of each mark holds its track ID.

Fig. 5 (Left) ("toys") contains frames from a video sequence taken in the laboratory. The sequence introduces a notable change in viewing angle (pay attention to the high variation of angle of the shadow of the pen tracked by Track 5). In spite of camera motion, most of the detected points are stable. Track 5, for example, has a short

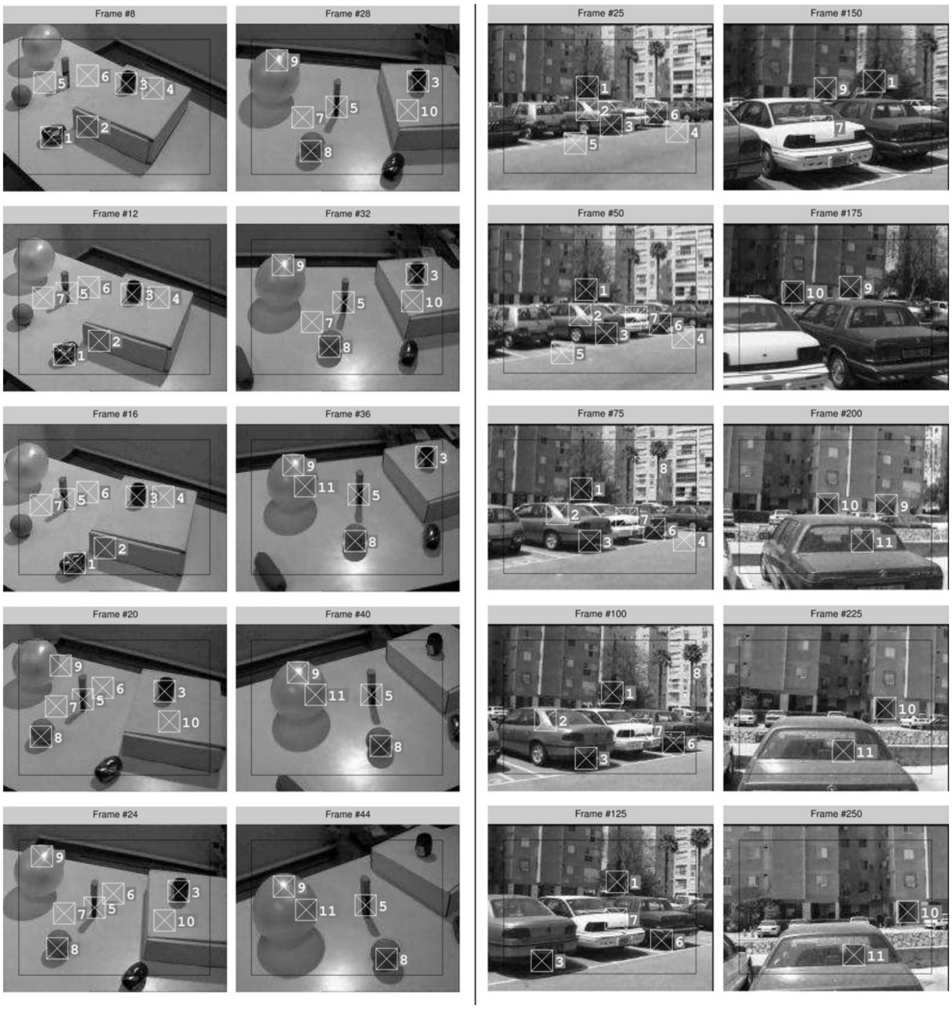

Fig. 5. (Left) Toys: Video sequence of objects in the laboratory. Note, for example, track 3 which follows the same object as long as it is in the frame, or track 8 which consistently detects the tennis ball. (Right) Parking-lot: an outdoors video sequence of a parking lot. Despite the variability and parallax in the scene, tracking is correctly maintained in the vast majority of the cases. Track 1, for example, correctly tracks the tree in frames #1–#170. Pay attention to the change of scales of that tree in frames #25 and #150 (top row).

erroneous detection at the beginning of the sequence (for 4 frames), but for most of the sequence (37 frames out of 46) it tracks the 3D object (a pen) correctly.

Fig. 5 (Right) (‘‘parking-lot’’) shows a sequence of a parking lot. Frames #25–#125 demonstrate faithful tracking despite a considerable zoom. In frames #175–#250, tracks 9 and 10 correctly follow the background building, while track 11 consistently detects a seat inside a parking car. The parallax depicted by the relative motion of these tracks could be easily used for 3D scene correspondence.

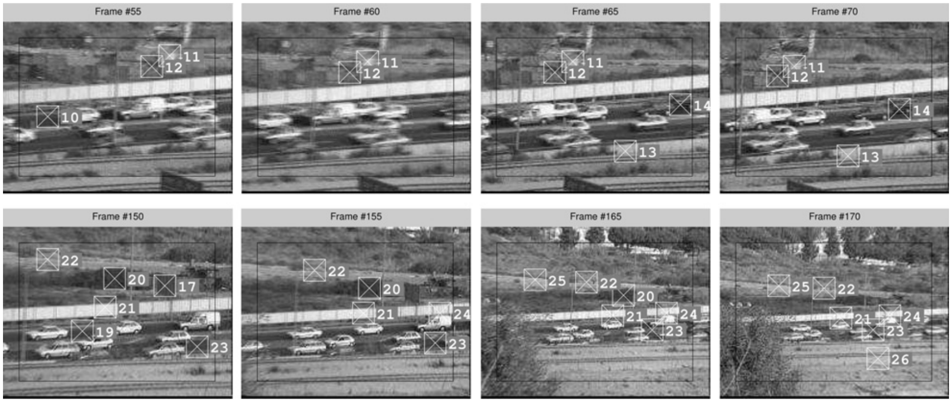

Fig. 6. Traffic: A sight of a highway from a nearby hill. The scene is very dynamic, combining several effects: camera motion, scene objects motion, and zoom. Results are worse then in the toys or parking-lot sequences, but are still usable for algorithms requiring correspondence between successive frames. Note the fast camera motion of tracks 11 and 12 (top row) and the correct tracking despite strong zoom of track 22 (bottom row).

Fig. 6 ("traffic") is a sight of a highway from a nearby hill. The scene is extremely difficult to track. It combines a fast camera pan, motion of scene objects (cars), and 1:6 zoom out. Due to the substantial zoom, the stability of the hand-held camera becomes a significant factor for this sequence. As expected, the results of this sequence are worse then of the other two (toys and parking-lot), but still, most of the tracks are correct for most of the time even for this challenging video.

6. Performance evaluation

An important issue in stable point tracking is the method of evaluation of algorithms. Various evaluation techniques appear in the literature, but they suffer from certain flaws (see Section 2.3). To overcome the aforementioned flaws, this section defines two different measures: one is more relevant for maximal-time point tracking; the other, for correspondence of points in successive frames. These measures can serve for evaluation of 3D point tracking algorithms in general. Their reference to *tracking* of 3D scene points distinguishes them from existing evaluation measures.

6.1. Definition of a track

We first introduce the basic terminology. Our video sequence has k frames, each of $m \times n$ pixels. The *set of detected points, D*, is the set of all pixels which a tracker selected.

The *track ID function* $t{:}D \mapsto N$ associates numbers ("track IDs") with selected pixels, as determined by the tracker. In a certain frame, only one pixel may have a certain track ID. Let Q be the set of all points whose track ID is T: $Q \overset{\text{def}}{=} \{q_j \in D \mid \mathrm{t}(q_j) = T\}$.

The *set P_ε^T of all pixels at distance ε pixels from track T* is the set of all image points whose distance to a point with track ID T in their frame is less than or equal to ε.

$$P_\varepsilon^T \overset{\mathrm{def}}{=} \{p \mid \exists q \in Q \cap \mathrm{frame}(p) : \ \|p - q\| \leqslant \varepsilon\},$$

where frame(p) is the frame where p lies. In the rest of this section, p will denote an element from P_ε^T.

6.2. Selecting the correct scene point of a track

For the formal definition, let us assume we have an ideal function $\Gamma : P_\varepsilon^T \mapsto R^3$ which maps pixels in the video sequence to the 3D scene points from which they originated. That is, $\Gamma(p)$ is the 3D scene point whose projection on frame(p) is p itself. Also, let proj_f: $R^3 \mapsto R^2$ be the projection of a 3D scene point onto the plane of frame f.

An automatic tracker may sometimes fail to correctly track the same scene point, resulting in inconsistent tracking. Namely, a track may follow two different 3D points, each of them being tracked in a different part of the sequence. Such parts may be interleaved, or may involve more than two scene points. To quantify the level of deviation from ideal tracking, we define which 3D point in a track is the correct 3D scene point. We say that point Γ_1 is *the correct scene point of track T* if Γ_1 obtains the maximal tracking time among all scene points of pixels in P_ε^T, and its tracking in track T began earlier than any other 3D scene point with identical tracking time (if more than one scene point attains maximal tracking time). Formally, Γ_1 is *the correct scene point of track T* if $\forall \Gamma_3 \in R^3$:

$$|\{p \mid \Gamma(p) = \Gamma_1\}| > |\{p \mid \Gamma(p) = \Gamma_3\}|$$

or $\exists \Gamma_2 \in R^3$, $\forall \Gamma_3 \in R^3$:

$$|\{p \mid \Gamma(p) = \Gamma_1\}| \geqslant |\{p \mid \Gamma(p) = \Gamma_3\}|$$
$$\text{and } |\{p \mid \Gamma(p) = \Gamma_1\}| = |\{p \mid \Gamma(p) = \Gamma_2\}|$$

and Γ_1 begins to be tracked (under track ID T) earlier than Γ_2.

Bear in mind that both Γ_1 and Γ_2 are tracked under track ID T. The case where Γ_1 begins to be tracked earlier than Γ_2 divides up into two cases:

1. There exists a frame f, where the projections of Γ_1 and Γ_2 are distant more than ε pixels from one another: $\exists f$: $\|\mathrm{proj}_f(\Gamma_1) - \mathrm{proj}_f(\Gamma_2)\| > \varepsilon$. Under this assumption, we say that Γ_1 is the correct point iff:

   $$\min\{f \mid \mathrm{proj}_f(\Gamma_1) \in P_\varepsilon^T\} < \min\{f \mid \mathrm{proj}_f(\Gamma_2) \in P_\varepsilon^T\}.$$

   This definite minimum is assured to exist because in every frame, track T has at most one detected point.
2. In every frame, the projections of Γ_1 and Γ_2 are distant less than or exactly ε pixels: $\forall f$, $\|\mathrm{proj}_f(\Gamma_1) - \mathrm{proj}_f(\Gamma_2)\| \leqslant \varepsilon$. In this case, one may arbitrarily select whether Γ_1 or Γ_2 is the correct point (as their projections are close enough).

6.3. Defining completeness

One way to evaluate the performance of the algorithm is to evaluate its completeness. Intuitively, a track is *complete*, if the same 3D scene point is being tracked, up to a certain level of noise, in every frame where that 3D point appears.

The *completeness measure of track T* is the percent of frames where the correct point Γ_1 has been tracked with track ID T from the set of all frames where Γ_1 appears (i.e., the potential maximal track time):

$$\begin{aligned}\text{Completeness}_T &= 100 \times \frac{\text{Actual Correct Track Time}}{\text{Potential Track Time}}\\ &= 100 \times \frac{|\{f \mid \text{proj}_f(\Gamma_1) \in P_\varepsilon^T\}|}{|\{f \mid \text{proj}_f(\Gamma_1) \in R|_f\}|},\end{aligned}$$

where $R|_f$ is the $m \times n$ pixels rectangle captured by frame f.

The *completeness measure of a tracker for a video sequence* is the average completeness measure over all the tracks it detected for the specific video sequence.

Existing evaluation criteria (cf. Section 2.3) do not attempt to evaluate how persistent a feature tracker is, thus ignoring an important aspect of feature tracking.

To correctly evaluate the position in the following image, the 3D position Γ and the projection matrices proj_f need to be known, but we do not need to have these: as is the case for almost any measuring scheme, we assume that a ground truth is given in order to be compared with. In this paper, the real point correlation has been carried out manually. The comparative norms we propose are aimed at comparing the general ranking of methods, and thus should be used on an accepted set of well defined sequences, for which a ground truth could be calculated. We take two measures to reduce the effect of subjective judgment. First, a rigorous mathematical definition is presented, leaving only Γ and proj_f for manual extraction. Second, a permissive noise level $\varepsilon = 3$ compensates for possible discrepancies between different manual calculations.

6.4. Stability of tracking

A stability measure is a percentage of points correctly matched between images i and i+1. Formally, the *stability measure of frames f_i and f_{i+1} with allowed noise of ε pixels* is:

$$\text{Stability}_\varepsilon(i, i+1) = 100 \times |\{j \in S \mid \|\text{proj}_{f_{i+1}}(\Gamma(p_j^i)) - p_j^{i+1}\| \leqslant \varepsilon\}|/r,$$

where W.L.O.G $S = \{1,\ldots,r\}$ denotes all track IDs common to both frames, and $p_1^i, \ldots, p_r^i$ and $p_1^{i+1}, \ldots, p_r^{i+1}$ are the detected points for the corresponding tracks and frames.

For many practical purposes (e.g., the correspondence problem), a full tracking of 3D points, or even allocation of a single track ID to a single scene point are not a must. In such applications, we look for a reliable association of several anchor points in frame i with points in frame i+1, which are the projection of the same scene point. When associating points in frame i+1 with points in frame i+2, the set of associated scene points may change. The stability measure is adequate for such applications.

The stability measure resembles the "repeatability" measure of [4] in some aspects. The main difference is that "stability" takes automatic tracking into account, thus examining the detection–tracking system as a whole, while "repeatability" aims merely at the detection part, thus implicitly assuming tracking is always correct. This assumption may bias the evaluation. An example is a feature detector which selects all points in a large image region: a tracker cannot distinguish the points due to their high number and proximity. Thus, an evaluation of a detector along with a tracker is more apt to real life tasks.

Another difference from "repeatability" is the group of points which are expected to repeat. "Repeatability" considers all interest points in the part of the scene which appears in two successive frames. In contrast, "stability" uses the number of *tracks* which follow points in both frames (r). Thus, if a track ends at a certain frame, its last point is no longer expected to be detected in the next frame, even if its 3D point still appears in that frame. This is because for the correspondence task, one may ignore such a track. (Bear in mind, that in order to evaluate how long a 3D point has been tracked we use the completeness measure).

7. Experimental results

Various feature trackers have been suggested in the literature (see Section 2). To evaluate the performance of our tracker, we compare the D_{arg}-based tracking algorithm with four other algorithms: Junction Detection [3], ImpHarris [4], KLT [2,26] and Harris-Laplacian [5]. Four of the algorithms under study: Junction Detection, ImpHarris, Harris-Laplacian, and D_{arg} use an identical tracker based on Kalman filter (the one described in Section 5). They all share the same tracker code and identical parameters. The KLT algorithm is itself a combined detector–tracker, and as such cannot be split.

The original implementation [4] of ImpHarris uses 1% threshold of maximum observed interest point strength. This threshold is too low for tracking: it results in a large number of adjacent feature points, even in neighboring pixels, and in many cases in connected edges rather than isolated points (Fig. 7). This phenomenon reduces the ability of a tracker to correctly associate features in successive frames. Fig. 8 (parking-lot sequence only) shows that increasing the threshold to 20% improves tracking results of ImpHarris. Therefore, our comparison would refer to

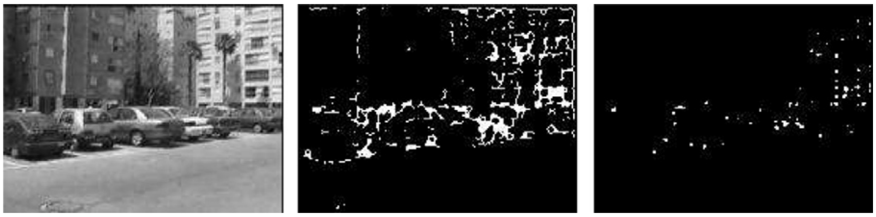

Fig. 7. (Left) Frame #10 of the parking-lot sequence. (Middle) ImpHarris, 1% threshold. (Right) ImpHarris, 20% threshold. The increased threshold improves tracker ability to distinguish between feature points, leading to a more stable detection–tracking system.

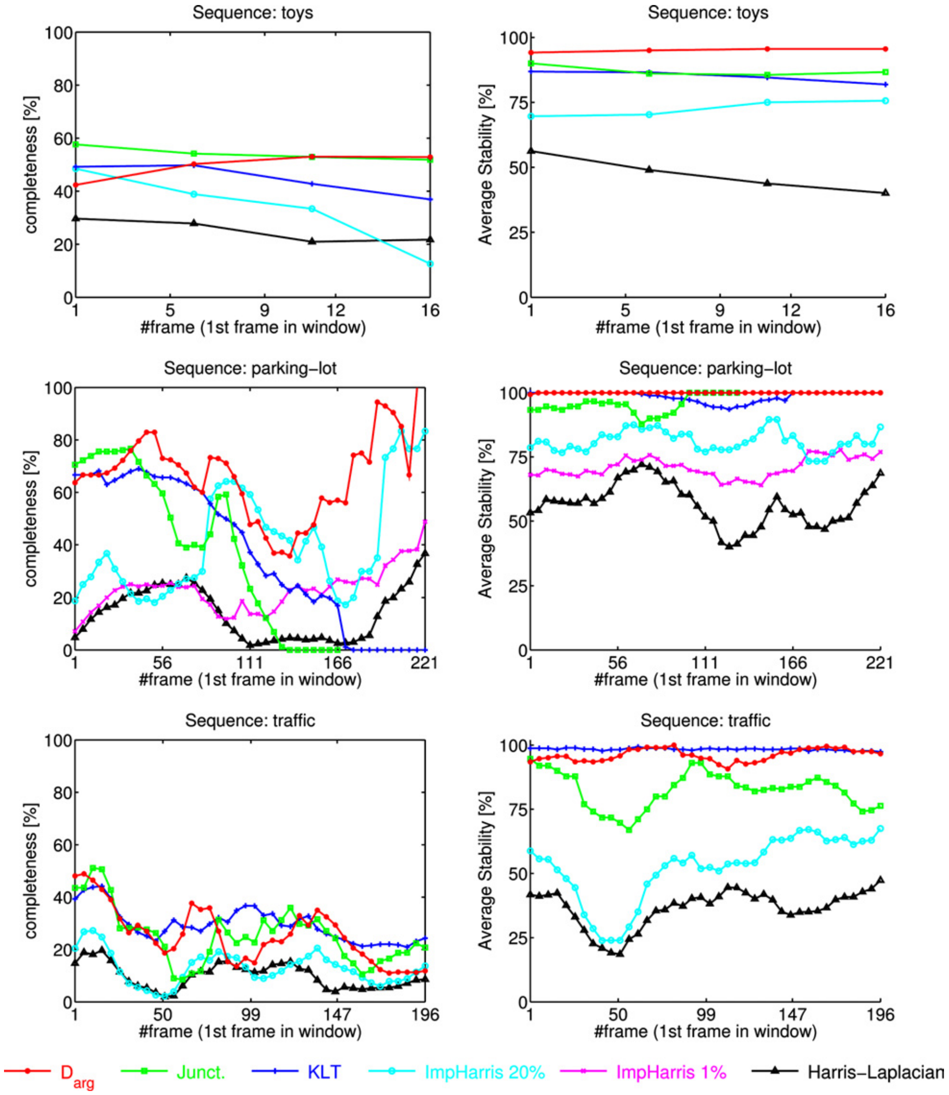

Fig. 8. (Left) Completeness comparison. *On the toys and traffic sequences:* D_{arg} performs as well as Junction Detection and KLT and most of the time better than ImpHarris or Harris-Laplacian. *On the parking-lot sequence:* D_{arg} attains better completeness than the other five trackers. (Right) Stability comparison. Tracking by D_{arg} is more stable than Junction Detection, ImpHarris or Harris-Laplacian. For the toys sequence, D_{arg} is more stable than KLT. For the parking-lot and traffic sequences, D_{arg} and KLT equate.

the 20% threshold version of ImpHarris. Similarly, we employed a 20% threshold for the Harris part of the Harris-Laplacian detector of [5].

In all trackers, tracks shorter than a certain percentage of the length of the video sequence are ignored (toys: 25%; parking-lot 25%; and traffic 10%). The threshold is lower for the traffic sequence due to its higher variability.

To follow the development in time of the completeness and stability measures, a sliding window over frames in the video sequence is employed. The calculation of the measures refers to the frames of the window as if they were the whole video sequence. The window length is: 30 frames, and it shifts by 5 frames each time. The allowed noise level in all sequences is: $\varepsilon = 3$ pixels.

7.1. Completeness comparison

Fig. 8 (Left) shows graphs of the completeness measure for the toys, parking-lot, and traffic sequences. Each axes system shows sliding completeness for the five algorithms: D_{arg}, Junction Detection, KLT, ImpHarris (20% threshold) and Harris-Laplacian (20% threshold). The parking-lot sequence was tested also with the original (1% threshold) version of ImpHarris.

The graphs show that the completeness of D_{arg} is at least comparable to that of Junction Detection and KLT, and usually better than ImpHarris or Harris-Laplacian. *On the traffic sequence*: results for D_{arg}, Junction Detection and KLT are similar for this sequence, being a very difficult one. *On the toys sequence*: Junction Detection performs better than KLT. In part of the sequence, D_{arg} attains higher completeness than the rest of the trackers. *On the parking-lot sequence*: D_{arg} performs better than Junction Detection, KLT, ImpHarris (in both versions) or Harris-Laplacian. The most significant difference between Junction Detection or KLT and D_{arg} is at the last part of the sequence, when D_{arg} reaches very high completeness values (even 100%), while the completeness of Junction Detection and KLT drops. The reason is the high parallax at the end of the sequence, when a car is very close to the camera, while the background buildings are quite far. D_{arg} copes well with parallax, as the feature it tracks is intrinsic to the 3D object. ImpHarris (20%) copes well with the last part of this sequence, but to a lower extent than D_{arg}. Harris-Laplacian (20%) can cope with the parallax, but it attains lower completeness rates in general.

7.2. Stability comparison

Fig. 8 (Right) introduces the sliding average stability criterion: $\text{stability}_{\varepsilon}\ (i, i+1)$, for the four video sequences. As the graphs show, the stability of D_{arg} is higher than that of the other three trackers for the toys and parking-lot sequences. In parts of the parking-lot sequence, KLT equates with D_{arg}. In a small part of this sequence, Junction Detection also equates with D_{arg}. For the traffic sequence we can order the trackers by their stability (descending order): KLT, D_{arg}, Junction Detection, ImpHarris and Harris-Laplacian. In parts of this sequence D_{arg} equates with KLT.

7.3. No-tracking comparison

Another criterion for tracker comparison would be the *no-track time*: the total time a tracker failed to track *any* point at all. We compare the total no-track time over all three video sequences together. Their total number of frames is: $46 + 252 + 227 = 525$.

Harris-Laplacian (20%) maintained tracking at all times. This may be a result of the high number of tracks it generates. Among the other algorithms, D_{arg} has the minimal no-track time: merely 4 no-track frames in all three video sequences. This no-track time is significantly lower then that of the other three methods (ImpHarris: 24 frames; KLT: 81 frames; and Junction Detection: 121 frames).

7.4. Results summary

We see that D_{arg} is more stable than Junction Detection, ImpHarris or Harris-Laplacian, and sometimes (toys sequence) also more than KLT; Sometimes (parking-lot and traffic seq.) D_{arg} and KLT equate. In addition, the stability of D_{arg} is not at the expense of completeness, as D_{arg} maintains completeness of tracking at least comparable to the Junction Detection and KLT trackers (sometimes even a better completeness), and better than the ImpHarris and Harris-Laplacian detectors.

7.5. Interpreting the results

When one interprets the results displayed here, one must refrain from over-generalization of the graphs introduced here, because the graphs rely merely on three video sequences. A comparison from which a "best" detector can be declared, should base on a much more extensive comparison, say on thousands of video sequences. Due to the large amount of manual work, this kind of results analysis is beyond the scope of this paper.

The issue of test sequences refers not only to the amount of sequences, but also to their distribution. ImpHarris, for example, may provide better results than D_{arg} in some edge-intensive sequences, while D_{arg} may yield better results in edge-sparse or edge-saturated domains, being more reliable there (as shown by this paper and in [6]). Different techniques are better suited for different scenes, so their combination should be employed when scene characteristic is unknown a priori. One such combination of ImpHarris at edge-intensive image domains and a detector of extrema of the intensity at smooth domains is reported in [41]. The robustness of D_{arg} may improve the stability of the maxima detector there, and is a subject for future research.

Another factor affecting the stability of algorithms is the amount of features they generate. A large number of features may make the tracking task more difficult. D_{arg}, for example, generated a relatively low number of features (10–20) in the three sequences under study, while Harris-Laplacian (20%) generated 136 tracks for the traffic sequence. In our comparison we adopted the original parameters of the algorithms, except for the Harris-based operators for which an attempt was made to increase stability by higher thresholds. Exploring algorithm parameters beyond the Harris-based attempts is outside the scope of this study, and is a subject for future research.

To evaluate the suitability of algorithms for higher level tasks, we test the detection–tracking system as a whole, rather than its separate components. While it is rather clear that such a comparison of the four algorithms employing an identical

Kalman filter is "fair" and differences in results are due to differences in detected features, the comparison with KLT that uses a built-in tracker may raise some issues. However, we have chosen to include KLT in our comparison since we consider it a commonly used method, and as such it serves as a basis for comparison. It should also be noted that KLT tracking is considered more advanced than a mere Kalman filtering.

8. Task-oriented measures

As explained in Section 6, evaluation of algorithms is task-dependent: different tasks require different evaluation schemes. Fig. 9 ("Flower Garden") demonstrates the problem: tracking may be selective to certain areas of the scene. The scene contains a tree close to the camera and a flower garden farther away. Nonetheless, KLT tracks merely the background, but not the tree. D_{arg} tracks both. For tasks which need the tree be tracked, D_{arg} would be more useful than KLT. However, this fact is reflected by neither completeness nor stability. On the contrary, KLT performs better according to these measures (Fig. 10).

To cope with the task-dependency of evaluation schemes, we suggest that during the computation of completeness and stability one would apply a weighting scheme to the video sequence under consideration. Different parts of the scene would gain different a priori weights according to the higher level task. A track which follows

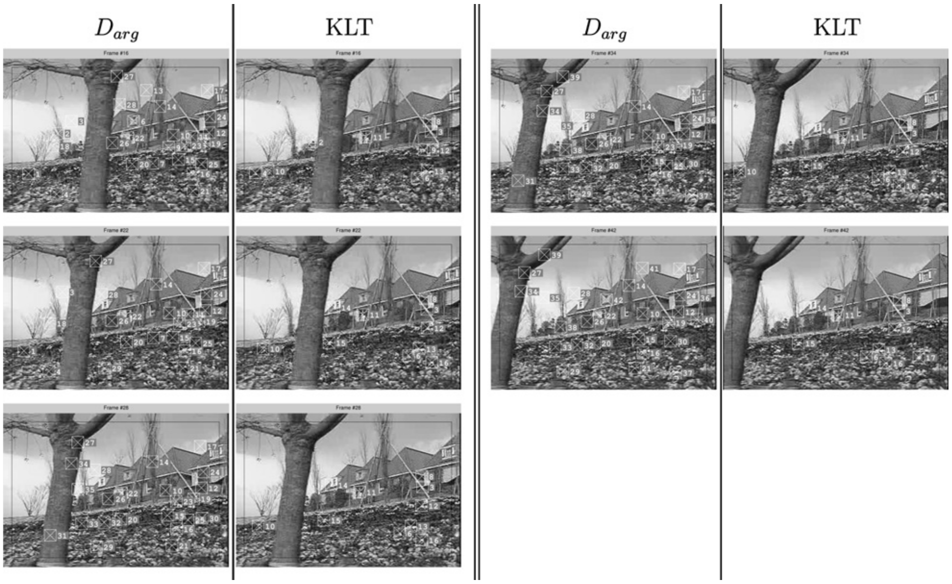

Fig. 9. Tracking the "Flower Garden" video sequence. (Left) Tracking by D_{arg}. Both the tree and the background are being tracked. Pay attention in particular to tracks 27 and 34 which detect the tree. (Right) Tracking by KLT tracks only background objects.

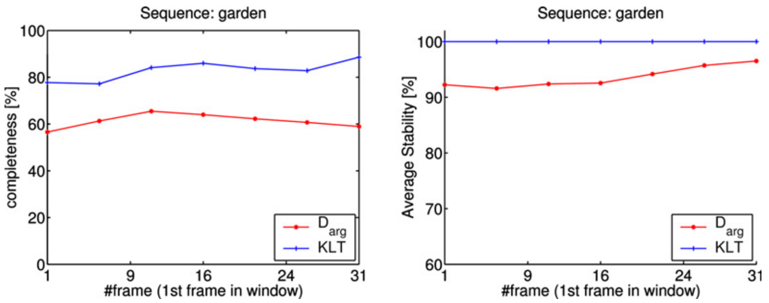

Fig. 10. Completeness and stability comparison for the "Flower Garden" video sequence. Both measures show that KLT performs better than D_{arg}. The measures cannot reflect the fact that D_{arg} (but not KLT) tracks the tree, because it is a smooth 3D edge-sparse object. Applications in which detection of the tree is crucial, may use a priori weighting to give the tree a higher weight; such a weighting may change the preferred algorithm. In our case, such a task-oriented weighting would favor D_{arg}.

a certain scene point would receive its weight. In this manner, the overall analysis would fit better for the required task. This solution is similar, in a sense, to the a priori distribution of viewpoints employed in [40] when determining stable views of objects.

When evaluating tracking of the "Flower Garden" for applications which need tracking of the tree, one would allocate higher a priori weights to the tree than to the background. Because only D_{arg} tracks the tree, this weighting would give rise only to the completeness and stability of D_{arg}, but not to that of KLT. The weighted measures may then show that D_{arg} is superior to KLT. Indeed, for an application where the tree ought to be tracked, D_{arg} is to be preferred to KLT, a fact reflected only by the weighted measures.

9. Conclusions

We have presented an operator for scene-consistent detection of feature points in video sequences. The operator efficiently detects local extrema of the intensity function in twice continuously differentiable image domains, and is insensitive to image edges. Observing that the zero crossing of the gradient argument is a highly prominent feature, we analytically show that this zero crossing relates to specific features of the intensity surface, which, in turn, relates to specific local features of the 3D scene geometry. Based on this operator, a commonly used algorithm for stable point tracking (using a 2D multi-target Kalman filter tracker) is described. Several video sequences demonstrate the high robustness maintained by the algorithm.

Two measures, completeness and stability, are introduced in order to evaluate performance of algorithms for feature point tracking as well as correspondence establishing tasks. These measures overcome various flaws in existing evaluation measures of feature point trackers. The *completeness* measure is aimed at maximizing the tracking time of a 3D scene point. The goal of the *stability* measure is to keep consistent tracking of 3D scene points between successive frames (but the set of tracked scene points may change between frames). Applying task-oriented a priori weighting to these measures was shown to improve the suitability of the suggested measures to the original tasks for which tracking was initiated. The suggested measures are generic, and are suggested as a basis for comparison of 3D point tracking algorithms in general.

We have used the suggested measures in a comparison of our tracker with four other detection–tracking methods. The comparison leads us to the conclusion that a combination of detectors should be employed. The way to combine edge-based and non-edge-based detectors (e.g., corner detectors and D_{arg}) is not trivial, as a sophisticated algorithm that will determine when to use each method is necessary; it is a subject for future research.

Acknowledgments

We thank Professor Ronny Kimmel for his helpful remarks on level sets. The original image of Fig. 2B is courtesy of www.freeimages.co.uk. The original image of Fig. 2C is courtesy of the US Department of Agriculture. This research was supported by grants from the Minerva Minkowski center for geometry, the Excellence Center for Geometric Computing of the Israel Academy of Science, and Israel Ministry of Science.

Appendix A. Response to the intensity surface—Proof

Proof of Theorem 1. Let us define the domain: $D \stackrel{\mathrm{def}}{=} \{(x,y) \in \mathrm{R}^2 \mid y \neq 0 \text{ or } x > 0\}$, and let: $\arctan|_D : D \mapsto \mathrm{R}$ denote the 2D arctan function Eq. (1) reduced to domain D. For each $(x,y) \in D$, $\arctan|_D$ is differentiable. By the chain rule, and based on the differentiability of $\frac{\partial I(x,y)}{\partial x}$ and $\frac{\partial I(x,y)}{\partial y}$, the compound function: $\theta|_E(x,y) \stackrel{\mathrm{def}}{=} \arctan|_D\left(\frac{\partial I(x,y)}{\partial y}, \frac{\partial I(x,y)}{\partial x}\right)$, where $E \stackrel{\mathrm{def}}{=} \{(x,y) \in \mathrm{R}^2 \mid \frac{\partial I(x,y)}{\partial y} \neq 0 \text{ or } \frac{\partial I(x,y)}{\partial x} > 0\}$, is also differentiable. Therefore, if $(x,y) \in E$, then $|\frac{\partial}{\partial y}\theta(x,y)| = |\frac{\partial}{\partial y}\theta|_E(x,y)| < \infty$.

It follows, that if $\frac{\partial}{\partial y}\theta(x,y) \rightarrow \pm\infty$, then $(x,y) \notin E$, which implies: $(x,y) \in \neg E = \{(x,y) \in \mathrm{R}^2 \mid \frac{\partial I(x,y)}{\partial y} = 0 \text{ and } \frac{\partial I(x,y)}{\partial x} \leqslant 0\}$. Let $(x_0, y_0) \in \neg E$ be a point, where: $\lim_{y \rightarrow y_0} \frac{\partial}{\partial y}\theta(x,y)|_{x=x_0} = \pm\infty$. From the above, necessarily: $\frac{\partial I(x,y)}{\partial y}|_{x=x_0, y=y_0} = 0$.

Because of the continuous differentiability of $\frac{\partial I(x,y)}{\partial y}$ (and similarly $\frac{\partial I(x,y)}{\partial x}$), if one examines a small enough neighborhood of (x_0, y_0), then at each side of y_0, $\frac{\partial I(x,y)}{\partial y}$ has a constant sign at that side. From here on, we assume all points (x,y) are in that ε y-neighborhood, and the x-rate is $x = x_0$. We next examine the signs of $\frac{\partial I(x,y)}{\partial y}$ and $\frac{\partial I(x,y)}{\partial x}$ at points (x,y) of a sufficiently small neighborhood of (x_0, y_0):

1. (x_0, y_0) is a point of inflection of $I(x_0, y)$, i.e., $\frac{\partial I(x,y)}{\partial y}$ has a constant sign (except for the point of inflection (x_0, y_0) itself, where $\frac{\partial I(x,y)}{\partial y}|_{(x_0,y_0)} = 0$):
   (a) $\forall y : y \neq y_0,\ \frac{\partial I(x,y)}{\partial y} > 0$
   (b) $\forall y : y \neq y_0,\ \frac{\partial I(x,y)}{\partial y} < 0$.

In both cases (a) and (b), $\theta(x,y)$ has a removable singularity, because the 2D arctan in quadrants I, II ($\arctan_{|\{y:y>0\}}(y,x)$), is continuous, and similarly for quadrants III, IV ($\arctan_{|\{y:y<0\}}(y,x)$). Therefore, there is *no* jump discontinuity.

(c) $\forall y : y \neq y_0,\ \frac{\partial I(x,y)}{\partial y} = 0$. Let us examine $\frac{\partial I(x,y)}{\partial x}$:

(i) $\forall y : y \neq y_0,\ \frac{\partial I(x,y)}{\partial x} < 0$. $\theta(x_0, y) = \pi$ for all $y \neq y_0$. A removable discontinuity exists, which contradicts the assumption that $\frac{\partial}{\partial y}\theta(x,y) \to \pm\infty$.

(ii) $\forall y : y \neq y_0,\ \frac{\partial I(x,y)}{\partial x} \geqslant 0$. For all y, $\theta(x,y) = 0$, so no jump discontinuity occurs.

(iii) $\forall y : y < y_0,\ \frac{\partial I(x,y)}{\partial x} \geqslant 0$, and $\forall y : y > y_0,\ \frac{\partial I(x,y)}{\partial x} < 0$ (or vice versa; the other case is similar). Jump discontinuity occurs in this case, and $\frac{\partial}{\partial y}\theta(x,y) \to +\infty$.

$$\theta(x,y) = \begin{cases} 0 & \text{if } y < y_0 \\ \pi & \text{if } y > y_0 \end{cases}$$

Cases 2, 3, and 4 of the theorem are summarized in Tables 1–3 (respectively). The left column of each table describes the sub-case under consideration. The middle column

Table 1
Case 2: $\forall y : y < y_0,\ \frac{\partial I(x,y)}{\partial y} > 0$ and $\forall y : y > y_0,\ \frac{\partial I(x,y)}{\partial y} = 0$ (or vice versa)

Condition	$\theta(x,y) =$	$\theta(x,y)$ has:
$y < y_0\quad \frac{\partial I(x,y)}{\partial x} > 0$ $y > y_0\quad \frac{\partial I(x,y)}{\partial x} < 0$	$\begin{cases} \arctan\left(\frac{\partial I(x,y)}{\partial y} / \frac{\partial I(x,y)}{\partial x}\right) + \pi & \text{if } y \leqslant y_0 \\ \pi & \text{if } y > y_0 \end{cases}$	Continuity
$y > y_0,\ \frac{\partial I(x,y)}{\partial x} = 0$	$\begin{cases} \theta(x,y) > 0 & \text{if } y < y_0 \\ \theta(x,y) = 0 & \text{if } y > y_0 \end{cases}$	Jump discontinuity
$y < y_0\quad \frac{\partial I(x,y)}{\partial x} = 0$	$\begin{cases} \frac{\pi}{2} & \text{if } y < y_0 \\ 0 \text{ or } \pi & \text{if } y > y_0 \end{cases}$	Jump discontinuity
$y < y_0\quad \frac{\partial I(x,y)}{\partial x} > 0$ $y > y_0\quad \frac{\partial I(x,y)}{\partial x} < 0$	$\begin{cases} \arctan\left(\frac{\partial I(x,y)}{\partial y} / \frac{\partial I(x,y)}{\partial x}\right) < \frac{\pi}{2} & \text{if } y < y_0 \\ \pi & \text{if } y > y_0 \end{cases}$	Jump discontinuity
$y < y_0,\ \frac{\partial I(x,y)}{\partial x} < 0$ $y > y_0,\ \frac{\partial I(x,y)}{\partial x} > 0$	$\begin{cases} \arctan\left(\frac{\partial I(x,y)}{\partial y} / \frac{\partial I(x,y)}{\partial x}\right) + \pi > \frac{\pi}{2} & \text{if } y < y_0 \\ 0 & \text{if } y > y_0 \end{cases}$	Jump discontinuity

Table 2
Case 3: $\forall y : y < y_0,\ \frac{\partial I(x,y)}{\partial y} < 0$ and $\forall y : y > y_0,\ \frac{\partial I(x,y)}{\partial y} = 0$ (or vice versa)

Condition	$\theta(x,y) =$	$\theta(x,y)$ has
$y > y_0,\ \frac{\partial I(x,y)}{\partial x} \leqslant 0$	$\begin{cases} \theta(x,y) < 0 & \text{if } y < y_0 \\ \theta(x,y) = 0 \text{ or } \pi & \text{if } y > y_0 \end{cases}$	Jump discontinuity
$y < y_0\quad \frac{\partial I(x,y)}{\partial x} \leqslant 0$	$\begin{cases} \arctan\left(\frac{\partial I(x,y)}{\partial y} / \frac{\partial I(x,y)}{\partial x}\right) - \pi \leqslant -\frac{\pi}{2} & \text{if } y < y_0 \\ 0 \text{ or } \pi & \text{if } y > y_0 \end{cases}$	Jump discontinuity

Table 3
Case 4: For a minimum point (x_0, y_0), i.e., $\forall y : y < y_0,\ \frac{\partial I(x,y)}{\partial y} < 0$ and $\forall y : y > y_0,\ \frac{\partial I(x,y)}{\partial y} > 0$

Condition	$\theta(x,y) =$	$\theta(x,y)$ has
$y < y_0 \quad \frac{\partial I(x,y)}{\partial x} \leqslant 0$	$\begin{cases} \arctan\left(\frac{\partial I(x,y)}{\partial y} / \frac{\partial I(x,y)}{\partial x}\right) - \pi \leqslant -\frac{\pi}{2} & \text{if } y < y_0 \\ \theta(x,y) > 0 & \text{if } y > y_0 \end{cases}$	Jump discontinuity
$y > y_0 \quad \frac{\partial I(x,y)}{\partial x} \leqslant 0$	$\begin{cases} \theta(x,y) < 0 & \text{if } y < y_0 \\ \arctan\left(\frac{\partial I(x,y)}{\partial y} / \frac{\partial I(x,y)}{\partial x}\right) + \pi \geq \frac{\pi}{2} & \text{if } y > y_0 \end{cases}$	Jump discontinuity
$y \neq y_0 \quad \frac{\partial I(x,y)}{\partial x} > 0$	$\arctan(\frac{\partial I(x,y)}{\partial y} / \frac{\partial I(x,y)}{\partial x})$	Continuity

The case of a maximum point is equivalent.

shows the behavior of $\theta(x,y)$ under that constraint. The right column indicates whether or not a jump discontinuity of $\theta(x,y)$ may occur in this case.

Appendix B. Intensity extrema under perspective projection

Lemma 1. *Let (x,y,z) be a Cartesian coordinate system, and let (u,v) be the coordinate system which results from the perspective projection of a 3D surface $z(x,y)$ on an image plane $I_{\text{pers}}(x,y) : u = -\frac{fx}{z(x,y)},\ v = -\frac{fy}{z(x,y)}$, where f is the focal length.*

If there exists δ for which $\forall x, y$: $|x - x_0| < \delta$ and $|y - y_0| < \delta$, then necessarily there exists δ_2 for which $\forall u, v$: $|u - u_0| < \delta_2$ and $|v - v_0| < \delta_2$.

Proof. To prove the lemma, we consider the bounds on $z(x,y)$ as well as on x and y themselves. The 3D surface $z(x,y)$ consists merely of positive values in the strong sense. Therefore, $\exists M > 0$: $\forall x, y$: $z(x,y) > M > 0$. The image domain is finite and therefore bounded: $\exists A > 0$: $-A \leqslant x \leqslant A$, and $\exists B > 0$: $-B \leqslant y \leqslant B$. The continuity of $z(x,y)$ at point (x_0,y_0) means:

$$\forall \epsilon_1 > 0\ \exists \delta_1 > 0 : \ \forall x, y : \ | x - x_0 | < \delta_1,\ | y - y_0 | < \delta_1 : \ | z(x,y) - z(x_0,y_0) | < \epsilon_1.$$

Let: $z_0 = z(x_0,y_0)$. The required constraint on u follows directly from these properties:

$$u - u_0 = \frac{x_0 f}{z_0} - \frac{xf}{z(x,y)} < f\left(\frac{\delta_1}{M} + \frac{A}{M^2}\epsilon_1\right).$$

Let: $\delta_2 = f(\frac{\delta_1}{M} + \frac{A}{M^2}\epsilon_1)$. We have: $u - u_0 < \delta_2$. The proofs that $u - u_0 > -\delta_2$ and that $|v - v_0| < \delta_2$ are similar in nature and are therefore omitted. □

Theorem 2. *Let $I_{\text{orth}}(x,y)$ be the intensity function produced by orthographic projection of a Lambertian surface $z(x,y)$ with constant albedo illuminated by a point light source at infinity (as described in Section 4.2.1):*

$$I_{\text{orth}}(x,y) = \rho \overrightarrow{N}(x,y) \cdot \overrightarrow{L} = \rho \cos(\alpha(x,y)),$$

where ρ is the (constant) albedo; $\overrightarrow{N}(x,y)$, a normal to the 3D surface $z(x,y)$; $\overrightarrow{L}$, light source direction (assumed constant); and $\alpha(x,y) = \angle(\overrightarrow{N}(x,y), \overrightarrow{L})$. Obviously, $\alpha(x,y) \in [0, \frac{\pi}{2}]$.

Let $I_{\mathrm{pers}}(u,v)$ be the perspective projection of $z(x,y)$. This means that:

$$I_{\mathrm{pers}}(u,v) = I_{\mathrm{orth}}(x,y) = \rho\cos(\alpha(x,y)).$$

If (x_0,y_0) is a point of local maximum of $I_{\mathrm{orth}}(x,y)$, then the perspective projection of $(x_0,y_0,z(x_0,y_0))$, denoted (u_0,v_0), is a local maximum of $I_{\mathrm{pers}}(u,v)$.

Proof. By definition of a local maximum at point (x_0,y_0), the orthographic projection image $I_{\mathrm{orth}}(x,y)$ satisfies:

$$\exists\delta > 0 : \forall x,y : |x - x_0| < \delta, |y - y_0| < \delta : I_{\mathrm{orth}}(x_0,y_0) > I_{\mathrm{orth}}(x,y).$$

Now, according to Lemma 1, the last equation implies:

$$\exists\delta_2 > 0 : \forall u,v : |u - u_0| < \delta_2, |v - v_0| < \delta_2 : I_{\mathrm{orth}}(x_0,y_0) > I_{\mathrm{orth}}(x,y).$$

Substituting $I_{\mathrm{pers}}(u,v) = I_{\mathrm{orth}}(x,y)$ we get:

$$\exists\delta_2 > 0 : \forall u,v : |u - u_0| < \delta_2, |v - v_0| < \delta_2 : I_{\mathrm{pers}}(u_0,v_0) > I_{\mathrm{pers}}(u,v)$$

$\Rightarrow (u_0,v_0)$ is a local maximum of $I_{\mathrm{pers}}(u,v)$. □

Corollary 1. *Let $I_{\mathrm{pers}}(u,v)$ be the intensity function produced by perspective projection of a constant albedo Lambertian surface $z(x,y)$ illuminated by a point light source at infinity (as described in Section 4.2.1): $I_{\mathrm{pers}}(u,v) = \rho\overrightarrow{N}(x,y)\cdot\overrightarrow{L} = \rho\cos(\alpha(x,y))$, where ρ is the constant albedo; $\overrightarrow{N}(x,y)$, a normal to the 3D surface $z(x,y)$; $\overrightarrow{L}$, the light source direction (assumed constant); and $\alpha(x,y) = \angle(\overrightarrow{N}(x,y), \overrightarrow{L}) \in [0, \frac{\pi}{2}]$. If at point (x_0,y_0) the angle $\alpha(x,y)$ has a local minimum, then the intensity function $I_{\mathrm{pers}}(u,v)$ has a local maximum at (u_0, v_0), the projection of (x_0,y_0).*

Proof. The corollary follows directly from the decreasing monotonicity of the cosine at $[0, \frac{\pi}{2}]$, and Theorem 2, which establishes the relation between maxima at the two coordinate systems (x,y) and (u,v). □

Similar theorems also apply for the case of a local minimum of the intensity function. The importance of these theorems and corollaries is that they relate the intensity features detected by Y_{arg} with the geometric features of the 3D surface $z(x,y)$ at the detected locations. The fact that Y_{arg} detects certain geometric features of the 3D surface explains its robustness.

References

[1] H.J. Wolfson, Model based object recognition by 'Geometric Hashing', in: Proc. 1st Eur. Conf. on Computer Vision, Antibes, France, Springer-Verlag, Berlin, 1990, pp. 526–536.

[2] C. Tomasi, T. Kanade, Shape and motion from image streams: a factorization method—part 3: detection and tracking of point features. Technical Report CMU-CS-91-132, Carnegie Mellon University, Pittsburg, PA, USA, April 1991.

[3] T. Lindeberg, Feature detection with automatic scale selection, Internat. J. Comput. Vis. 30 (2) (1998) 79–116.

[4] C. Schmid, R. Mohr, C. Bauckhage, Evaluation of interest point detectors, Internat. J. Comput. Vis. 37 (2) (2000) 151–172.

[5] K. Mikolajczyk, C. Schmid, Indexing based on scale invariant interest points, in: Proc. 8th Internat. Conf. on Computer Vision, Vancouver, Canada, 2001, pp. 525–531.

[6] A. Tankus, Y. Yeshurun, Convexity-based visual camouflage breaking, Comput. Vis. Image Understand. 82 (3) (2001) 208–237.

[7] Y. Yeshurun, Attentional mechanisms in computer vision, in: V. Cantoni, S. Levialdi, V. Roberto (Eds.), Artificial Vision, Academic Press, New york, 1997, pp. 43–52, Chapter 2.

[8] T. Lindeberg, Detecting salient blob-like image structures and their scales with a scale-space primal sketch: a method for focus-of-attention, Internat. J. Comput. Vis. 11 (3) (1993) 283–318.

[9] O. Stasse, Y. Kuniyoshi, G. Cheng, Development of a biologically inspired real-time visual attention system, in: Seong-Whan Lee, Heinrich H. Bülthoff, Tomaso Poggio, (Eds.), Biologically Motivated Computer Vision, number 1811 in LLNCS, pp. 150–159, Seoul, Korea, Springer, May 2000.

[10] R. Deriche, G. Giraudon, Accurate corner detection: an analytical study, in: Proc. 3rd Internat. Conf. on Computer Vision, 1990, Osaka, Japan, 1990.

[11] S. Frantz, K. Rohr, H. Siegfried Stiehl. Multi-step procedures for the localization of 2D and 3D point landmarks and automatic roi size selection, in: Hans Burkhardt, Bernd Neumann (Eds.), Proc. 5th Eur. Conf. on Computer Vision, vol. 1, Freiburg, Germany, June 1998, pp. 687–703.

[12] M.A. Ruzon, C. Tomasi, Corner detection in textured color images, in: Proc. 7th IEEE Internat. Conf. on Computer Vision, vol. 2, Kerkyra, Greece, September 1999, pp. 1039–1045.

[13] R. Laganiére, Morphological corner detection, in: Proc. 6th Internat. Conf. on Computer Vision, Bombay, India, January 1998, pp. 280–285.

[14] D.G. Lowe, Object recognition from local scale-invariant features, in: Proc. 7th IEEE Internat. Conf. on Computer Vision, vol. 2, Kerkyra, Greece, September 1999, pp. 1150–1157.

[15] C. Harris, M. Stephans, A combined corner and edge detector, in: The 4th Alvey Vision Conference, 1988, pp. 147–151.

[16] S.B. Kang, R. Szeliski, H.-Y. Shum, A parallel feature tracker for extended image sequences, Comput. Vis. Image Understand. 67 (3) (1997) 296–310.

[17] Q. Zheng, R. Chellappa, Automatic feature point extraction and tracking in image sequences for arbitrary camera motion, Internat. J. Comput. Vis. 15 (1995) 31–76.

[18] V. Rehrmann, Object-oriented motion estimation in color image sequences, in: Hans Burkhardt, Bernd Neumann (Eds.), Proc. 5th Eur. Conf. on Computer Vision, vol. 1, Freiburg, Germany, June 1998, pp. 704–719.

[19] S.M. Smith, J.M. Brady, ASSET-2: real-time motion segmentation and shape tracking, Trans. Pattern Anal. Mach. Intell. 17 (8) (1995) 814–820.

[20] D. Beymer, P. McLauchlan, B. Coifman, J. Malik, A real-time computer vision system for measuring traffic parameters, in: Proc. IEEE Internat. Conf. on Computer Vision and Pattern Recognition, San Juan, PR, June 1997, pp. 495–501.

[21] S.M. Smith, J.M. Brady, SUSAN—a new approach to low level image processing, Internat. J. Comput. Vis. 23 (1) (1997) 45–78.

[22] W.Förstner, E.Gülch, A fast operator for detection and precise location of distinct points, corners and centres of circular features, in: Proc. Intercommission Conf. on Fast Processing of Photogrammetric Data, Interlaken, Switzerland, 1987, pp. 281–305.

[23] C. Harris, The DROID 3D Vision System. Technical Report 72/88/N488U, Plessey Research, Roke Manor, 1988.

[24] P. Beardsley, P.H.S. Torr, A. Zisserman, 3D model acquisition from extended image sequences, in: B. Buxton, R. Cipolla (Eds.), Proc. 4th Eur. Conf. on Computer Vision, Springer-Verlag, Berlin, 1996, pp. 683–695.

[25] L. Bretzner, T. Lindeberg, Qualitative multi-scale feature hierarchies for object tracking, in: Second International Conference on Scale-Space Theories in Computer Vision, Kerkyra, Greece, September 1999, pp. 117–128.

[26] J. Shi, C. Tomasi. Good features to track, in: IEEE Computer Society Conference on Computer Vision and Pattern Recognition, Seattle, June 1994, pp. 593–600.

[27] T. Tommasini, A. Fusiello, E. Trucco, V. Roberto, Making good features track better, in Proc. IEEE Internat. Conf. on Computer Vision and Pattern Recognition, Santa Barbara, CA, USA, June 1998.

[28] J. Verestóy, D. Chetverikov, Experimental comparative evaluation of feature point tracking algorithms, in Evaluation and Validation of Computer Vision Algorithms, 2000, pp. 183–194.

[29] Y. Song, L. Goncalves, E. Di Bernardo, P. Perona, Monocular perception of biological motion—detection and labeling, in: Proc. 7th IEEE Internat. Conf. on Computer Vision, vol. 2, Kerkyra, Greece, September 1999, pp. 805–812.

[30] P. Tissainayagam, D. Suter, Performance prediction analysis of a point feature tracker based on different motion models, Comput. Vis. Image Understand. 84 (2001) 104–125.

[31] Y.G. Leclerc, Q.-T. Luong, P. Fua, Measuring the self-consistency of stereo algorithms, in: Proc. 6th Eur. Conf. on Computer Vision, vol. 1, Dublin, Ireland, June/July 2000, pp. 282–298.

[32] A.P. Pentland, Local shading analysis, IEEE Trans. Pattern Anal. Mach. Intell. 6 (2) (1984) 170–187.

[33] R. Zhang, P.-S. Tsai, J.E. Cryer, M. Shah, Shape from shading: a survey, IEEE Trans. Pattern Anal. Mach. Intell. 21 (8) (1999) 690–705.

[34] A. Tankus Y. Yeshurun, Convexity-based camouflage breaking, in: Proc. 15th Internat. Conf. on Pattern Recognition, vol. 1, Barcelona, Spain, September 2000, pp. 454–457.

[35] B.K.P. Horn, Robot Vision, The MIT Press/McGraw-Hill Book Company, Cambridge/New york, 1986.

[36] A.M. Bruckstein, On shape from shading, Comput. Vis. Graph. Image Process. 44 (1998) 139–154.

[37] R. Kimmel, A.M. Bruckstein, Global shape from shading, Comput. Vis. Image Understand. 62 (3) (1995) 360–369.

[38] R. Kimmel, J.A. Sethian, Optimal algorithm for shape from shading and path planning, J. Math. Imaging Vis. 14 (3) (2001) 237–244.

[39] P.T. Sander, S.W. Zucker, Singularities of principal direction fields from 3D images, IEEE Trans. Pattern Anal. Mach. Intell. 14 (3) (1992) 309–317.

[40] D. Weinshall, M. Werman, On view likelihood and stability, IEEE Trans. Pattern Anal. Mach. Intell. 19 (2) (1997) 97–108.

[41] T. Tuytelaars, L. Van Gool, Matching widely separated views based on affine invariant regions, Int. J. Comp. Vis. 59 (2004) 61–85.

[42] L.S. Shapiro, Affine Analysis of Image Sequences, Cambridge University Press, Cambridge, England, 1995.