Scene-Consistent Detection of Feature Points in Video Sequences

Ariel Tankus Yehezkel Yeshurun
School of Computer Science
Tel-Aviv University
Tel-Aviv 69978, Israel
{arielt,hezy}@post.tau.ac.il

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 two other feature trackers. The suggested criteria are generic and could serve other researchers as well for performance evaluation of stable point detectors.

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 visual information processing.

Despite the large amount of work invested in detection of feature points, there is no clear definition of its goal. The "Attentional" attitude to this task (sometimes called: "Interest Points" or "Regions of Interest" detection) states that detected points should attract computational resources, as is apparently the case in biological systems [3].

A different view of the task defines it as a consistent selection of a subset of image pixels, regardless of their "attentional" value. Different names for this approach are: "Anchor Points" or "Stable Point" detection. These methods do not attempt to generally focus attention, but rather, to consistently locate image points relating to the same 3D scene points. Such points could either be used for object recognition, or as correspondence points for recovering 3D characteristics of the scene. Corners ([1], [7], [2]) and junctions ([4]) are considered Anchor Points. Other non-attentional sources of stable point detection: [5] selects stable points at maxima and minima of a Difference of Gaussian function applied in scale space. [8] uses direct gray values processing for anchor points in object recognition; see a survey of interest point detectors there.

The main goal of this paper is **robust detection of scene-consistent feature points in video sequence**s. "Robust" means consistent detection of points in noisy images, while "scene-consistent" means that the algorithm should consistently detect the same 3D scene point over multiple video frames, regardless of illumination changes, pose variations or parallax. This implies that detection which depends merely on the local geometry of scene objects would be appropriate. The intrinsic property that we use is **convexity**; we use an operator for detection of convex or concave patches in the image.

This paper is structured as follows: Section 2 sketches the operator for detection of scene-consistent points in static images, that was suggested in [10]. Section 3 then shows analytically that this method detects specific features of the image intensity function I(x,y), and proves that these image-space features correspond to the local 3D geometry of objects in the scene. These theorems are novel. They completely characterize the domains of strong (i.e., infinite) response of the operator, thus forming the theoretical basis explaining why the operator is highly robust.

Section 4 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 could not be dealt with by static images alone: parallax, camera motion and 3D object transformations. The operator copes well with these effects, because it responds to intrinsic properties of 3D objects (as Sect. 3 proves). In Sect. 5, we rigorously define two measures for evaluating tracking algorithms: completeness (w.r.t 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 two other trackers (Sect. 6). Section 7 concludes the discussion.

2. Operator for Feature Detection

In order to accomplish scene-consistent detection of feature points in video sequences, we first present an operator that has been suggested [10] for detecting points in static images. It detects convex or concave image patches. Intuitively, it looks for local "circles" where the gradient of the intensity function points outward along the whole circle. Such "circles" yield either convex or concave intensity functions. However, the operator does *not* look for these circles *explicitly*, but rather, it takes advantage of the discontinuity of the 2D $\arctan$ function for fast and robust detection of such domains. This section defines the operator.

The gradient map of an image in Cartesian coordinates is: $\nabla \mathrm{I}(x,y) = (\frac{\partial}{\partial x}\mathrm{I}(x,y),\ \frac{\partial}{\partial y}\mathrm{I}(x,y))$. In polar coordinates, the gradient argument is: $\theta(x,y) = \arg(\nabla \mathrm{I}(x,y)) = \arctan(\frac{\partial}{\partial y}\mathrm{I}(x,y),\ \frac{\partial}{\partial x}\mathrm{I}(x,y))$, where the 2D $\arctan$ function is defined by:

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

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

$$\mathbf{Y}_{arg} \overset{def}{=} \frac{\partial}{\partial y}\theta(x,y) \approx [G_\sigma(x)D_\sigma(y)] * \theta(x,y) \qquad (1)$$

where $G_\sigma(t)$ is the 1D Gaussian with mean 0, and standard deviation σ, and $D_\sigma(t) = \frac{d}{dt}G_\sigma(t)$.

Since Y_{arg} is orientation dependent, we use the isotropic version D_{arg}, which sums Y_{arg} over all orientations. The intuition behind the operator is that only specific intensity structures give rise to a zero crossing of the intensity gradient argument. In this case, the y-derivative approaches infinity due to the discontinuity ray of the 2D arctan. In practice, this appears as a strong response of Y_{arg}. An example of the domains where the strong D^2_{arg} response occurs appears in Fig. 1 (Right). In Sect. 3.1, we characterize the specific features of the intensity surface which cause an infinite response of D_{arg}, and in Sect. 3.2, we show that these intensity surface features relate to specific details of the local 3D geometry of the scene.

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.

3. 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.

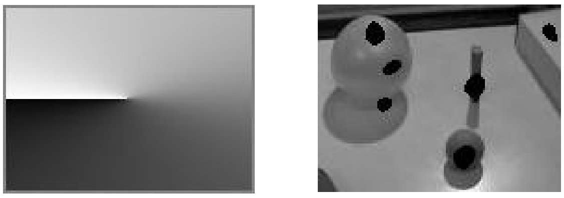

Figure 1: ***Left:*** *The 2D arctan.* ***Right:*** *An image with the areas of maximal D^2_{arg} response marked.*

3.1. Response to the Intensity Surface

We qualitatively characterize the behavior of Y_{arg} in continuous ("well-behaving") image domains, namely when the original graylevel function $f(x,y)$ is twice continuously differentiable. Our basic observation is that $\frac{\partial}{\partial y}\theta(x,y)$ approaches infinity at (x_0,y_0) due to a jump-discontinuity at $\theta(x_0,y_0)$:

1. Because $f(x,y)$, $\frac{\partial f(x,y)}{\partial x}$ and $\frac{\partial f(x,y)}{\partial y}$ are continuously differentiable, and for all points (x_0,y_0) the left- and right-hand side 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: $\frac{\partial f(x,y)}{\partial x}$ and $\frac{\partial f(x,y)}{\partial y}$ are differentiable anywhere, so at (x_0,y_0), $\arctan(y,x_0)$ is continuous. Because $\arctan(y,x_0)$ is differentiable at all its continuity points, $\theta(x,y)$ is also differentiable.
   (b) If $\theta(x,y)$ has a removable singularity: The estimation of Y_{arg} is achieved using a convolution (Eq. 1), which is an integral. The integral of a function with a removable singularity is identical to that of the fixed function (i.e., when the value at the singular point is set to create a continuous function). The result of the convolution does *not* approach infinity.
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 feature points. Formally,

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

1. *$\forall y,\ \frac{\partial f(x,y)}{\partial y}|_{x=x_0} = 0$ and $\forall y < y_0$,*
   *$\frac{\partial f(x,y)}{\partial x}|_{x=x_0} \geq 0$, and $\forall y > y_0$, $\frac{\partial f(x,y)}{\partial x}|_{x=x_0} < 0$.**
2. *$\forall y < y_0,\ \frac{\partial f(x,y)}{\partial y}|_{x=x_0} > 0$ and*
   *$\forall y > y_0,\ \frac{\partial f(x,y)}{\partial y}|_{x=x_0} = 0$, and **
   - *(a) $\forall y > y_0$, $\frac{\partial f(x,y)}{\partial x}|_{x=x_0} = 0$, or:*
   - *(b) $\forall y < y_0$, $\frac{\partial f(x,y)}{\partial x}|_{x=x_0} = 0$, or:*
   - *(c) $\forall y < y_0$, $\frac{\partial f(x,y)}{\partial x}|_{x=x_0} > 0$, and*
     *$\forall y > y_0$, $\frac{\partial f(x,y)}{\partial x}|_{x=x_0} < 0$. **
3. *$\forall y < y_0,\ \frac{\partial f(x,y)}{\partial y}|_{x=x_0} < 0$ and*
   *$\forall y > y_0,\ \frac{\partial f(x,y)}{\partial y}|_{x=x_0} = 0$, **
   except when: $\forall y : y \neq y_0$, $\frac{\partial f(x,y)}{\partial x}|_{x=x_0} > 0$.
4. *(x_0, y_0) is a local extremum of $f(x_0, y)$, except when:*
   $\forall y : y \neq y_0$, $\frac{\partial f(x,y)}{\partial x}|_{x=x_0} > 0$.

** 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.*

Due to lack of room, we will only survey the main ideas of the proof of theorem 1 (see [11] for the complete proof).

As $f(x,y)$ is twice continuously differentiable, $\frac{\partial f(x,y)}{\partial x}$ and $\frac{\partial f(x,y)}{\partial y}$ are continuously differentiable. Because the 2D $\arctan$ function is differentiable in the whole plane except for the negative x-axis and the origin (that is, differentiable at domain $\{(x,y) \in R^2 \,|\, x > 0 \text{ or } y \neq 0\}$), the composition of the functions (i.e., $\theta(x,y)$) is differentiable at $\{(x,y) \in R^2 \,|\, \frac{\partial f(x,y)}{\partial x} > 0 \text{ or } \frac{\partial f(x,y)}{\partial y} \neq 0\}$. It follows that $\frac{\partial}{\partial y}\theta(x,y) \to \pm\infty$ may hold merely in the domain $D \overset{def}{=} \{(x,y) \in R^2 \,|\, \frac{\partial f(x,y)}{\partial x} \leq 0 \text{ and } \frac{\partial f(x,y)}{\partial y} = 0\}$. However, as claimed above, only jump discontinuities would lead to $\frac{\partial}{\partial y}\theta(x,y) \to \pm\infty$. In order to get a complete and precise characterization of configurations that lead to a jump discontinuity, we assume point (x_0, y_0) is in domain D, and examine $\frac{\partial f(x,y)}{\partial x}$ and $\frac{\partial f(x,y)}{\partial y}$ in a y-neighborhood of (x_0, y_0) (we use a y-neighborhood, because we derive $\theta(x,y)$ in the y-direction). Namely, we examine the signs of $\frac{\partial f(x,y)}{\partial x}|_{x=x_0}$ and $\frac{\partial f(x,y)}{\partial y}|_{x=x_0}$ at the left- and right-hand sides of y_0, to see whether substituting them into $\arctan(y,x)$ would cause $\theta(x,y)$ to cross the discontinuity ray of $\arctan(y,x)$. If at a certain neighborhood $\theta(x,y)$ indeed crosses the discontinuity ray (=domain D), a jump discontinuity occurs there. In this case $\frac{\partial}{\partial y}\theta(x,y) \to \pm\infty$ holds. Let us examine one such a case (the same modus operandi serves at the analysis of the rest of the cases):

Let us assume (x_0, y_0) is a local minimum of $f(x_0, y)$: $\forall y : y < y_0$, $\frac{\partial f(x,y)}{\partial y} < 0$ and $\forall y : y > y_0$, $\frac{\partial f(x,y)}{\partial y} > 0$:

1. If $\forall y : y < y_0,\ \frac{\partial f(x,y)}{\partial x} \leq 0$:
   $\forall y < y_0$: $\theta(x,y) = \arctan(\frac{\partial f}{\partial y}/\frac{\partial f}{\partial x}) - \pi \leq -\frac{\pi}{2}$.
   Because $\forall y : y > y_0$, $\frac{\partial f(x,y)}{\partial y} > 0$
   (i.e., quadrants I, II), necessarily:
   For $y < y_0$: $\theta(x,y) > 0$. ⇒ Jump discontinuity.
2. If $\forall y : y > y_0,\ \frac{\partial f(x,y)}{\partial x} \leq 0$:
   $\forall y > y_0$: $\theta(x,y) = \arctan(\frac{\partial f(x,y)}{\partial y}/\frac{\partial f(x,y)}{\partial x}) + \pi \geq \frac{\pi}{2}$.
   Because $\forall y : y < y_0$, $\frac{\partial f(x,y)}{\partial y} < 0$
   (i.e., quadrants III, IV), necessarily:
   For $y < y_0$: $\theta(x,y) < 0$. ⇒ Jump discontinuity.
3. If $\forall y : y \neq y_0,\ \frac{\partial f(x,y)}{\partial x} > 0$:
   Quadrants I, IV ⇒ continuous $\theta(x,y)$ or removable singularity. [This is the “except” part of case (4) of Theorem 1.] Pay attention, that when $\forall y : y \neq y_0,\ \frac{\partial f(x,y)}{\partial x} > 0$, the 2D function $f(x,y)$ has *no* extremum (only the 1D function: $f(x_0, y)$ has). In other words, when the 2D function $f(x,y)$ has an extremum, necessarily $\frac{\partial}{\partial y}\theta(x,y) \to \pm\infty$ (only the non-“expect” cases hold).

This proves the part of case (4) in theorem 1 referring to minimum. The complete proof analyzes the rest of the cases as well. The important point to note, is that the domains are characterized according to the signs of $\frac{\partial f(x,y)}{\partial x}$ and $\frac{\partial f(x,y)}{\partial y}$; these changes of signs of derivatives occur in specific differential geometry structures of the intensity function (e.g., at extremum points). The next section would show that these specific intensity structures relate to specific 3D scene structures. Doing so would mean that certain 3D scene structures lead to certain intensity structures, which, in turn, lead to Y_{arg} approaching infinity. In other words, what we detect using Y_{arg} is certain 3D scene structures. This would explain the stability of Y_{arg}: it responds to intrinsic properties of the scene object.

3.2. Response to Local 3D Scene Structure

So far, the analysis referred merely to the connection between Y_{arg} and the intensity function. The following theorem and corollaries would establish a connection between some of the domains where Y_{arg} approaches infinity and the three dimensional scene object.

Theorem 2 *Let $R(p,q)$ be the reflectance map of a 3D surface $z(x,y)$, where $p(x,y) = \frac{\partial z(x,y)}{\partial x}$ and $q(x,y) = \frac{\partial z(x,y)}{\partial y}$. Let us assume $R(p,q)$ is differentiable at $(p(x,y),\ q(x,y))$ and $p(x,y)$ and $q(x,y)$ are differentiable w.r.t both x and y. Let us also assume that $p(x,y)$, $q(x,y)$, $\frac{\partial}{\partial x}q(x,y)$ and $\frac{\partial}{\partial y}p(x,y)$ are continuous, and these derivatives are defined in an open domain containing point (x,y).*

If at point (x,y)*:* $\frac{\partial}{\partial y}R(p,q) = 0$ *and* $\frac{\partial}{\partial x}R(p,q) < 0$ [*i.e., domain where* $\theta(x,y)$ *is discontinuous*]*, then if* $\frac{\partial^2 z(x,y)}{\partial y \partial x} \neq 0$ *or* $\frac{\partial^2 z(x,y)}{\partial y^2} \neq 0$*:*

$$\Delta(x,y) > 0 \quad \text{or} \quad \Delta(x,y) < 0$$

and if $\frac{\partial^2 z(x,y)}{\partial y \partial x} = 0$ *and* $\frac{\partial^2 z(x,y)}{\partial y^2} = 0$*:*

$$\frac{\partial^2 z(x,y)}{\partial x^2} > 0 \quad \text{or} \quad \frac{\partial^2 z(x,y)}{\partial x^2} < 0$$

where: $\Delta(x,y) = \frac{\partial^2 z(x,y)}{\partial x^2}\frac{\partial^2 z(x,y)}{\partial y^2} - \left(\frac{\partial^2 z(x,y)}{\partial x \partial y}\right)^2$ *is the discriminant of the surface:* $z(x,y)$*.* [*i.e., the 3D point is elliptic, hyperbolic or parabolic, according to the case*].

Again, the complete proof of theorem 2 can be found in [11], and next is a brief survey of the main ideas.

From the differentiability demands of the theorem and the constraints $\frac{\partial}{\partial y}R(p,q) = 0$ and $\frac{\partial}{\partial x}R(p,q) < 0$:

$$\frac{\partial R(p,q)}{\partial y} = \frac{\partial R(p,q)}{\partial p}\frac{\partial p(x,y)}{\partial y} + \frac{\partial R(p,q)}{\partial q}\frac{\partial q(x,y)}{\partial y} = 0 \tag{2}$$

$$\frac{\partial R(p,q)}{\partial x} = \frac{\partial R(p,q)}{\partial p}\frac{\partial p(x,y)}{\partial x} + \frac{\partial R(p,q)}{\partial q}\frac{\partial q(x,y)}{\partial x} < 0 \tag{3}$$

assuming $\frac{\partial}{\partial y}p(x,y) = \frac{\partial^2 z(x,y)}{\partial y \partial x} \neq 0$, dividing (2) by $\frac{\partial}{\partial y}p(x,y)$, and substituting into (3) yields:

$$\frac{\partial R(p,q)}{\partial x} = -\frac{\partial R(p,q)}{\partial q}\Delta(x,y) \bigg/ \frac{\partial q(x,y)}{\partial x} < 0$$

where $\Delta(x,y)$ is defined above. This implies: $\Delta(x,y) > 0$ or $\Delta(x,y) < 0$. The other part of the proof is similar in nature.

Corollary 1 [Qualitative Classification of Response Points] *Under the conditions of theorem 2, if at point* (x,y)*:* $\frac{\partial R(p,q)}{\partial y} = 0$ *and* $\frac{\partial R(p,q)}{\partial x} < 0$*, then if* $\frac{\partial^2 z(x,y)}{\partial y \partial x} \neq 0$ *or* $\frac{\partial^2 z(x,y)}{\partial y^2} \neq 0$*, then at this point the 3D surface* $z(x,y)$ *has either an* elliptic *point (*$\Delta(x,y) > 0$*) or a* hyperbolic *point (*$\Delta(x,y) < 0$*), and if* $\frac{\partial^2 z(x,y)}{\partial y \partial x} = 0$ *and* $\frac{\partial^2 z(x,y)}{\partial y^2} = 0$*, then at this point the 3D surface has a* parabolic *point .*

Recalling that the reflectance map of a Lambertian surface illuminated by a point light source at infinity is: $R(p,q) = \frac{1 + p_s\, p + q_s\, q}{\sqrt{1+p_s^2+q_s^2}\sqrt{1+p^2+q^2}}$, where $(-p_s,\, -q_s,\, 1)$ is the light source direction, another corollary follows directly:

Corollary 2 [Qualitative Classification of Y_{arg} Response to Lambertian Surfaces] *Let* $z(x,y)$ *be a Lambertian surface, illuminated by a point light source at infinity. If at point* (x,y)*,* Y_{arg} *approaches* $\pm\infty$ *and* $\frac{\partial}{\partial x}R(p(x,y),q(x,y)) < 0$ *there (e.g.,* (x,y) *is an extremum of* $f(x,y)$ *as a function of* y*), than point* (x,y) *is an elliptic, hyperbolic or parabolic point of surface* $z(x,y)$*.*

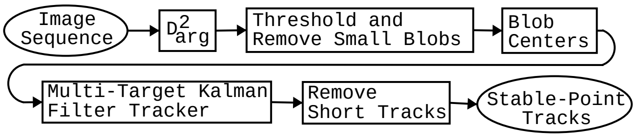

Figure 2: *Scene-consistent points detection and tracking algorithm.* ***Upper row blocks****: The stable-points locator.* ***Lower row blocks****: The tracking facility. Tracking is independent of feature detection.*

The last corollary shows, that Y_{arg} would respond to certain elliptic, hyperbolic or parabolic points on a Lambertian surface illuminated by a point light source at infinity. This establishes the connection between Y_{arg} response and the *geometric* features of the 3D scene object, leading to stability of the detected points. The discussion is incomplete without referring to specular reflection: Specular reflection indeed distract Y_{arg}, being [to a certain extent] a virtual image of the light source.

4. The Algorithm

The algorithm can be divided up into two independent parts: stable point location, and point tracking.

The **stable-point locator** is based on the D_{arg} operator: Locations where $D^2_{arg} \to \infty$ are stable, or in other words: consistently follow a 3D object. As the input is discrete and bounded, the algorithm actually looks for the maximum of D^2_{arg} (by thresholding). The stable points are the centers of gravity of the blobs of thresholded D^2_{arg}. These stable points are the only input to the point tracker: it has no knowledge about the mechanism producing its input points, or any additional knowledge of the image.

The **point tracker** is a classic multi-target 2D Kalman-filter tracker, assuming constant velocity. We compensate for the lack of a-priori knowledge of the real motion model by setting the position components of the state vector to the measurements themselves 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 locator, rather than the filtering process.

4.1. Demonstration by Video Sequences

Let us present three of the video sequences we used to test the algorithm. Tracking one of these sequences (“parking-lot”) and the well-known “Flower Garden” sequence using the suggested algorithm is presented in the attached MPEG movie. Only the interior of the marked black frame participates in the tracking (to avoid boundary conditions). Track

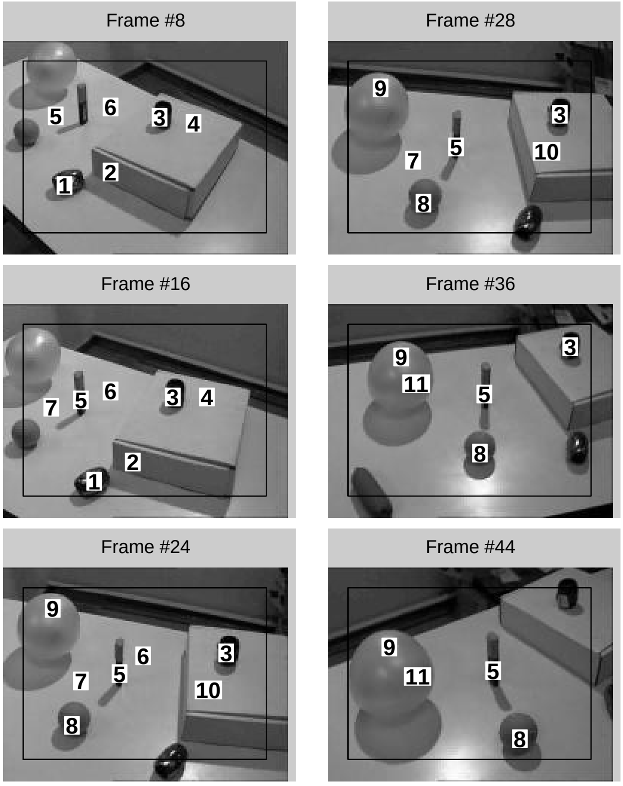

Figure 3: *Toys: Video sequence of objects in the laboratory. Note, for example, track 3 which follows the same object as long as it is in the frame, or track 8 which consistently detects the tennis ball. Some erroneous effects occur, but in general, tracking is consistent with the 3D scene.*

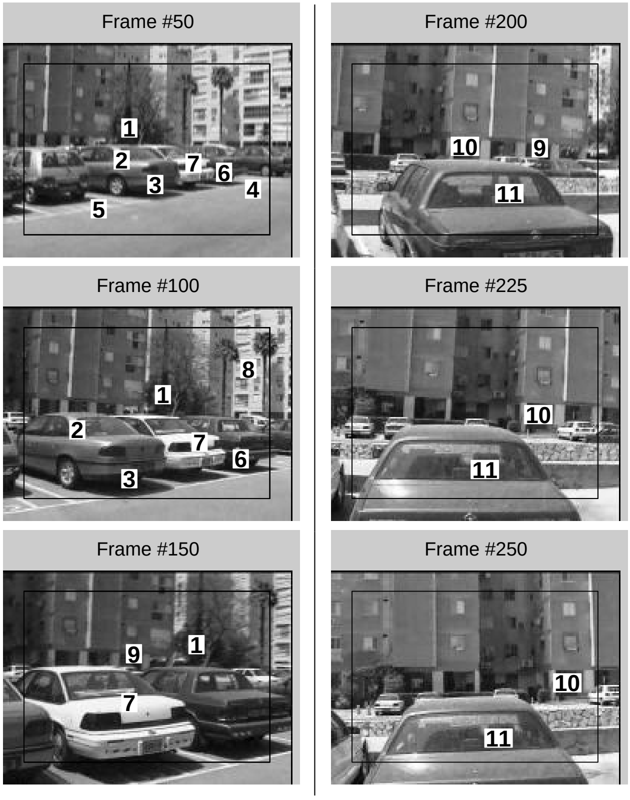

Figure 4: *Parking-lot: An outdoors video sequence.* ***Left:*** *Tracks 1 and 7 correctly tracks the tree and car despite significant scale differences and parallax.* ***Right:*** *Tracks 10 and 11 demonstrate how D_{arg} copes well with parallax.*

numbers are marked on the images. The exact feature point is the center of each label.

Figure 3 (“toys”) contains frames from a video sequence taken in the laboratory. The sequence demonstrates a notable change in viewing angle. Most of the detected points are stable, despite camera motion. Track 5 (for example) has a short erroneous detection at its beginning (for 4 frames), but for most of the sequence (37 frames out of 46) it tracks the 3D object correctly.

Figure 4 (“parking-lot”) shows frames of a video sequence of a parking lot, taken by a hand held camera. The left column exhibits consistent tracking in spite of a considerable zoom. In the right column, track 10 correctly(!) follows the background building; track 11, the car. The parallax depicted in the relative motion of these tracks could be used for 3D scene correspondence.

Figure 5 (“traffic”) samples a video of a highway. The scene is very dynamic and combines several effects: fast camera motion, scene objects motion (cars) and zooming. This yields frequent dynamic changes: scene points disappear more often, so the length of tracks is objectively limited. As expected, the results of this sequence are worse then of the previous two examples, yet still, most of the tracks are correct most of the time, even for this challenging video, and can thus serve as an input to algorithms requiring correspondence of points between successive frames.

5. Evaluating the Performance of the Algorithm

An important issue in scene-consistent point tracking is how to evaluate algorithms. The following sections define two measures: one is more relevant when the goal is maximal-time point tracking; the other, when correspondence of points in successive frames is sought. The goal of these measures is to quantify the consistency of the tracks with the 3D scene.

Scene-consistency might not be enough for certain tasks (e.g., collinear points do not fit a 3D scene reconstruction task, although they might be tracked well). To overcome this, one may apply a-priori weights to pixels in the video sequence according to the higher level task, and calculate weighted versions of the completeness and stability measures we are about to present.

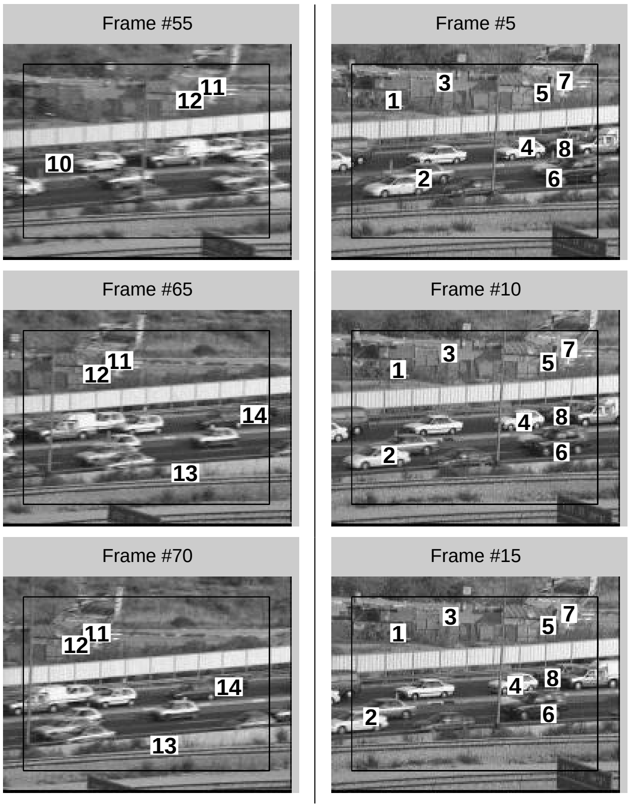

Figure 5: *Traffic: A sight of a highway.* ***Left:*** *Camera rotates clockwise; tracks 11 and 12 consistently follow scene points.* ***Right:*** *Tracks 1, 3, 5, 7 remain in place, as the background is static, while tracks 2, 4, 8 follow cars in motion. Track 6 is erroneous in this part of the sequence.*

5.1. Completeness of Tracking

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 it appears.

5.1.1 Definition of a Track

Let us define the video sequence *pixel space* as: $\Omega = N_n \times N_m \times N_k$ where each frame is of $n \times m$ pixels, and there are k frames in the sequence. Let the *set of detected points* $D \subset \Omega = N_n \times N_m \times N_k$ be the set of all pixels which the tracker selected. The *track ID function*: $t : D \longmapsto N$ defines the track ID, as determined by the tracker. Let $(a, b, c) \in D$. $\forall (a', b', c') \in D \setminus \{(a, b, c)\}$, if $t(a', b', c') = t(a, b, c)$, then $c' \neq c$. This formally states that in a certain frame, only one pixel can get a certain track ID. We say that image point set $Q = \{q_1, ..., q_\alpha\} \subseteq D$ *has track ID* T iff $t(q_1) = ... = t(q_\alpha) = T$. Let Q denote the maximal point set which has track ID T; i.e., $\forall p \in D \setminus Q,\ t(p) \neq T$.

The *set of all pixels at a distance of* ε *pixels from track* T is the maximal set of points $P^T_\varepsilon = \{p_1, ..., p_\beta\} \subset \Omega$ such that $\forall p_i = (u_1, v_1, f)\ \exists q_j = (u_2, v_2, f) \in Q : \| p_i - q_j \| \leq \varepsilon$ $(i = 1, ..., \beta,\ j = 1, ..., \alpha)$. That is, P^T_ε is the set of all image points whose distance from the point with track ID T in their frame is less than or equal to ε.

5.1.2 Selecting The Correct Scene Point of a Track

Let $\Gamma : P^T_\varepsilon \longmapsto R^3$ be a mapping which, given a point $p_i = (u_1, u_2, f) \in P^T_\varepsilon$ $(1 \leq i \leq \beta)$, returns the 3D scene point $\Gamma(p_i)$ whose projection on the image plane of frame f is p_i. We say that point Γ_1 $(\Gamma_1 \in R^3)$ is the *correct scene point of track* T, if the cardinality of the set of pixels whose 3D scene point is Γ_1: $\{p_i \in P^T_\varepsilon \,|\, \Gamma(p_i) = \Gamma_1\}$ is maximal among all scene points whose projection is at distance ε from track T ($\Gamma(P^T_\varepsilon) = \{\Gamma(p_i) \,|\, p_i \in P^T_\varepsilon\}$):

$$\| \{p_i \in P^T_\varepsilon \,|\, \Gamma(p_i) = \Gamma_1\} \| > \| \{p_j \in P^T_\varepsilon \,|\, \Gamma(p_j) = \Gamma_2\} \|$$

for all $\Gamma_2 \in R^3$. If more than one 3D scene point attains maximal cardinality, then Γ_1 is the correct scene point of track T, if its tracking in track T began earlier (i.e, at an earlier frame) than tracking Γ_2 in track T. Formally: Let

$$\| \{p_i \in P^T_\varepsilon \,|\, \Gamma(p_i) = \Gamma_1\} \| \geq \| \{p_j \in P^T_\varepsilon \,|\, \Gamma(p_j) = \Gamma_3\} \|$$

for all $\Gamma_3 \in R^3$, and let $\Gamma_2 \in R^3$ be such that:

$$\| \{p_i \in P^T_\varepsilon \,|\, \Gamma(p_i) = \Gamma_1\} \| = \| \{p_j \in P^T_\varepsilon \,|\, \Gamma(p_j) = \Gamma_2\} \|$$

Let us take a look only at frames where the projections of Γ_1 and Γ_2 (i.e., points p_i, p_j) are different (=their distance $>$ 2ε). Formally, we assume that $\exists f_i, f_j : \exists p_i = (u_1, v_1, f_i) \in P^T_\varepsilon$, $\Gamma(p_i) = \Gamma_1$, $\exists p_j = (u_2, v_2, f_j) \in P^T_\varepsilon$, $\Gamma(p_j) = \Gamma_2$, so that $\| p_i - proj_{f_i}(\Gamma(p_j)) \| > 2\varepsilon$, where $proj_f : R^3 \longmapsto R^2$ is the projection transformation of a 3D scene point onto the plane of frame f. Under this assumption, we say that Γ_1 is the correct point iff:

$$\min\{f_i \,|\, p_i = (u_i, v_i, f_i) \in P^T_\varepsilon \text{ and } \Gamma(p_i) = \Gamma_1\} < \min\{f_j \,|\, p_j = (u_j, v_j, f_j) \in P^T_\varepsilon \text{ and } \Gamma(p_j) = \Gamma_2\}$$

This definite minimum is assured to exist because in every frame, track T has at most one detected point.

Intuitively, if two scene points were allocated the same track ID T for the same (maximal) time, we choose the scene point whose tracking began earlier in the video sequence to be the correct scene point.

As a minor case, if for all f_i, f_j so that $\exists p_i = (u_1, v_1, f_i) \in P^T_\varepsilon$, $\Gamma(p_i) = \Gamma_1$ and $\exists p_j = (u_2, v_2, f_j) \in P^T_\varepsilon$, $\Gamma(p_j) = \Gamma_2$, the points are close enough:
$\| p_i - proj_{f_i}(\Gamma(p_j)) \| < 2\varepsilon$, then one may arbitrarily (but consistently) select whether Γ_1 or Γ_2 is the correct point (as their projections are close enough). For example, one may always choose the scene point whose projection on the earliest frame where the point is being tracked has lowest y-rate, and if the y-rates are equal, lowest x-rates.

5.1.3 Defining Completeness

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

$$completeness_T \overset{def}{=} 100\times\frac{\text{Actual Correct Track Time}}{\text{Potential Track Time}} =$$

$$= 100\times\frac{\| \{f_i \mid \exists p_i = (u_i, v_i, f_i),\ p_i \in P_\varepsilon^T \text{ and } \Gamma(p_i) = \Gamma_1\} \|}{\| \{p \in \Omega \mid \Gamma(p) = \Gamma_1\} \|}$$

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.

5.2. Stability of Tracking

For many practical purposes (e.g., the correspondence problem), a full tracking of 3D points is not a must. In such applications, we look for a reliable association of several feature points in one frame with the points in the successive frame, which are the projection of the same scene point. When associating points in the successive frame with points in the frame following it, the set of scene points the association refers to, might change. This leads to the stability criterion.

Let us examine a pair of successive frames: $f_i,\ f_{i+1}$. W.L.O.G, let $1, ..., r$ denote all track IDs which are common to both frames. Let $p_1^i, ..., p_r^i$ and $p_1^{i+1}, ..., p_r^{i+1}$ be the detected points for the corresponding tracks and frames. Let $proj_f\ :\ R^3 \longmapsto R^2$ be the projection transformation of a 3D scene point onto the plane of frame f. The *stability measure of frames* f_i *and* f_{i+1} *with allowed noise of* ε *pixels* is:

$$stability_\varepsilon(i,\ i+1) \overset{def}{=}$$

$$= 100\times\frac{|\ \{j \in \{1, \ldots, r\}\ :\| \ proj_{i+1}(\Gamma(p_j^i)) - p_j^{i+1} \ \| \leq \varepsilon\}\ |}{r}$$

(This measure resembles the "repeatability" measure of [8]; there, however, only the detection part is handled, thus implicitly assuming a correct association of corresponding points (i.e., tracking)).

6. Experimental Results

Various feature trackers have been suggested in the literature; examples are: [12], [9], [6]. In order evaluate the performance of our tracker, we compare our D_{arg}-based tracking algorithm with two other algorithms: Junction Detection [4] and KLT [9].

Junctions are detected in [4] according to the curvature of the level curves of the intensity function, multiplied by the gradient magnitude raised to the power of three. Scale is automatically selected by normalizing the derivatives. We track the junctions of [4] using a Kalman-filter tracker.

The KLT (Kanade-Lucas-Tomasi) tracking algorithm[1] [9] is based on a model of affine image change. Features are selected to maximize tracking quality. Monitoring tracking quality is based on a measure of dissimilarity.

In all trackers, tracks shorter then a certain percentage of video sequence length are ignored: 25% for the toys and parking-lot sequences, and 10% for the traffic sequence (the traffic video sequence has a higher variability). Identical thresholds were set for all trackers.

6.1. Completeness Comparison

Figure 6 (Left) shows graphs of the completeness measure for the toys, parking-lot and traffic sequences, for each of the three algorithms.

In order to follow the development in time of the completeness measure, a sliding window over frames in the video sequence is employed. The window length is: 30 frames; it shifts by 5 frames each time. The allowed noise level in all sequences is: $\varepsilon = 3$ pixels.

The graphs show that the completeness of D_{arg} is at least comparable to that of the other two trackers. For the toys sequence, in part of the sequence D_{arg} performs better than the other two trackers. For the parking-lot sequence, D_{arg} performs significantly better than the other two trackers, especially at the last part of the sequence, where its ability to cope with parallax is displayed. For the traffic sequence, the three trackers attain similar results.

6.2. Stability Comparison

Figure 6 (Right) introduces the stability criterion for the three trackers on the three video sequences. The graphs show the sliding average stability over windows of 30 frames, shifted by 5 frames each time. The allowed noise level in all sequences is: $\varepsilon = 3$ pixels. As the graphs show, the stability of D_{arg} is higher than that of either Junction Detection or KLT for the toys and parking-lot sequences. In parts of the parking-lot sequence, KLT equates with D_{arg}. For the traffic sequence, KLT performs better than D_{arg}, and D_{arg} performs better than Junction Detection. In parts of this sequence D_{arg} equates with KLT.

We see that D_{arg} is more stable than Junction Detection, and sometimes (toys seq.) also more than KLT; Sometimes (parking-lot and traffic seq.) D_{arg} and KLT equate. One should also take into account the fact that the performance of D_{arg} in terms of stability is not at the expense of completeness, as D_{arg} maintains its level of completeness of tracking at least comparable to the other trackers (sometimes even a better completeness).

[1] Implementation: Stan Birchfield, Stanford Univ., ver. 1.1.5, 7/10/98.

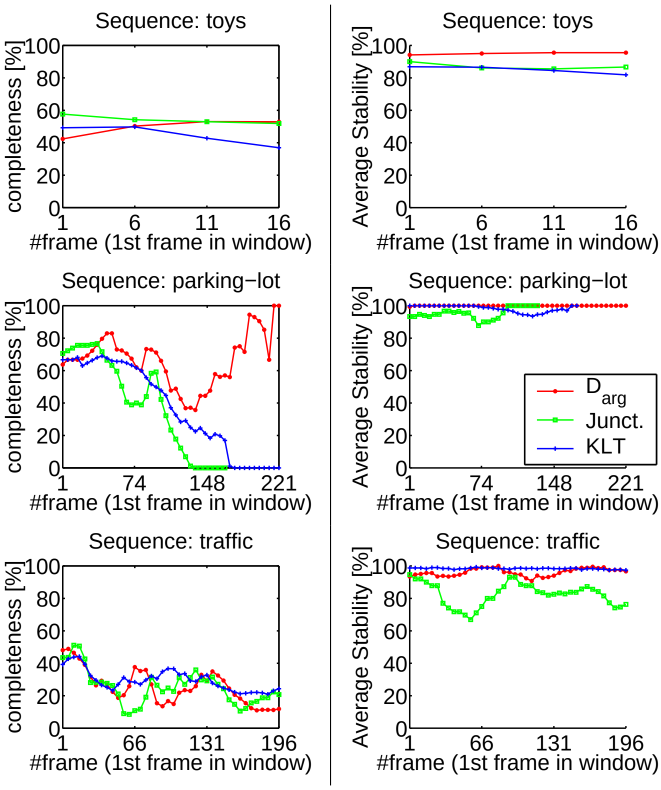

Figure 6: ***Left:*** *Completeness comparison.* D_{arg} *performs as well as the other two trackers on the toys and traffic sequences, and better on the parking-lot sequence.* ***Right:*** *Stability comparison. Tracking by* D_{arg} *is more stable than Junction Detection. For the toys sequence,* D_{arg} *is more stable than KLT. For the parking-lot and traffic sequences,* D_{arg} *and KLT equate.*

6.3. No-Tracking Comparison

Our last 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, for each of the three trackers. The total length of the 3 video sequences is: $46 + 252 + 227 = 525$ frames.

D_{arg} achieves the minimal no-track time: only 4 frames without any tracking in all three video sequences. This no-track time is significantly less then that of the other two methods (KLT: 81 frames; Junction Detection: 121 frames).

7. Conclusions

We have presented a convexity-based method for scene-consistent feature points detection in video sequences. 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 object tracking as well as correspondence establishing tasks. These measures overcome various flaws in existing evaluation measures of feature point trackers. We have used them in a comparison of our tracker with two other 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).

The suggested measures are generic; they can serve as a basis for comparison of 3D point tracking algorithms for other researchers as well.

References

[1] S. Frantz, K. Rohr, and H. S. Stiehl. Multi-Step Procedures for the Localization of 2D and 3D Point Landmarks and Automatic ROI Size Selection. In *European Conference on Computer Vision*, pp. 687–703, Germany, 1998.

[2] R. Laganière. Morphological Corner Detection. In *Intl. Conf. on Computer Vision*, pp. 280–285, Bombay, India, 1998.

[3] T. Lindeberg. Detecting Salient Blob-Like Image Structures and Their Scales with a Scale-Space Primal Sketch: A Method for Focus-of-Attention. *International Journal of Computer Vision*, 11(3):283–318, 1993.

[4] T. Lindeberg. Feature Detection with Automatic Scale Selection. *Intl. Journal of Computer Vision*, 30(2):79–116, 1998.

[5] D. G. Lowe. Object Recognition from Local Scale-Invariant Features. In *International Conference on Computer Vision*, pp. 1150–1157, Kerkyra, Greece, 1999.

[6] V. Rehrmann. Object-Oriented Motion Estimation in Color Image Sequences. In *European Conference on Computer Vision*, pp. 704–719, Germany, 1998.

[7] M. A. Ruzon and C. Tomasi. Corner detection in textured color images. In *International Conference on Computer Vision*, pp. 1039–1045, Kerkyra, Greece, 1999.

[8] C. Schmid, R. Mohr, and C. Bauckhage. Evaluation of Interest Point Detectors. *International Journal of Computer Vision*, 37(2):151–172, 2000.

[9] J. Shi and C. Tomasi. Good Features to Track. In *Intl. Conf. on Comp. Vision and Pat. Recog.*, pp. 593–600, Seattle, 1994.

[10] A. Tankus and Y. Yeshurun. Convexity-Based Visual Camouflage Breaking. *Computer Vision and Image Understanding*, 82(3):208–237, June 2001.

[11] A. Tankus and Y. Yeshurun. Detection of Scene-Consistent Points in Video Sequences. *Technical Report, TR-CS-200102*, 2001.

[12] Q. Zheng and R. Chellappa. Automatic Feature Point Extraction and Tracking in Image Sequences for Arbitrary Camera Motion. *Intl. Journal of Computer Vision*, 15:31–76, 1995.