# Real-Time Feature Tracking and Outlier Rejection with Changes in Illumination

Hailin Jin
Washington University
Electrical Engineering
Campus Box 1127
St. Louis, MO 63130
hljin@ee.wustl.edu

Paolo Favaro
Washington University
Electrical Engineering
Campus Box 1127
St. Louis, MO 63130
fava@ee.wustl.edu

Stefano Soatto
University of California, Los Angeles
Computer Science
Los Angeles, CA 90095, and
Washington University, St.Louis
soatto@ucla.edu, soatto@ee.wustl.edu

## Abstract

*We develop an efficient algorithm to track point features supported by image patches undergoing affine deformations and changes in illumination. The algorithm is based on a combined model of geometry and photometry that is used to track features as well as to detect outliers in a hypothesis testing framework. The algorithm runs in real time on a personal computer, and is available to the public.*

## 1 Introduction

Tracking the deformation of image regions has proven to be an essential component of vision-based systems in a variety of applications ranging from control systems [7] to human-computer interactions [3], medical imaging [1, 8] and mosaicing, just to mention a few. In visual tracking the main interest goes to establish region correspondences between images obtained from a moving camera. Once correspondence has been established, the temporal evolution of the deformation of each region can be used, for instance, as a combined measurement of motion and structure of the scene. To this end it is important for features to be tracked reliably for as long as possible. The longer the baseline and the smaller the error, the more accurate the reconstruction [2]. A popular technique for visual tracking on unstructured scenes is to minimize the sum of squared differences of images intensities, usually referred to as SSD matching. Much work has been based on this principle, starting with the pioneering work of Lucas and Kanade [6] that establishes matching between frames adjacent in time. As Shi and Tomasi note [9], interframe matching is not adequate for applications where the correspondence for a finite size image patch over a long time span is needed. Indeed, interframe tracking is prone to cumulative error when trajectories are integrated over time. On the other hand, when considering matching over long time spans, the geometric deformations of image regions become significant and more complex models are necessary. Shi and Tomasi show that the affine transformation is a good tradeoff among model complexity, speed and robustness.

However, SSD or correlation-based tracking algorithms usually assume that the changes in the scene appearance are only due to geometric deformations. Thus, when changes in illumination are relevant, these approaches tend to perform poorly. Hager and Belhumeur in [4] describe an SSD-based tracker that compensates for illumination changes. Their approach is divided into three phases: first a target region is defined, then a basis of reference templates for the illumination change is acquired and stored, and finally tracking is performed based on the prior knowledge of the templates.

In order to decide whether tracking of a feature is feasible or not, it is necessary to monitor its quality along the sequence. Shi and Tomasi [9] proposed a rule based on the image residual, which they called "dissimilarity", that discards features when the estimation of the displacement parameters cannot be performed reliably. Along this direction Tommasini et al. [10] proposed a robust method to detect and reject outliers. They use the X84 rule borrowed from robust statistics, which achieves robustness employing median and median deviation instead of the usual mean and standard deviation.

As computers are becoming more and more powerful, a growing range of applications can be implemented in real-time with all the benefits that follow. For instance, structure from motion algorithms have been implemented as fast as 30 frames per second [5]. Such systems use feature tracking as measurements and hence speed is of paramount importance. Furthermore, visual trackers that do not rely on prior knowledge on the structure or motion of the scene and are insensitive to environmental changes, for instance illumination, open to a wide variety of applications.

In this paper we propose a system that performs real-time visual tracking in the presence of illumination changes. No off-line computations or prior assumptions are made either

on structure or illumination of the scene. Finally, we provide a principled method to reject outliers that do not fit, for instance at occluding boundaries.

## 2 Image deformation models

### 2.1 Geometry

Let $\mathbf{X}$ be the coordinate of a point $P$ on a surface $S$ in the scene. Let $\mathbf{x} = \pi(\mathbf{X})$ be the projection of $P$ on the image plane, where $\pi$, depending on the imaging process, can be a perspective projection: $\pi(\mathbf{X}) = [\frac{X_1}{X_3}\ \frac{X_2}{X_3}]^T$, or a spherical projection: $\pi(\mathbf{X}) = \frac{\mathbf{X}}{\|\mathbf{X}\|}$. We will not make distinctions between the homogeneous coordinates $\mathbf{x} = [x\ y\ 1]^T$ and the 2-D coordinates $[x\ y]^T$. $I(\mathbf{x}, t)$ denotes the intensity value at the location $\mathbf{x}$ of an image acquired at time $t$. Away from discontinuities in $S$, generated for example by occluding boundaries, the deformations of the images of $S$ can be described as image motion:

$$I(\mathbf{x}, 0) = I(g(\mathbf{x}), t) \quad \forall\, \mathbf{x} \in \mathcal{W} \tag{1}$$

where $\mathcal{W}$ is the region of interest in the image, and $g(\cdot)$ is, in general, a nonlinear time-varying function which depends on an infinite number of parameters (the surface $S$):

$$g(\mathbf{x}) = \pi(R(t)\mathbf{x}\rho + T(t)) \ \text{ with } \rho \mid \mathbf{x}\rho = \mathbf{X} \in S \tag{2}$$

where $(R(t), T(t)) \in SE(3)$ is a rigid change of coordinates between the inertial reference frame, that we choose to coincide with the camera reference system at time 0, and the moving reference frame (at time $t$).

Clearly, having real-time operation in mind, we need to restrict the class of deformations to a finite-dimensional one that can be easily computed. The most popular feature tracking algorithms rely on a purely translational model:

$$g(\mathbf{x}) = \mathbf{x} + d \qquad \forall\, \mathbf{x} \in \mathcal{W}_T \tag{3}$$

where $\mathcal{W}_T$ is a window of a certain size, and $d$ is the 2-D displacement of $\mathbf{x}$ on the image plane. This model results in very fast algorithms [6], although for it to be valid one has to restrict the size of the window, thereby losing the beneficial effects of averaging. Typical sizes for windows range from $3 \times 3$ to $7 \times 7$, depending on the complexity of the scene, the sample rate of the frame grabber and the resolution of the frame grabber, beyond which the model is easily violated after a few frames. Therefore, a purely translational model is only valid locally in space and time. A richer model can be obtained by considering Euclidean transformations of the plane, i.e. $g(\mathbf{x}) = U\mathbf{x} + d$ where $U \in SO(2)$ describes a rotation on the plane. A slightly richer model where the linear term is not restricted to be a rotation, is an affine transformation:

$$g(\mathbf{x}) = A\mathbf{x} + d \qquad \forall\, \mathbf{x} \in \mathcal{W}_A \tag{4}$$

where $A \in \mathcal{GL}(2)$ is a general linear transformation of the plane coordinates $\mathbf{x}$, $d$ is the 2-D displacement, and $\mathcal{W}_A$ is the window of interest. This affine model has been proposed and tested by Shi and Tomasi [9].

Because of image noise, the equation (1) in general does not hold exactly. If the motion function $g(\cdot)$ can be approximated by a finite set of parameters $\alpha$, then the problem can be posed as to determine $\widehat{\alpha}$ such that:

$$\widehat{\alpha} = \arg\min_{\alpha} \int_{\mathcal{W}} \|I(\mathbf{x}, 0) - I(g_\alpha(\mathbf{x}), t)\| d\mathbf{x}. \tag{5}$$

for some choice of norm $\|\cdot\|$, where $\alpha = \{d\}, \{A, d\}$ in the translational and affine model respectively. Notice that the residual to be minimized is computed in the measurements (i.e. image intensity).

### 2.2 Photometry

In real environments brightness or contrast changes are unavoidable phenomena that cannot always be controlled. It follows that modeling light changes is necessary for visual trackers to operate in a general situation.

Consider a light source $\mathcal{L}$ in the 3-D space and suppose we are observing a smooth surface $S$. As Figure 1 explains,

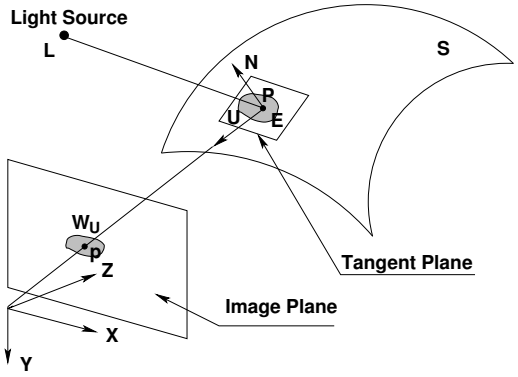


**Figure 1.** *Image formation process when illumination is taken into account: the intensity value at p on the image plane depends in general on the light $\mathcal{L}$ distribution, the observer position, the surface normal $N$ at $P$ and the albedo function $E$ of $S$.*

the intensity value of each point on the image plane depends on the portion of incoming light from the source $\mathcal{L}$ that is reflected by the surface $S$, and is described by the *Bidirectional Reflectance Distribution Function* (BRDF). When the light source $\mathcal{L}$ is far enough from the surface $S$, the incident light rays are approximately parallel. In a similar manner, if the observer is far enough from the surface $S$, the viewing angle for each point on the surface can be approximated

with a constant. Finally, we assume that for a point $P$ on the smooth surface $S$, it is possible to consider a neighborhood $U$ around $P$ such that normal vectors to $S$ do not change within $U$, i.e. in $U$ the surface is a plane.

Under the above assumptions, and assuming that the surface is Lambertian, the BRDF simplifies considerably and the intensity observed at the point $\mathbf{x}$ can be modeled as:

$$I(\mathbf{x}) = \lambda_E E(\mathbf{X}) \qquad \forall\, \mathbf{x} \in \mathcal{W}_U \tag{6}$$

where $E$ is the albedo function of $S$, $\mathcal{W}_U = \pi(U)$ and $\lambda_E$ is constant and depends on the angle between the incident light direction and the surface normal. On the other hand, due to the camera automatic gain (acting on the brightness parameter) or to reflections coming from neighboring objects, it is necessary to introduce an additive term in the equation (6) to take into account for these effects. Therefore, a more appropriate model turns out to be:

$$I(\mathbf{x}) = \lambda_E E(\mathbf{X}) + \delta_E \qquad \forall\, \mathbf{x} \in \mathcal{W}_U \tag{7}$$

where $\delta_E$ is constant for any $\mathbf{x} \in \mathcal{W}_U$. $\lambda_E$ and $\delta_E$ can be thought as parameters that represent respectively the contrast and brightness changes of the image. When either the camera or the scene is subject to motion, these parameters will change and so will $\lambda_E$ and $\delta_E$. We define the following as our model for illumination changes:

$$I(\mathbf{x}, 0) = \lambda(t) I(g(\mathbf{x}), t) + \delta(t) \qquad \forall\, \mathbf{x} \in \mathcal{W}_U \tag{8}$$

where $\lambda(t)$ and $\delta(t)$ are defined as:

$$\begin{aligned} \lambda(t) &= \tfrac{\lambda_E(t)}{\lambda_E(0)} \\ \delta(t) &= \delta_E(t) - \tfrac{\lambda_E(t)}{\lambda_E(0)} \delta_E(0) \qquad t > 1. \end{aligned}$$

### 2.3 Computing geometric and photometric parameters

The combination of the geometry and photometry gives the following:

$$I(\mathbf{x}, 0) = \lambda(t) I(A(t)\mathbf{x} + d(t), t) + \delta(t) \qquad \forall\, \mathbf{x} \in \mathcal{W}_U. \tag{9}$$

Because of image noise and because both the affine motion model and the affine illumination model are approximations, equation (9) in general does not hold exactly. Therefore, we pose the problem as an optimization problem: find the parameters $A$, $d$, $\lambda$ and $\delta$ that minimize the following discrepancy:

$$\epsilon = \int_{\mathcal{W}_U} [I(\mathbf{x}, 0) - (\lambda I(A\mathbf{x} + d, t) + \delta)]^2 w(\mathbf{x}) d\mathbf{x} \tag{10}$$

where $w(\cdot)$ is a weight function. Note that to simplify the notations, we have dropped the time index $t$ for the parameters. In the simplest case, $w(\mathbf{x}) \equiv 1$. However, in general, the shape of $w(\cdot)$ depends on the application. For instance, it can be a bell-like function to emphasize the window center. To carry out the minimization, we approximate the modeled intensity using a first-order Taylor expansion around:

$$A = I_d \;\; d = 0 \;\; \lambda = 1 \;\; \delta = 0. \tag{11}$$

We have:

$$\lambda I(A\mathbf{x} + d, t) + \delta \simeq \lambda I(\mathbf{x}, t) + \delta + \nabla I \frac{\partial \mathbf{y}}{\partial u}(u - u_0) \tag{12}$$

where $\nabla I$ is the gradient of the image intensity computed at $\mathbf{x}$, $\mathbf{y} = A\mathbf{x} + d$. $u$ collects the geometric parameters $A$ and $d$: $u = [a_{11}\ a_{12}\ a_{21}\ a_{22}\ d_1\ d_2]$ and $u_0 = [1\ 0\ 0\ 1\ 0\ 0]$, where $A = \{a_{ij}\}$, $d = [d_1\ d_2]^T$. $\frac{\partial \mathbf{y}}{\partial u}$ is the derivative of $\mathbf{y}$ with respect to $u$. Rewriting equation (12) in matrix form, we have:

$$I(\mathbf{x}, 0) = F^T(\mathbf{x}, t) z \tag{13}$$

where $F(\mathbf{x}, t) = [xI_x\ yI_x\ xI_y\ yI_y\ I_x\ I_y\ I\ 1]^T$, $z = [a_{11}\ a_{12}\ a_{21}\ a_{22}\ d_1\ d_2\ \lambda\ \delta]^T$ and $x$ and $y$ are the coordinates of $\mathbf{x}$.

The problem reduces to determining $z$ for each patch. Multiplying equation (13) by $F^T(\mathbf{x}, t)$ on both sides, and integrating over the whole window $\mathcal{W}_U$ with the weight function $w(\cdot)$, we have the following linear $8 \times 8$ system:

$$Sz = a \tag{14}$$

where

$$a = \int_{\mathcal{W}_U} F^T(\mathbf{x}, t) I(\mathbf{x}, 0) w(\mathbf{x}) d\mathbf{x} \tag{15}$$

and

$$S = \int_{\mathcal{W}_U} F^T(\mathbf{x}, t) F(\mathbf{x}, t) w(\mathbf{x}) d\mathbf{x}. \tag{16}$$

If we consider the pixel quantization, the integral becomes a summation. We write $S$ in a block-matrix form:

$$S = \sum_{\mathbf{x} \in \mathcal{W}_U} \begin{bmatrix} T & U \\ V & W \end{bmatrix} w(\mathbf{x}) \tag{17}$$

where

$$T = \begin{bmatrix} x^2 I_x^2 & xyI_x^2 & x^2 I_x I_y & xyI_x I_y & xI_x^2 & xI_x I_y \\ xyI_x^2 & y^2 I_x^2 & xyI_x I_y & y^2 I_x I_y & yI_x^2 & yI_x I_y \\ x^2 I_x I_y & xyI_x I_y & x^2 I_y^2 & xyI_y^2 & xI_x I_y & xI_y^2 \\ xyI_x I_y & y^2 I_x I_y & xyI_y^2 & y^2 I_y^2 & yI_x I_y & yI_y^2 \\ xI_x^2 & yI_x^2 & xI_x I_y & yI_x I_y & I_x^2 & I_x I_y \\ xI_x I_y & yI_y I_x & xI_y^2 & yI_y^2 & I_x I_y & I_y^2 \end{bmatrix} \tag{18}$$

$$V^T = U = \begin{bmatrix} xI_x I & xI_x \\ yI_x I & yI_x \\ xI_y I & xI_y \\ yI_y I & yI_y \\ I_x I & I_x \\ I_y I & I_y \end{bmatrix} \tag{19}$$

and

$$W = \begin{bmatrix} I^2 & I \\ I & 1 \end{bmatrix}. \tag{20}$$

$T$ is the matrix computed in the algorithm of Shi and Tomasi, which is based on geometry only. $W$ comes from our model of photometry. $U$ and $V$ are the cross terms between geometry and photometry. Finally, when $S$ is invertible, $z$ can be computed as:

$$z = S^{-1}a. \tag{21}$$

From equation (21), one can compute all the parameters. However, it will only give a rough approximation for $z$ because of the first-order approximation in equation (12). To achieve a higher accuracy one can, for example, employ a Newton-Raphson-style iteration. This can be done by approximating equation (9) around the previous solution, and iterating equation (21) until the variation in all the parameters is negligible. Note that, a simple implementation of Newton-Raphson minimization algorithm would have involved the Hessian matrix of the cost function, which requires second derivatives of image intensities. In our minimization procedure, one does not need to compute the Hessian matrix. It has been noticed experimentally that this modification improves speed and robustness of the minimization algorithm.

## 3 Hypothesis test-based outlier rejection

To decide whether features are being tracked successfully or not, we could examine the value of the discrepancy (10) between the intensities of the image patch at time $t_0$ and the reconstruction at time $t_0$ from the image patch at time $t$. However, such discrepancy function does not compensate for differences in the intensity variation among the patches of interest. A patch with high variation gives high residual because of pixel quantization and interpolation during the matching. A suitable discrepancy function turns out to be the normalized cross-correlation. Hence, our rejection rule discards features whose normalized cross-correlation falls below a fixed threshold. Tipical values range from $0.80$ to $0.95$.

Another practical issue to consider is the evaluation of the information content of an image patch. When a patch shrinks significantly along one or both directions, the information it carries might become meaningless to the purposes of the minimization. Based on this reasoning, we introduce another monitoring scheme: let $\mathcal{W}_I(t_0)$ and $\mathcal{W}_I(t)$ be the window for a patch $I$ at time $t_0$ and $t$ respectively; compute the ratio between the area or $\mathcal{W}_I(t)$ and the area of $\mathcal{W}_I(t_0)$. We discard the patch $I$ if the computed ratio falls below a threshold with value between 0 (the feature is no longer visible) and 1 (the areas are identical).

## 4 Experiments

Figure 2 shows 8 images from a sequence of about 400 frames. The scene consists of a box rotating along the vertical axis. The box first rotates from left to right, then it comes back to the initial position. As it can be seen, during the rotation the illumination changes substantially.

Figure 4 shows the residuals of our algorithm versus Shi-Tomasi. Figure 3 shows the evolution of a selected patch. The top eight images are the views at different time instants. The sequence in the middle is the reconstruction of the patch at time $t_0$ using the geometric parameters estimated by Shi-Tomasi's algorithm. The bottom eight images are the reconstructed patches based on our estimation. Note that not only the appearance is estimated correctly, but also the change in illumination is compensated for. Figure 5 shows the estimated $\lambda$ (image contrast) and $\delta$ (image brightness). Both estimates come back to the initial state, as expected. In this test, the Shi-Tomasi tracker cannot track this patch after approximately 200 frames.

A second set of experiments is devoted to show our outlier rejection rule based on the residual. Figure 6 shows the setting for this experiment, where, among the others, it has been chosen a patch (number 4) that will be occluded during the motion. Figure 7 shows the evolution of the residuals for 5 selected patches using both Shi-Tomasi and our algorithm. As one can see, residuals can increase for different reasons. In particular feature 3 is correctly tracked by Shi-Tomasi tracker until frame 200, but its residual is comparable to feature 4 that is an outlier. Therefore, the usual outlier rejection rule would discard also those features that are instead valid ones.

We implement our algorithm on a personal computer. The code and detailed documentation are available at `http://www.ee.wustl.edu/~hljin/research`. In our test (on a 1GHz PIII computer), the program can track 40 patches of size $7 \times 7$ pixels in 15 milliseconds.

## 5 Conclusions

We have presented an extension of the algorithm of Shi-Tomasi to take into consideration changes in illumination and reflection. The estimation of parameters is done using an iterative optimization scheme. The computational cost is low and the algorithm has been implemented in real-time. We have made our real-time implementation (both code and documentation) available to the public. We tested our algorithm on real images and the experimental results show that our model is accurate enough to keep tracking under significant changes in illumination; furthermore, we showed that in real environments taking into account for light changes

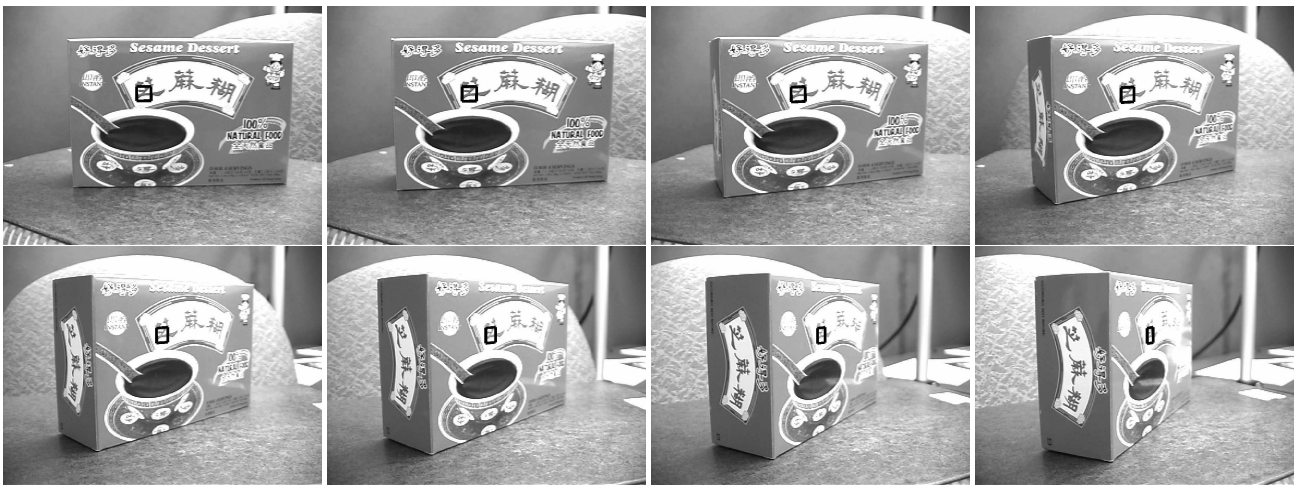


**Figure 2.** *Eight images from the* sesame *sequence. The superimposed squares show the regions of interest.*

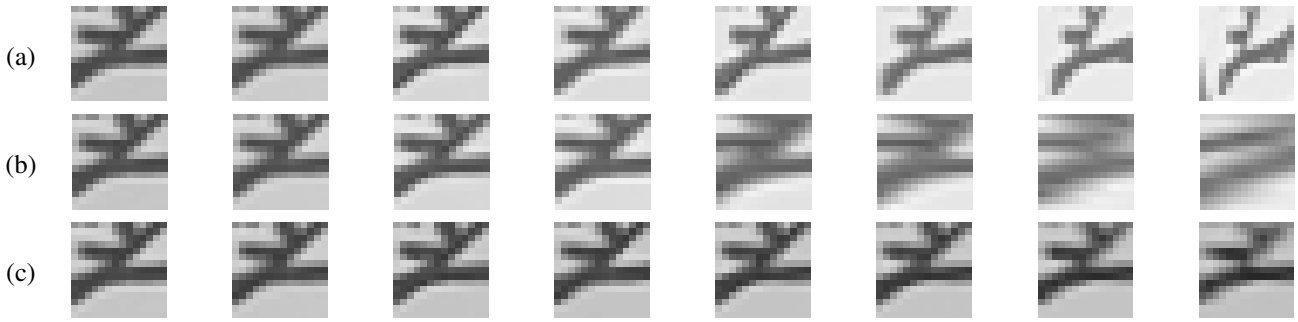


**Figure 3.** *Some snapshots of the region of interest from the* sesame *sequence: (a) original sequence as it evolves in time; (b) reconstruction of the initial patch using Shi-Tomasi algorithm; (c) reconstruction of the initial patch using the illumination-invariant tracker. As it can be seen the illumination change leads Shi-Tomasi to not converge properly. Our method maintains the appearance of the patch constant throughout the sequence.*

is necessary in order to track longer. Moreover, the computed residual is invariant to changes in illumination, which allows to monitor point features and to reject outliers correctly.

**Acknowledgements**

This research is supported in part by ARO grant DAAD19-99-1-0139 and Intel grant 8029.

## References

[1] E. Bardinet, L.D. Cohen, and N. Ayache. Tracking medical 3d data with a deformable parametric model. In *European Conference on Computer Vision*, 1997.

[2] A. Chiuso, P. Favaro, H. Jin, and S. Soatto. "mfm": 3-d motion from 2-d motion causally integrated over time. In *European Conference on Computer Vision*, 2000.

[3] T.J. Darrell, B. Moghaddam, and A.P. Pentland. Active face tracking and pose estimation in an interactive room. In *IEEE Computer Vision and Pattern Recognition*, 1996.

[4] G.D. Hager and P.N. Belhumeur. Efficient region tracking with parametric models of geometry and illumination. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 20(10):1025–1039, October 1998.

[5] H. Jin, P. Favaro, and S. Soatto. Real-time 3-d motion and structure of point features: Front-end system for vision-based control and interaction. In *IEEE Computer Vision and Pattern Recognition*, 2000.

[6] B.D. Lucas and T. Kanade. An iterative image registration technique with an application to stereo vision. In *International Joint Conference on Artificial Intelligence*, pages 674–679, 1981.

[7] N.P. Papanikolopoulos, P.K. Khosla, and T. Kanade. Visual tracking of a moving target by a camera mounted on a robot: A combination of control and vision. *IEEE Trans. Robotics and Automation*, 9:14–35, 1993.

[8] P. Shi, G. Robinson, T. Constable, A. Sinusas, and J.S. Duncan. A model-based integrated approach to track myocardial deformation

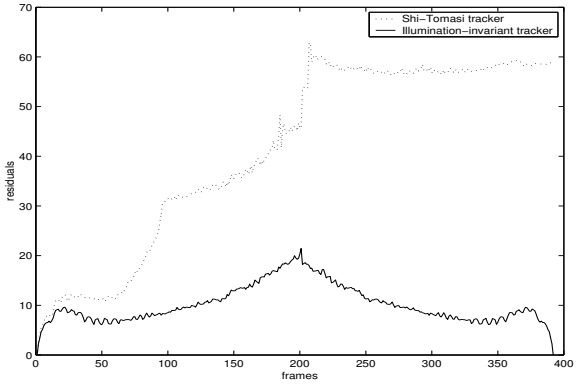


**Figure 4.** *Evolution of SSD residuals. After* 196 *steps Shi-Tomasi tracker cannot compensate for changes in illumination. From frame 197 to frame* 392 *the box comes back to its original position and the patch can be matched exactly (i.e. the residual goes to zero) as expected.*

using displacement and velocity constraints. In *International Conference on Computer Vision*, 1995.

[9] C. Tomasi and J. Shi. Good features to track. In *IEEE Computer Vision and Pattern Recognition*, 1994.

[10] T. Tommasini, A. Fusiello, E. Trucco, and V. Roberto. Making good features to track better. In *IEEE Computer Vision and Pattern Recognition*, 1998.

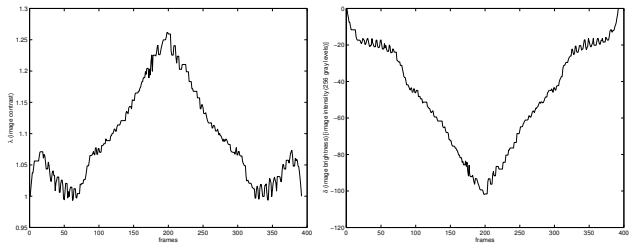

**Figure 5. Left***: Evolution of* $\lambda$ *(image contrast). The image contrast increases until frame* 197 *and then goes back to* 1 *at frame* 392*, where the box returned to the original position.* **Right***: Evolution of* $\delta$ *(image brightness). The image brightness decreases until frame* 197 *and then increases going back to* 0 *where the patch returned to the original position.*

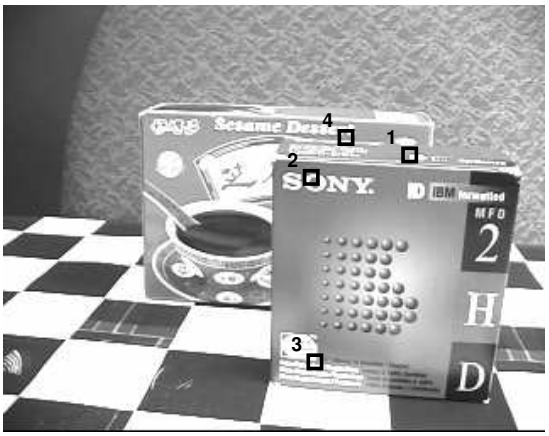


**Figure 6.** *One snapshot from the* "two boxes" *sequence. In this sequence some features have been chosen using the selection algorithm of Shi-Tomasi. Feature 4 is chosen to show the behavior of the residual when there is partial occlusion. Feature 3 is an example of point feature that has high residual (when light changes are not accounted for) comparable to occluded features residual (see feature 4).*

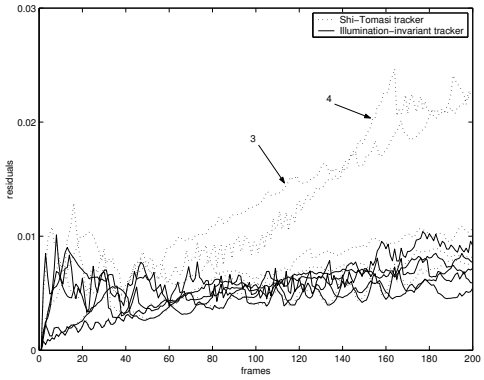


**Figure 7.** *Evolution of SSD residual for the* two boxes *sequence. The residuals of features 3 and 4 are comparable. Thus, usual monitoring for occluded features becomes inadequate when the environmental light changes. The other features have residuals that are comparable in both Shi-Tomasi and illumination-invariant trackers. The other features do not suffer from strong light changes. Feature 3 is subject to the reflections from the checker board, while feature 4 becomes partially occluded.*