Illumination-Invariant Tracking via Graph Cuts

Daniel Freedman and Matthew W. Turek
Computer Science Department, Rensselaer Polytechnic Institute, Troy, NY 12180

Abstract

Illumination changes are a ubiquitous problem in computer vision. They present a challenge in many applications, including tracking: for example, an object may move in and out of a shadow. We present a new tracking algorithm which is insensitive to illumination changes, while at the same time using all of the available photometric information. The algorithm is based on computing an illumination-invariant optical flow field; the computation is made robust by using a graph cuts formulation. Experimentally, the new technique is shown to quite reliable in both synthetic and real sequences, dealing with a variety of illumination changes that cause problems for density based trackers.

Keywords: tracking, illumination invariance, optical flow, graph cuts.

1 Introduction

Illumination changes are a ubiquitous phenomenon in many computer vision applications. Figure 5 shows two nearby frames in a video sequence: as the individual walks out of the shadow, his appearance changes quite drastically. The ability to cope with illumination changes is therefore an important aspect of many vision systems. In this paper, we will examine the problem of illumination invariant tracking.

A traditional method for dealing with illumination changes in tracking algorithm has been to use illumination invariant features, such as edges. In principle, the entire set of contour-tracking algorithms are invariant to illumination. (In practice, of course, edge-detection is somewhat dependent on illumination conditions.) However, the computer vision community has recently witnessed the development of several excellent tracking methodologies that are based primarily on tracking photometric, i.e. illumination-dependent, variables (such as intensity, colour, or texture). In particular, the performance of the mean-shift tracker [9, 10] has been shown to be outstanding in many important applications, outperforming many of the contour-based approaches. Unfortunately, because of the reliance of such methods on photometric variables, they are inherently sensitive to illumination changes. A desirable objective, therefore, is to design a tracker which retains one of the main benefits of the mean-shift tracker, namely its use of the available photometric information, while at the same time achieving a robustness to illumination changes.

In this paper, we use an approach to tracking based on optical flow. In principle, optical flow should be a natural starting point for tracking algorithms; an accurate computation of the flow field contains nearly all of the motion information that can be gleaned from a pair of consecutive images, and is therefore an ideal primitive for tracking. In general, however, most recent tracking algorithms have tended to eschew optical flow (with a few notable exceptions, e.g. [12]); the reason is that the optical flow computation is *not* believed to be completely accurate or reliable. To make matters worse, optical flow seems to depend inherently on stable illumination; after all, we cannot derive equations based on the assumption of “brightness constancy” if the brightness is not, in fact, constant. Thus, using optical flow for tracking faces the two problems of unreliability of the flow computation, and dependence on constant illumination. Our challenge is to design an algorithm for flow computation which is robust, and which is illumination invariant.

The major theoretical contribution of this paper is an algorithm for “illumination-invariant optical flow.” The optical flow field is found by optimizing an energy function based on a graph cuts formulation. The flow computation can be made robust by using a non-convex smoothness term; this fact is well known from prior graph cut based methods for optical flow, such as [4]. However, such approaches (like all approaches to optical flow) depend on brightness constancy, and hence are illumination dependent. Our technique, by contrast, assumes that illumination can be transformed between frames. This transformation is not explicitly computed; instead, it is approximated by looking at corresponding pairs of pixels in both images. Specifically, if two pixels in the first image have similar intensities, we expect that (in most cases) the corresponding pair in the second image will have similar intensities; and the same hold true for dissimilar pairs. This simple notion allows one to compute flow fields even when the illumination changes significantly.

The outline of the remainder of the paper is as follows. In Section 2, we describe related work. Section 3 sets out the relevant energy function, and describes how the optical flow computation can be used for tracking. Section 4 shows results on synthetic and real video sequences, and compares these results with mean-shift tracking. Section 5 concludes.

2 Related Work

There are three areas of computer vision which bear on the work presented in this paper: graph cuts, optical flow, and tracking. We briefly review the most relevant literature in each case.

The use of graph cuts to solve energy minimization problems with label sets of size greater than two was introduced in [5, 4], in which the concepts of α-expansions and α-β swaps were introduced. A deeper treatment of the problem of which functions can be minimized via graph cut techniques was given in [20], which introduced the concept of regularity. "Visual correspondence" problems, including optical flow, were treated in [4, 19, 18]; in particular, [18] bears a relation to the work in this paper. [25] uses graph cut techniques for extracting motion layers. Graph cut techniques have been used for other purposes as well, including multi-camera stereo, clustering, and segmentation.

The optical flow literature is vast, so no attempt will be made to summarize it here; for a comprehensive review of the literature up to 1998, please see [1, 14], and references therein. The main findings of these studies are that the best performing optical flow techniques are the algorithms of Lucas and Kanade [21] and Fleet and Jepson [13]; algorithms which rely on a global smoothness term do not perform as well. Of course, the graph cuts formulations are not reviewed in these papers, as this literature emerged after 1999; the non-convex smoothness terms which are allowed for graph cuts formulations (see above references) make for much more effective global smoothness terms. Other relevant papers in optical flow include the work of Irani on low-rank constraints [16], and the subsequent nonlinear extensions [24, 23]. All optical flow approaches mentioned are based on the notion of brightness constancy.

The tracking literature is very large, so we focus on two aspects of it. First, there is the family of mean-shift trackers [9, 8, 10], arguably the most popular trackers available today. These algorithms, which operate through a density-matching operation, get excellent results for a variety of sequences, and do so in real-time. Second, we mention the class of illumination-invariant trackers. Linear subspace illumination models were used in [15] to actually model the illumination changes. Illumination was also explicitly modeled in [6] for the purposes of head-tracking. A transformed colour space was used in [11], which is theoretically invariant to a number of different illumination changes, while [22] employs an MRF approach. There has also been some work in the area of shadow removal for tracking, such as [17]. Finally, the field of contour tracking relies on edges, which are illumination invariants; examples of popular contour trackers include [2, 7].

3 Theoretical Contributions

In this section, we introduce the main theoretical contribution of this work: a method for computing illumination invariant optical flow. This notion itself is somewhat counter-intuitive, as optical flow typically relies heavily on constancy of illumination (i.e. "brightness constancy"). However, as we have already argued, the reliable computation of a flow-field which is robust even in the presence of illumination changes, is a linchpin of our approach to tracking. The reason the approach is promising is because optical flow makes use of all of the available photometric information. Contrast this, for example, with contour trackers: while these trackers are in principle insensitive to illumination changes, they throw away much of the valuable photometric information in the scene, and thereby lose robustness.

We will begin by reviewing traditional algorithms which compute ordinary (illumination dependent) optical flow based on a graph cuts formulation. We then go on to discuss the main idea of the section: the new energy function which captures the notion of illumination invariant optical flow. Subsequently, we review the algorithms for optimizing the new energy function, i.e. the graph cut techniques themselves. Finally, we discuss the important issue of how to use the optical flow computation in order to track.

3.1 Traditional Optical Flow via Graph Cuts

In this section, we describe the traditional method of casting optical flow in a graph cuts formulation, which was first introduced in [5, 4]. Optical flow is an example of a "visual correspondence" problem, a problem of relating pixels in one image to those in another image. Similar problems crop up in stereo, though the energy functions may be different.

Let us denote the images at time t and $t+1$ by I_t and I_{t+1}, respectively. A pixel is given by p, the set of pixels is $\mathcal{P}$, and the set of pairs of neighbouring pixels is $\mathcal{N} \subset \mathcal{P} \times \mathcal{P}$. Let the flow vector for the pixel p in image I_t be given by δ_p; that is, the pixel p in image I_t flows to the position $p + \delta_p$ in image I_{t+1}. In this case, we may formulate the energy function as

$$E(\{\delta_p\}_{p\in\mathcal{P}}) = \lambda \sum_{p\in P} \psi_{\sigma_1}(|I_{t+1}(p+\delta_p) - I_t(p)|) + (1-\lambda) \sum_{(p,q)\in\mathcal{N}} \psi_{\sigma_2}(\|\delta_p - \delta_q\|)$$

where ψ_σ is a robustifying function, designed to deal with outliers, such as:

$$\psi_\sigma(z) = \begin{cases} z & z < \sigma \\ \sigma & \text{otherwise}. \end{cases}$$

(Note that if $\sigma = \infty$, then ψ_σ is no longer robustifying; while if the δ_p's are quantized, then choosing σ sufficiently small makes ψ_σ into a Potts function.)

The first term ensures brightness constancy: a pixel in the image I_t flows to a pixel in I_{t+1} with close to the same intensity. This is the key idea behind optical flow. The second term is a regularizing term: this ensures that the flow-field is relatively smooth. In the experimental papers [1, 14], it was noted that optical flow algorithms with global smoothness terms generally did not perform as well as those without, such as Lucas-Kanade [21] and Fleet-Jepson [13]. However, unlike the smoothness terms examined in these papers, the current smoothness term is a *non-convex* term, which does not excessively penalize pairs of neighbouring vectors which are dissimilar. This effectively allows for discontinuities in the flow-field that would be oversmoothed if a traditional quadratic smoothness term were used.

We postpone a discussion of how to optimize such a function until Section 3.3. For the moment, we simply note that good experimental results were attained on a few pairs of images in [4].

3.2 Illumination Invariant Optical Flow

Our goal is to find an energy function which computes the optical flow, but does so independent of whether the illumination has changed. Obviously, one can no longer use an approach based on brightness constancy; a pixel will not necessarily map to a pixel with similar intensity, as that pixel may be illuminated differently in the second image. Imagine, for example, an individual walking into a shadow cast by a building; a formerly light-coloured pixel on the individual's shirt may now appear considerably darker.

Suppose, to begin with, that there is a global illumination change across the scene. Let the range of the photometric variable of interest (grayscale intensity, colour, or even texture) be given by $\mathcal{I}$. Then the illumination change is described by a function

$$f : \mathcal{I} \to \mathcal{I}$$

Combining this with the notion of optical flow gives the following relationship, which replaces brightness constancy:

$$I_{t+1}(p + \delta_p) = f(I_t(p)) \tag{1}$$

Of course, we don't have access to the function f; however, knowing that there is a functional relationship allows us to recast equation (1) as

$$I_t(p) \text{ close to } I_t(q) \Rightarrow I_{t+1}(p + \delta_p) \text{ close to } I_{t+1}(q + \delta_q)$$
$$I_t(p) \text{ far from } I_t(q) \Rightarrow I_{t+1}(p + \delta_p) \text{ far from } I_{t+1}(q + \delta_q)$$

We have not been very explicit about the meanings of "close to" or "far from;" these qualitative ideas will be replaced by quantitative notions shortly, in formulating the energy function. Note that in recasting in this fashion, we have assumed something about f. While the "close to" condition is fairly generic, following directly from the idea of continuity of f, the "far from" condition could be violated easily if f is many-to-one. However, we have found that in practice the "far from" condition is not only satisfied for most illumination changes, but its utilization is critical to the success of the algorithm.

Before writing out the problem formally, let us first discuss the case of a non-global illumination change. This is likely to occur in many practical settings: as an object moves through a shadow, the part of the object which is in the shadow has a different function f than the part of the object which is still out of the shadow. Thus, we expect that the above qualitative relations should be replaced by

$$\text{if } p \text{ is near } q \text{ then}$$
$$I_t(p) \text{ close to } I_t(q) \Rightarrow I_{t+1}(p + \delta_p) \text{ close to } I_{t+1}(q + \delta_q)$$

and similarly for the second case, in which $I_t(p)$ is far from $I_t(q)$. In other words, the f relation is only defined locally. Of course, the above relationship will not hold where there is sharp changes in illumination, such as the case where p is on one side of a shadow and q is on the other side of the shadow. This difficulty will be taken care of by the energy function formulation.

Using these ideas, we may formulate the energy function. To begin with, let us quantize both images I_t and I_{t+1}; we denote the quantized versions as $\hat{I}_t$ and $\hat{I}_{t+1}$, respectively. This allows us to make more precise the notions of "close to" and "far from" described in the previous paragraph: $I_t(p)$ is close to $I_t(q)$ if $\hat{I}_t(p) = \hat{I}_t(q)$, and $I_t(p)$ is far from $I_t(q)$ if $\hat{I}_t(p) \neq \hat{I}_t(q)$. Thus, there is a fairly sharp split between pixels with similar and dissimilar intensities; however, we will find that this causes no problems in practice. It is important to note that the two images I_t and I_{t+1} may very well be quantized differently; indeed, this is the likely outcome if there is a sharp illumination change between the frames. For example, suppose that the sun recedes behind a cloud, leading to an image I_{t+1} with a smaller dynamic range than its predecessor I_t; this I_{t+1} will be quantized more finely than I_t.

Now, introduce the following notation:

$$\Theta[x_1, x_2] = \begin{cases} 1 & \text{if } x_1 = x_2, \\ -1 & \text{otherwise.} \end{cases}$$

Then the energy function can be written

$$E = (1 - \lambda) \sum_{(p,q) \in \mathcal{N}} \psi_\sigma(\|\delta_p - \delta_q\|) + \tag{2}$$
$$\lambda \sum_{(p,q) \in \mathcal{M}} \{1 - \Theta[\hat{I}_t(p), \hat{I}_t(q)]\, \Theta[\hat{I}_{t+1}(p + \delta_p), \hat{I}_{t+1}(q + \delta_q)]\}$$

To understand the energy function, let us examine the data term more carefully. The term

$$1 - \Theta[\hat{I}_t(p), \hat{I}_t(q)]\, \Theta[\hat{I}_{t+1}(p + \delta_p), \hat{I}_{t+1}(q + \delta_q)]$$

is 0 when either

- $\hat{I}_t(p) = \hat{I}_t(q)$ and $\hat{I}_{t+1}(p + \delta_p) = \hat{I}_{t+1}(q + \delta_q)$ or
- $\hat{I}_t(p) \neq \hat{I}_t(q)$ and $\hat{I}_{t+1}(p + \delta_p) \neq \hat{I}_{t+1}(q + \delta_q)$

In other words, we incur no penalty when the relationship ("close to" or "far from") between the pixels p and q in the image I_t is the same as the relationship between the pixels they map to, i.e. $p+\delta_p$ and $q+\delta_q$, in the image I_{t+1}. On the other hand, if the relationship between pairs of pixels in the two images is opposite, then we incur a penalty of 2. As a result, in minimizing the energy, we will seek to ensure that pairs of pixels in the first image are related in the same way as the pairs they map to in the second image, in as many cases as possible.

Note that the data term is summed over $\mathcal{M}$, which is a neighbourhood relation on the set of pixels $\mathcal{P}$, i.e. $\mathcal{M} \subset \mathcal{P} \times \mathcal{P}$. However, it may be the case that $\mathcal{M}$ is different from the neighbourhood relation $\mathcal{N}$. In particular, $\mathcal{N}$ will usually contain only the nearest neighbours (an 8-neighbourhood in our case), as this is sufficient to ensure that the flow-field is smooth. By contrast, $\mathcal{M}$ encapsulates the extent to which the illumination change is global; if $\mathcal{M} = \mathcal{N}$, then the illumination change is assumed to be completely local, whereas if $\mathcal{M} = \mathcal{P} \times \mathcal{P}$, then the change is completely global. Any intermediate $\mathcal{M}$ is possible.

Unfortunately, the energy function in (2) cannot be minimized by graph cut algorithms. The reason is the the data term, which usually depends only on single variable terms, now depends on pairs of variables. In other words, whereas the data term can normally be written $D_p(\delta_p)$, it now has the form $D_{p,q}(\delta_p, \delta_q)$. As a result, the regularity conditions [20] which the energy function must satisfy in order to apply the expansion or swap algorithms, must apply not only to the smoothness term (which in this case does satisfy the regularity condition), but also to the data term. It is easy to show that the data term will not, in general, satisfy this condition.

As a result, we emend the function in (2) slightly:

$$E = (1 - \lambda) \sum_{(p,q) \in \mathcal{N}} \psi_\sigma(\|\delta_p - \delta_q\|) + \lambda \sum_{(p,q) \in \mathcal{M}} \{1 - \Theta[\hat{I}_t(p), \hat{I}_t(q)]\, \Theta[\hat{I}_{t+1}(p + \delta_p), \hat{I}_{t+1}(q + \delta_p)]\} \tag{3}$$

where the only change we have made is to replace $q + \delta_q$ by $q + \delta_p$. This approximation pretends that the pixel q is mapped to $q + \delta_p$, rather than $q + \delta_q$; if the neighbourhood $\mathcal{M}$ is local (e.g. $\mathcal{M} = \mathcal{N}$) and the flow is smooth, then this will be a very good approximation in most places. At flow discontinuities, it will not be a good approximation, but due to the robust nature of both the data and smoothness terms (both ignore outliers), this should not substantially affect the flow computation. From the mathematical point of view, the data term now only depends on a single term (i.e. δ_p); as a result, expansion or swap moves on the energy function now satisfy the regularity conditions of [20], and we can therefore apply these graph cut methods.

It is worth noting that occlusions can be incorporated into the energy function in (3) using methods similar to those explored in [4] (i.e. using the occlusion as an extra label). None of the results of Section 4 use this term, although one example tracks through a small occlusion; however, we have found that inclusion of this term does lead to successful tracking through reasonable sized occlusions in practice. We do not say anything more about occlusions at this time.

3.3 Optimization Techniques

In order to optimize the energy function described in (3), we may use methods based on graph cuts. As we have mentioned, the function satisfies the conditions necessary for these methods to work: the pairwise (i.e. smoothness) term is a metric in δ_p, δ_q. As a result, we can apply either of the usual techniques, α-expansions or α-β swaps. In all of the experiments, we have used the α-expansion methodology; therefore, we briefly describe this technique here. For greater details, the reader is referred to [4].

The idea behind the graph cuts paradigm is to convert an energy function with a label set of arbitrary size to an energy function with a binary label set, by allowing a certain large "move" of the labels. When this is achieved, a multiway cut problem become an ordinary cut problem, which can be solved in polynomial time via max-flow algorithms. In particular, the conversion which is done for the case of α-expansions is to allow moves of the following kind:

$$\forall p, \quad \delta'_p = \delta_p \text{ or } \alpha$$

for a fixed α. The binary label set which is induced is therefore "remain the same" ($= 0$), or "change to α" ($= 1$). The graph cut algorithm optimally determines which pixels p should change their values to α, and which should retain their old values, in polynomial time. In particular, we use the algorithm described in [3], which is very fast in practice.

3.4 From Optical Flow to Tracking

We would like to use the computed optical flow in order to track. There are a variety of ways of doing this, including those mentioned in [12]. Here we make use of the simplest possible method. Namely, we characterize the object as a collection of pixels $\mathcal{O}$. In order to propagate this collection of pixels forward in time, we simply propagate each of them individually according to their optical flows:

$$\mathcal{O}_{t+1} = \{p + \delta_p : p \in \mathcal{O}_t\}$$

Because of the simplicity of this idea, it has several drawbacks. The main disadvantage is the fact that the object itself is not characterized in the normal fashion, as a compact connected subset of the plane. As a result, we are not guaranteed that it be connected; in principle, it can break up into several small components within a few frames. However, given a reasonably strong smoothness constraint, this is not observed in practice. Indeed, in all of the examples we show in Section 4, the object remains a connected component of the plane. In future work, we plan to use explicit constraints to ensure that this constraint is met.

4 Results

In this section, we demonstrate the efficacy of our algorithm in dealing with illumination changes. Several sequences are used, ranging from a completely synthetic scene to real scenes taken with a handheld DVI video camera. The video sequences were originally captured in color, but for simplicity we have converted them to grayscale, typically by choosing an individual color channel. We compare the results of the optical flow based technique with that of the mean shift algorithm.

The following parameter values were used for the illumination invariant tracker. λ, the smoothness-data tradeoff, was set in the range $0.1 - 0.25$. σ, the scale of the robust function used in the smoothness term, was set to 1.5. The two neighbourhood relations, $\mathcal{M}$ and $\mathcal{N}$, were both set to be 8-neighbourhoods. The quantization of any image was performed by dividing the dynamic range into 8 equally-spaced levels. The label set included flow vectors of up to 6 pixels in both vertical and horizontal directions. The result of using this large label-set is that the algorithm is not real-time, taking about a minute per frame; while this is problematic, we have in mind several strategies to drastically reduce the time of computation (see Section 5).

Two versions of the mean shift tracker were used. The first version obtains the target distribution from a single hand picked region. This distribution is static throughout the sequence. The second version uses the initial target distribution described above, but then updates the target distribution from the tracking region after the algorithm has converged on a frame. For clarity we denote this second version *adaptive mean shift tracking*. In the sections below, we present the results from the version of mean shift which worked best in each sequence. The scenes we present do not show large changes in object size, so the mean shift algorithm was run with a fixed scale.

4.1 Results on Synthetic Sequences

A completely synthetic sequence was generated; results are shown in Figure 1. The textured square is Gaussian random noise with mean 0 and standard deviation 20. The background of the image is zero (constant). Objects (including the background) on the right hand side of an image have had their intensity increased by 50 to simulate a sharp illumination boundary. Once the textured square crosses fully into the right hand side of the image its distribution is mean 50 and standard deviation 20. Figure 1 contains the results of a mean shift tracker (top row) and the illumination invariant tracker (bottom row) on the synthetic scene. Note that the mean shift tracker locks onto the illumination boundary. The square is well tracked by the illumination invariant tracker, however, and suffers minimal disturbance around the illumination boundary.

4.2 Results on Semi-Synthetic Sequences

Two semi-synthetic scenes were created by taking real video from a handheld camera and corrupting them with an additive sinusoidal intensity pattern. In the first sequence, the sinusoid was stationary with amplitude 50 and period 32 pixels. In the second scene a similar sinusoid translates left at 6 pixels per frame.

Figure 2 contains the results of the adaptive mean shift tracker (top row) and the illumination invariant tracker (bottom row) run on the stationary sinusoid sequence. The adaptive mean shift tracker begins to wander about 17 frames into the sequence and it locks onto the background a few frames later. The illumination invariant technique tracks the person throughout the sequence. A small amount of wandering occurs around frame 30, but the tracker has recovered well by frame 40.

Figure 3 shows the results of the adaptive mean shift tracker (top row) and the illumination invariant technique (bottom row) on the translating sinusoid sequence. The adaptive mean shift tracker loses the person early (less than 5 frames) into the sequence and instead locks onto the moving sinusoid. The illumination invariant tracker maintains good localization of the person throughout the tracking sequence.

4.3 Results on Real Sequences

Finally, we present results on two real sequences. The first sequence contains a person walking through two strong shadows. The second sequence shows a car driving behind a lamppost; this sequence contains minor illumination changes, which may be difficult to see from the images.

Figure 4 and Figure 5 contain the results of the adaptive mean shift tracker and the illumination invariant tracker, respectively, on the shadow sequence. The adaptive mean shift tracker begins to lock onto the darker background (and fails to recover) as the subject moves into an area of strong lighting. The illumination invariant tracker successfully follows the object of interest out of and into strong shadows.

Figure 6 contains the results of the adaptive mean shift tracker (top row) and the illumination invariant tracker (bottom row) on the car sequence. The illumination invariant tracker shows good resilience in the face of the partially occluding lamppost.

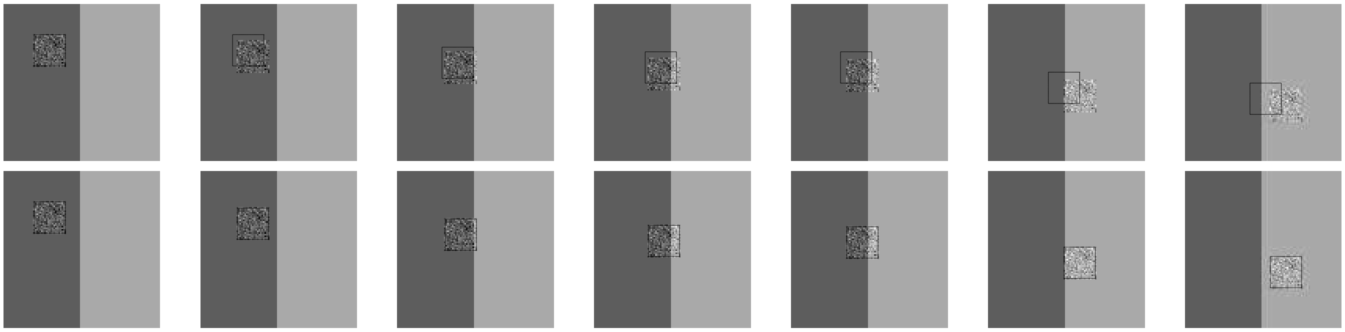

Figure 1. Frames 1, 5, 12, 16, 17, 30, and 36 from synthetic sequence. The textured square is Gaussian random noise with mean 0 and standard deviation 20. The right hand side has an intensity increase of 50. Top row: Results of using a mean shift tracker. Bottom row: Results of using the illumination invariant tracker.

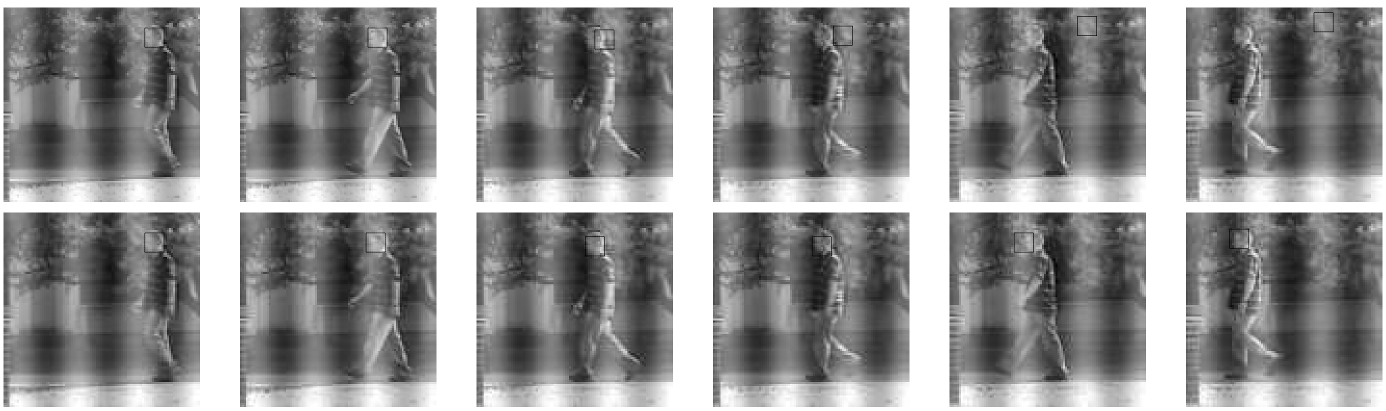

Figure 2. Frames 1, 10, 17, 20, 30, and 40 from the semi-synthetic sequence. The corrupting sinusoid is additive, with amplitude 50. Top row: Around frame 17, the adaptive mean shift tracker starts to wander. Bottom row: The illumination invariant technique tracks the person well for the whole sequence.

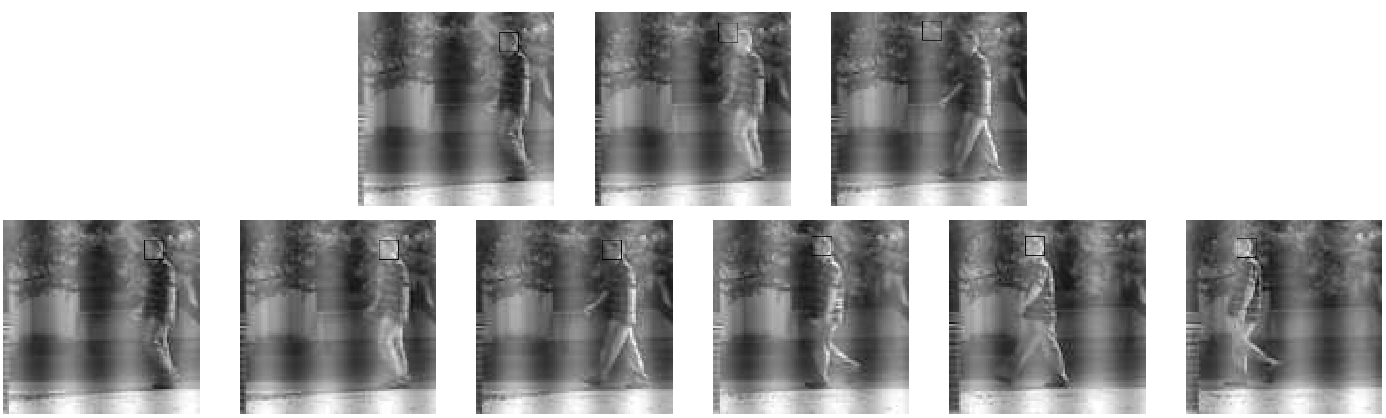

Figure 3. Top row: Frames 1, 5, 10. The adaptive mean shift tracker has almost completely lost the object by frame 5 and by frame 10 it has locked onto the moving sinusoidal pattern. Bottom row: Frames 1, 5, 10, 20, 30, and 40. The illumination invariant tracker maintains a secure focus on the target object throughout the sequence.

Figure 4. Results of using the adaptive mean shift tracker for a sequence with strong shadows. The figure shows frames 1, 10, 20, 30, 50, and 70. The tracker begins to lock onto the darker background (and fails to recover) as the subject moves into an area of strong lighting.

Figure 5. Results of using the illumination invariant tracker for a sequence with strong shadows. The figure shows frames 1, 10, 20, 30, 50, 70, 100, 150, and 210. The illumination invariant tracker successfully follows the object of interest out of and into strong shadows.

5 Conclusions and Directions for Future Research

The main contribution of this work has been the development of an optical flow algorithm which is invariant to illumination changes. This algorithm, which is posed as the optimization of an energy function, is solved via graph cut techniques. A straightforward pixel propagation method is used to convert the optical flow field into a tracking algorithm. The value of the algorithm has been demonstrated on a number of scenes with complex illumination changes, scenes which are challenging for state-of-the-art trackers.

There are two main directions for future research. First, we shall investigate a more sophisticated use of optical flow in tracking. We view the optical flow algorithm developed here as a low-level primitive; that is, the flow field can be used as the basis for a number of higher level algorithms, including tracking. In particular, when moving from flow to tracking, we want to ensure that the object's geometry remains reasonable; we also may wish to incorporate some simple dynamical assumptions. Both objectives may be achieved by a more sophisticated treatment of the flow field, once it has been computed. Our second direction for research is to find faster algorithms for computing the flow. Currently, the method based on alpha expansions can be costly, given the size of the label set. We are interested in investigating a coarse-to-fine approach, in which the coarseness is based not only on the true scale of the image, but also on the size of the label set. Such a method will lead to an algorithm which may be considerably faster.

Acknowledgments

This research was supported by the U.S. Army Intelligence and Security Command under contract W911W4-F-04-0131.

Figure 6. Frames 1, 2, 5, 10, 20, and 30 of a sequence with an occluder. Top row: The adaptive mean shift tracker wanders significantly. Bottom row: The illumination invariant tracker formulation maintains a lock on the car while it passes behind the occluding lamppost.

References

[1] J. Barron, D. Fleet, and S. Beauchemin. Performance of optical flow techniques. *Int. J. Comput. Vis.*, 12(1):43–77, 1994.

[2] A. Blake and M. Isard. Condensation - conditional density propagation for visual tracking. *Int. J. of Comput. Vis.*, 29(1):5–28, 1998.

[3] Y. Boykov and V. Kolmogorov. An experimental comparison of min-cut/max- flow algorithms for energy minimization in vision. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(9):1124–1137, 2004.

[4] Y. Boykov, O. Veksler, and R. Zabih. Fast approximate energy minimization via graph cuts. *IEEE Trans. Pattern Anal. Machine Intell.*, 23(11):1222–1239, 2001.

[5] Y. Boykov, O. Veksler, and R. Zabih. Fast approximate energy minimization via graph cuts. In *Proc. ICCV*, volume 1, pages 377–384, 2001.

[6] M. L. Cascia and S. Sclaroff. Fast, reliable head tracking under varying illumination. In *Proc. CVPR*, volume 1, pages 604–610, 1999.

[7] V. Caselles, R. Kimmel, and G. Sapiro. On geodesic active contours. *Int. J. Comput. Vis.*, 22(1):61–79, 1997.

[8] D. Comaniciu. An algorithm for data-driven bandwidth selection. *IEEE Trans. Pattern Anal. Machine Intell.*, 25(2):281–288, 2003.

[9] D. Comaniciu, V. Ramesh, and P. Meer. Real-time tracking of non-rigid objects using mean shift. In *Proc. Int. Conf. Comput. Vis. Pattern Recog.*, volume 2, pages 142–149, 2000.

[10] D. Comaniciu, V. Ramesh, and P. Meer. Kernel-based object tracking. *IEEE Trans. Pattern Anal. Machine Intell.*, 25(5):564–577, 2003.

[11] G. D. Cubber, H. Sahli, H. Ping, and E. Colon. A colour constancy approach for illumination invariant colour target tracking. In *Proc. IARP Workshop on Robots for Humanitarian Demining*, 2002.

[12] D. DeCarlo and D. Metaxas. Optical flow constraints on deformable models with applications to face tracking. *IJCV*, 38(2):99–127, 2000.

[13] D. Fleet and A. Jepson. Computation of component image velocity from local phase information. *Int. J. Comput. Vis.*, 5(1):77–104, 1990.

[14] B. Galvin, B. McCane, K. Novins, D. Mason, and S. Mills. Recovering motion fields: an evaluation of eight optical flow algorithms. In *Proc. BMVC*, 1998.

[15] G. Hager and P. Belhumeur. Efficient region tracking with parametric models of geometry and illumination. *IEEE Trans. Pattern Anal. Machine Intell.*, 20(10), 1998.

[16] M. Irani. Multi-frame optical flow estimation using subspace constraints. In *Proc. ICCV*, 1999.

[17] H. Jiang and M. Drew. Tracking objects with shadows. In *Proc. SPIE Image and Video Communications and Processing*, 2003.

[18] J. Kim, V. Kolmogorov, and R. Zabih. Visual correspondence using energy minimization and mutual information. In *Proc. ICCV*, 2003.

[19] V. Kolmogorov and R. Zabih. Computing visual correspondence with occlusions using graph cuts. In *Proc. ICCV*, 2001.

[20] V. Kolmogorov and R. Zabih. What energy functions can be minimized via graph cuts. In *Proc. ECCV*, 2002.

[21] B. Lucas and T. Kanade. An iterative image registration technique with an application to stereo vision. In *Proc. DARPA IU Workshop*, pages 121–130, 1981.

[22] M. S.Kamijo. ”illumination invariant and occlusion robust vehicle tracking by spatio-temporal mrf model. In *Proc. 9th World Congress on ITS*, 2002.

[23] L. Torresani and C. Bregler. Space-time tracking. In *Proc. ECCV*, 2002.

[24] L. Torresani, D. Yang, G. Alexander, and C. Bregler. Tracking and modeling non-rigid objects with rank constraints. In *Proc. CVPR*, 2001.

[25] J. Xiao and M. Shah. Motion layer extraction in the presence of occlusion using graph cut. In *Proc. CVPR*, 2004.