A resource allocation framework for adaptive selection of point matching strategies*

Quentin De Neyer[1] and Christophe De Vleeschouwer[1]

Université catholique de Louvain, 1348 Louvain-La-Neuve, Belgique,
quentin.deneyer@uclouvain.be

Abstract. This report investigates how to track an object based on the matching of points between pairs of consecutive video frames. The approach is especially relevant to support object tracking in close-view video shots, as for example encountered in the context of the Pan-Tilt-Zoom (PTZ) camera autotracking problem. In contrast to many earlier related works, we consider that the matching metric of a point should be adapted to the signal observed in its spatial neighborhood, and introduce a cost-benefit framework to control this adaptation with respect to the global target displacement estimation objective. Hence, the proposed framework explicitly handles the trade-off between the point-level matching metric complexity, and the contribution brought by this metric to solve the target tracking problem. As a consequence, and in contrast with the common assumption that only specific points of interest should be investigated, our framework does not make any a priori assumption about the points that should be considered or ignored by the tracking process. Instead, it states that any point might help in the target displacement estimation, provided that the matching metric is well adapted. Measuring the contribution reliability of a point as the probability that it leads to a crorrect matching decision, we are able to define a global successful target matching criterion. It is then possible to minimize the probability of incorrect matching over the set of possible (point,metric) combinations and to find the optimal aggregation strategy. Our preliminary results demonstrate both the effectiveness and the efficiency of our approach.

Keywords: active tracking, point matching, cost-benefit optimization

1 Introduction

Tracking represents an important task within the field of computer vision. It has been studied intensively for a broad range of applications including surveillance, sport, traffic monitoring, augmented reality etc. Among these applications, active tracking [9, 12, 6], also named autotracking in the literature, offers challenges in both control and computer vision: a camera is commanded (pan, tilt, zoom)

* Part of this work has been funded by the european FP7 SV3D project, and by the Belgian NSF.

automatically to follow a target, based on its own captured video stream. It thus forms a closed-loop system where the error signal is measured by the tracking of the target across successive frames. Recent advances in the ad hoc technology have made it possible to command cameras accurately at high frequency, with small delays. In addition recent computers can now process video streams in realtime for heavy tasks such as tracking. As a result, it now becomes possible to support active tracking over large areas, collecting close-up sequences while keeping camera locked on the object of interest.

In this paper, we are primarily interested in tracking close-up views of a target across a video stream. Since only a fragment of the target might be visible (due to the close-up nature of the view, but also to potential occlusions), we model the target based on the texture locally surrounding a set of points. Hence, our problem consists in inferring a correct target displacement, by analysing the correspondance probability distributions computed for a given set of points in two successive frames. The points and their matching metric should be jointly selected to maximize the chance of correct target displacement inferrence. This is in contrast with many previous works that use a single matching strategy and restrict their investigation to points of interest, which are characterized by highly discriminant and/or highly invariant features [4, 7, 1]. In addition, when performing tracking in real-time, the computational resources to handle each pair of consecutive frames are limited. Therefore, we introduce a cost-benefit framework to maximize the chance to infer a correct target displacement, under some global computational complexity constraint. The deployment of this framework ends-up in selecting the points to match, and in adapting their respective matching metrics to their textured neighborhood. It implies a formal definition of the notion of point matching benefit, defined to reflect the contribution of each individual point to a reliable estimate of the target displacement. The presented matching framework can then be embedded in an active tracking process [2].

To conclude this introduction, it is worth pointing out that the proposed framework, unlike most popular matching methods, does not make any a priori assumption about the characteristics of the points that should be considered to support the tracking. Instead, it provides a methodology to estimate whether a given point neighborhood is worth being matched with some given metric. Any point is potentially relevant to infer the target displacement between two frames. Specifically, its relevance is estimated in terms of matching reliability, as a function of the spatial neighborhood and of the envisioned matching metric.

The rest of the report is organized as follows. Section 2 motivates the approach in comparison with state-of-art tracking methods. Next, Section 3 introduces the notion of matching metric, and explains how points should be selected, and how their distance maps aggregated, so as to maximize the chance of correct target displacement inference. Section 4 then defines the cost-constrained target displacement estimation problem, and proposes a heuristic method to select the points and their matching metrics in a so called cost-reliability (CR) optimal way. Eventually, Section 5 presents how the CR-optimal points matching process can be used in practice. Section 6 presents preliminary experimental results, whilst Section 7 concludes.

2 Related works

A very common way to track objects in videos captured with still cameras consists in modeling the background and comparing every new image to this model, in order to detect foreground activity [13]. When the camera moves, it is possible to reconstruct a sort of puzzle background, by compensating the camera motion between frames. The performance of this method is, however, severely degraded by the inaccuracies of the compensations (inaccuracy of detected motion and optical geometry compensation). Therefore, the authors in [10] use background subtraction methods that are robust to small spatial drifts. However, these methods really break down in case of zoom changes.

Because the tracked objects in our applications undergo severe global deformations, while remaining locally quite invariant, we focus our investigations on techinques using local descriptors and do not study trackers using global descriptors like shape, contours or histograms. Moreover, for computational efficiency reasons, we adopt a sparse representation of the object and do not consider motion analysis over the whole frame [3] or target [5].

Discrete local features have received a significant amount of attention in the computer vision research community over the past 15 years. Among the numerous developed algorithms, the most popular and intensively used methods are certainly the Harris corner detector[4], the SIFT keypoints detector/descriptor [7] and the similar speeded-up version SURF [1], but there exist plenty of them. These approaches work symetrically by detecting key features in two images and then perform discrete associations [11]. Although these methods have shown to be very performant in multiple viewpoints matching, panoramic recontruction, object recognition, and other matching tasks, their performances in object tracking (especially for deformable objects) are questionable. One practical reason is that keypoints are frequently located at the boundary of the object to track and therefore include background information in their description, with the risk that points drift into the background. The boundaries of a target also undergo non-affine transformation because of their 3D nature, making them uncertain source of information. More fundamentally, the use of a saliency criterion to select the points to match between frames supposes that one knows what point will be good for tracking a priori. In contrast, we relax this assumption, and consider that any point could be useful to infer information about target displacement, provided that it is matched with an appropriated metric. In the following, we use matching metric to denote the combination of a point neighborhood descriptor (e.g. a squared grid of pixels) and its associated distance metric (e.g. euclidian distance). By allowing the use of multiple strategies to match multiple points, we also increase the system's capability in handling specific problems. As a prime example, a gradient-based descriptor will be sensible to motion blur on an image, but not color histogram based descriptors, while these are led to mistake in presence of similarly coloured background.

3 Points selection and distance maps aggregation for reliable target displacement estimation

In this section, we investigate the procedure for aggregating information arising from multiple points, and derive an objective criterion for point and associated matching metric selection. Section 3.1 first defines the notion of matching metric and formalizes the point matching process, providing the material to understand the subsequent developments. Section 3.2 then considers how to best aggregate distance maps, in order to maximize the chance of correct target displacement inference, while Section 3.3 introduces the additive metric that is used to quantify the reliability of the target displacement estimate resulting from the aggregation of the distance maps computed for a set of target points. This metric is fundamental because it allows to order a given set of points according to their respective contribution to a reliable target displacement estimation. It will be the core of the cost-reliability optimal selection method presented in Section 4.

3.1 Matching metric and distance map

We consider the matching of a point p from a reference image I to a query image J. In the case of video tracking, I and J are two successive frames in the sequence. The matching metric includes (1) the function $rep(p, I)$ that computes a feature vector (descriptor) describing the point p and its neighborhood in image I, (2) the distance used to evaluate the similarity between two feature vectors and (3) the search region S in frame J. A patch of pixels luminance around p, a SAD-distance and a square search window define a simple example of matching metric.

Let $\mathcal{P}$ be a set of N points representing an object of interest. Considering that a matching metric is assigned (arbitrarily) to each point in P, N distance maps are computed by applying the matching metrics between the reference points in I and every point belonging to the corresponding search region in J. For every point in $\mathcal{P}$, the minimum of the distance corresponds to the most probable point displacement.

3.2 Aggregation of distance maps for reliable target registration

For a given set $\mathcal{P}$ of N points, this section considers aggregation of N distance maps $dm_i(x)$, $i = 1...N$, $x \in S$ = search region. For the sake of simplicity and without loss of generality, we restrict the developments to $S = \left\{X, \bar{X}\right\}$, which means that there are only two possible displacements X and $\bar{X}$ of the target between frame I and J, X being the actual target displacement. The N points of $\mathcal{P}$ belong to the same object, and are thus assumed to follow the same translation, up to some deformation noise.

The N observations, corresponding to the matching of N points, are considered to be independent. The most probable displacement $\hat{X}$ is given below:

$$\hat{X} = \underset{x \in \{X, \bar{X}\}}{\text{argmax}}\ P(d = x | dm_i(X), dm_i(\bar{X})_{|i=1\dots n})$$

By using the (conditional) independance of the observations and the Bayes formula, after some developments, we get (f denoting the pdf)

$$\begin{aligned}\hat{X} &= \underset{x \in \{X, \bar{X}\}}{\text{argmax}} \prod_{i=1}^{N} f(dm_i(x)|d = x) f(dm_i(\bar{x})|d = x) \\ &= \underset{x \in \{X, \bar{X}\}}{\text{argmax}} \prod_{i=1}^{N} \frac{f(dm_i(x)|d = x)}{f(dm_i(x)|d = \bar{x})} \underbrace{f(dm_i(\bar{x})|d = x) f(dm_i(x)|d = \bar{x})}_{equal\ for\ x=X\ and\ x=\bar{X}}\end{aligned}$$

We assume that $f(dm_i(x)|d = x)$ and $f(dm_i(x)|d = \bar{x})$ are normally distributed with parameters $N(\mu_{i1}, \sigma_{i1})$ and $N(\mu_{i0}, \sigma_{i0})$ respectively. For the sake of simplicity, we also consider that $\sigma_{i0} = \sigma_{i1} = \sigma_i$. This is not a strong assumption, as the standard deviation reflects the variability of the distance, as a result of the deformations of the tracked object. Hence, by taking the logarithm of the product and after simplifications, we get

$$\hat{X} = \underset{x \in \{X, \bar{X}\}}{\text{argmin}} \sum_{i=1}^{N} \frac{(dm_i(x) - \mu_{i1})^2}{\sigma_i^2} - \frac{(dm_i(x) - \mu_{i0})^2}{\sigma_i^2}$$

using relation $a^2 - b^2 = (a + b)(a - b)$ and developing, we have

$$\begin{aligned}\hat{X} &= \underset{x \in \{X, \bar{X}\}}{\text{argmin}} \sum_{i=1}^{N} \cancel{2} dm_i(x) \frac{(\mu_{i0} - \mu_{i1})}{\sigma_i^2} - \underbrace{\frac{(\mu_{i1} + \mu_{i0})(\mu_{i0} - \mu_{i1})}{\sigma_i^2}}_{constant\ term} \\ &= \underset{x \in \{X, \bar{X}\}}{\text{argmin}} \sum_{i=1}^{N} \frac{(\mu_{i0} - \mu_{i1})}{\sigma_i^2} dm_i(x) = \underset{x \in \{X, \bar{X}\}}{\text{argmin}} \underbrace{\sum_{i=1}^{N} \frac{\mu(D_i)}{\sigma^2(D_i)} dm_i(x)}_{g(x)}, \end{aligned} \tag{1}$$

where $D_i = dm_i(\bar{x}) - dm_i(x) \propto N(\mu_{i0} - \mu_{i1}, \sigma_{i0}^2 + \sigma_{i1}^2)$. The likeliest displacement, given N distance metrics two observations in two candidate displacements X and $\bar{X}$, is thus the one that minimizes $g(x)$, a weighted sum of the distance maps computed in the candidate positions.

3.3 Selection of points for reliable target displacement estimation

Now that we have found how to aggregate the maps (when they are just composed of two measurements for each point, see Section 3.2), we have to figure out how to select the points whose maps should be aggregated to best estimate the target displacement. The points selection problem is formulated as follows: given a set $\mathcal{P}$ of P points, find the subset of indices Φ_N such that $\{p_i, i \in \Phi_N\} \subset \mathcal{P}$ of points that minimizes the risk of wrong target displacement inference, given the aggregation rule deduced in Section 3.2. Let D_{Φ_N} be the difference of aggregated observations at the wrong and the right displacement,

$$D_{\Phi_N} = g(\bar{X}) - g(X) = \sum_{i=1}^{N} \frac{\mu(D_i)}{\sigma^2(D_i)} \underbrace{(dm_i(\bar{X}) - dm_i(X))}_{\propto N(\mu(D_i),\sigma^2(D_i))}. \tag{2}$$

Given the decision rule defined by Equation (1), the probability of having an error is equal to the probability that D_{Φ_N} is negative. It must be minimized. Because D_{Φ_N} is a weighted sum of gaussian variables, it also follows a gaussian distribution and the probability of error is the lowest when maximizing its mean versus standard deviation ratio, or the square of it:

$$\hat{\Phi}_N = \underset{\Phi_N}{\operatorname{argmax}} \frac{\mu(D_{\Phi_N})^2}{\sigma(D_{\Phi_N})^2} \quad = \quad \underset{\Phi_N}{\operatorname{argmax}} \sum_{i\in\Phi_N} \frac{\mu_i^2}{\sigma_i^2} \quad = \quad = \underset{\Phi_N}{\operatorname{argmax}} \sum_{i\in\Phi_N} r_i^2$$

Hence, the objective function to maximize when selecting the points to aggregate appears to be the squared sum of the ratio between the mean and the standard deviation of the variables D_i associated to the selected points. The ratio $r_i = \mu_i^2/\sigma_i^2$ denotes the i^{th} point matching reliability.

4 Cost-reliability optimal target registration

The developments of section 3 consider a simplified registration problem, observing the distances in two displacement candidates for a set of N points, with predefined matching metrics. In the proposed application, we are interested in estimating the most reliable displacement within a predefined search window S, for a given amount of computational resources. Therefore, in this section, we extend the set of candidates to an arbitrary search area, and consider that the matching metric of each point can be selected within a set of metrics, so as to achieve an optimal trade-off between displacement estimation reliability and computational cost. Section 4.1 formulates the cost-benefit optimization problem, while Section 4.2 proposes a heuristic to compute an approximated solution.

4.1 Cost-constrained target registration problem definition

By definition, we consider that a target is registered between two frames once its (translational) displacement between the two frames has been estimated. We showed that for two displacement assumptions X_1 and X_2, under some reasonable hypotheses, this is done by minimizing the weighted sum of observations $dm_i(X_1)$ and $dm_i(X_2)$ associated to a set of points $\Phi_N = \{p_i, i = 1...N\}$. In Section 3.3, we have explained that the probability of incorrect target registration can be minimized by maximizing the sum of the points matching reliabilities r_i. In practice, of course, the matching process generates more than 2 observations for each point. Those are spread over the search area S, resulting in a distance map (see Section 3.1). According to the results of section 3.3 and considering observation of the whole distance maps, minimizing the error risk is equivalent to minimizing the probability of error in any location $j \neq X$ (X denotes the correct displacement) of the search region. This is achieved by maximizing the sum of ratios $r_i(j) = \mu_i^2(j)/\sigma_i^2(j)$, as estimated in the most ambiguous candidate displacement $j \in S$. In other words, to minimize the error likelihood, the set of N points $\hat{\Phi}_N$ should be selected so as to maximize the smallest (over all positions j) sum of reliabilities, i.e.

$$\hat{\Phi}_N = \underset{\Phi_N}{\operatorname{argmax}} \min_j \sum_{i \in \Phi_N} r_i(j) \tag{3}$$

Estimating $r_i(j)$ for every possible $j \in S$ is, however, too expensive in practice. Moreover, even if the $r_i(j)$ and underlying $\mu_i(j)$ and $\sigma_i(j)$ had been estimated for all positions j surrounding the i^{th} point in the query image, they could not be used for summing the distance maps, since, in practice, the actual position of the i^{th} point in the target frame is not known. In order to simplify the problem and to make it solvable, we approximate the map $r_i(j)$ by a constant r_i multiplied by a binary mask $b_i(j)$. This means that for each point p_i, we consider that the r_i function is characterized by two regions in the search window: one region where $r_i(j)$ is small, i.e. where the distance map has low values, and another region where it has high values. The distance map dm_i corresponding to the point p_i primarily helps in rejecting the displacements j corresponding to high distance map values, i.e. to position j for which $b_i(j) = 1$. Fig. 1 illustrates this approximation on several typical instances of distance maps, showing that the hypothesis is reasonably justified in practice.

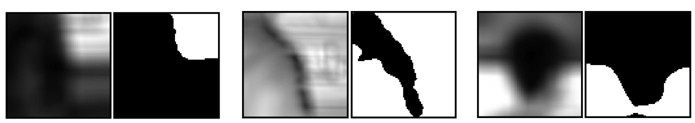

Fig. 1. Distance maps are often divided in two contrasted regions of likely and unlikely matching, so that $r_i(j)$ can be approximated by a constant r_i multiplying a binary mask

Two consequences arise from the above binary approximation. The first one is a simplification of the distance maps aggregation process. The distance maps $dm_i(j)$ can now be aggregated through a weighted sum, with a weight $r_i = \mu_i/\sigma_i^2$, assigned to each point p_i, that does not depend on j. The second consequence results from the fact that the area associated to large reliability values $r_i(j)$, for which $b_i(j) = 1$, is the one that helps the system to discriminate between the wrong displacement candidates and the correct one. To complement each others, points have thus to be selected so that they have complementary binary masks, i.e. so that they best disambiguate (reinforce the "wrong" status of) points that do not correspond to the target displacement. This notion of complementarity will be further discussed in Section 5.

Now that we have introduced our binary approximation, we come back to our target registration problem. For increased generality, instead of considering the selection of a set Φ_N of N points among P points to best estimate the target displacement, we consider that we have to assign, to each point p_i in $\mathcal{P}$, a matching metric $m(i) = m_k$, selected in the set $\mathcal{M}= \{m_k, k = 1...K\}$ of K admissible matching strategies. Interestingly, assuming that the point skipping decision defines a zero-complexity/zero-benefit matching metric option in $\mathcal{M}$, finding the optimal points matching strategies also implicitly provides a natural solution to select Φ_N, i.e. select the (number of) points that are worth to be matched, with respect to the global complexity constraint C. Hence, our problem writes as finding the optimal metric assignment m^*, that solves

$$\boldsymbol{m^*} = \underset{\boldsymbol{m}\in\mathcal{M}^N}{\operatorname{argmax}} \min_j \sum_{i=1}^{P} r_{ik} b_{ik}(j) \quad \text{subject to} \quad \sum_{i=1}^{P} c_{ik} < C, \tag{4}$$

where the matching reliability r_{ik}, the mask $b_{ik}(j)$, and the cost c_{ik} are associated to the pair (p_i, m_k), as defined in Section 3 and 4. This problem can be shown to be NP-hard, by reduction of the Knapsack problem [8].

4.2 Heuristic technique for approximate solution

A first simplification for solving (4) is to consider for each point only the best matching metric m_{k^*}, regarding the benefit/cost ratio. This metric satisfies $k^* = \operatorname{argmax}_k r_{ik}/c_{ik}$. The i^{th} point reliability is $r_i = r_{ik^*}$ and every point can be either matched with its optimal metric or not matched.

A natural way to solve problem (4) heuristically is to start from an empty set of points $\Phi = \{\}$ and add points one after one, selecting each point so as to disambiguate the matching at the most ambiguous location j on the distance map, assuming that the actual target displacement moves the i^{th} point to the center of the search range. Algorithm 1 presents the pseudo code of our solution. At each step, the most ambiguous location j^* is determined (line 3). Next the point that best disambiguate at location j^* is computed and added to Φ.

Algorithm 1 Heuristic point selection method

1: $\Phi = \{\}$; $N = 0$
2: **while** $\sum_{i\in\Phi_N} c_i < C$ and $N < P$ **do**
3: $\quad j^* = \operatorname{argmin}_j \sum_{i\in\Phi_N} r_i(j) b_i(J)$
4: $\quad i^* = \operatorname{argmax}_{i\notin\Phi_N} \frac{r_i}{c_i} b_i(j^*)$
5: $\quad \Phi = \{\Phi, p_{i^*}\}$; $N = N + 1$
6: **return** Φ

5 Towards practical implementation

This section aims at giving a better understanding of the theoretical results derived in Sections 3 and 4 through the discussion of practical implementation recommendations. Section 5.1 explains the implications of the choices of the initial points set, and motivates the partition of the search region for solving problem (4). It also proposes a method for estimating the parameters that are required to solve the problem. Section 5.2 gives an example of system implementation.

5.1 Principles to select the points set, and to estimate their associated distance map parameters

The only hypothesis to keep in mind when choosing the initial set $\mathcal{P}$ is that the observations should be mutually independant, conditionally to the target displacement (as requested in Section 3.2). This is achieved by selecting points that are not too close to each other. The set must be chosen, however, with $|\mathcal{P}| = P$ large enough to be certain to include enough points that are worth to be matched and can bring the variety/diversity of information required to solve (4), by maximizing the sum of reliabilities in every part of the search region.

When solving problem (4), using the heuristic approach described in Section 4.2 to select a point that disambiguate the matching in the most ambiguous location of the search region, it is quite cumbersome to consider each pixel j in S separately, while the distance maps and the associated reliability metrics $r_i(j)$ appear to be constant over large regions, and could be simplified by grouping pixels into subregions of S. Fig.2 shows three ways for partitioning the search region and their effect on the approximation of the pixel-precise mask $B(j)$ introduced in Section 4.1, and illustrated on Fig. 1. Hence, using this approximation, in the 3^{rd} line of algorithm 1 in Section 4.2, the most ambiguous *location* j^* becomes the most ambiguous *region* J^*. Note that these approximations only affect the choice of points to match and not the inference of the global target motion.

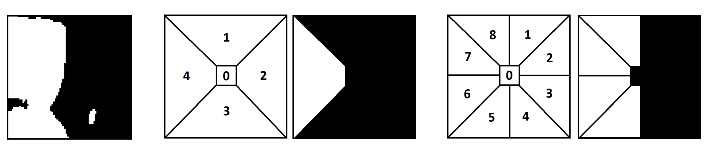

Fig. 2. The binary mask (left) approximation depends on the chosen partition

The solution presented in Section 4 to select the points and their associated matching metrics, but also to compute the weights for the optimal summation of the points distance maps require to know, for each possible association (p_i, m_k), the mean μ_{ik} and standard deviation σ_{ik} of the random variable D_{ik}, denoting the difference between the point matching distances computed in incorrect and correct positions. The optimal points subset selection also requires to estimate $b_{ik}(J)$. We propose to estimate those parameters based on the observation of the point neighborhood. This can be done, for instance, by computing the autocorrelation of point p_i in the reference image I, generating a distance map. This map can be thresholded to define the binary mask $b_{ik}(J)$, and the mean and variance parameters can then be computed over the sets of correct and incorrect matching distances.

5.2 Example of practical implementation

In this section, we show an example of implementation of the proposed framework. The method is then evaluated in Section 6. In order to fully exploit the possibilities of the approach, we consider a set $\mathcal{M} = \{m_1, m_2\}$ containing only two matching metrics, and a random set $\mathcal{P}$ with a large number of points. This contrasts with many solutions that seek for a small set of very strong points to match very precisely. The metric m_1 adopts a simple descriptor: a vector of pixel luminance values on a vertical line centered at the point location in the reference image. It performs a search in a square search region S.

Equivalently, m_2 considers a horizontal line as descriptor. Both matching metrics use the sum of absolute difference as distance for similarity evaluation. For the estimation of parameters μ_{ik} and σ_{ik}, k denoting the index of the matching metric, we compute the autocorrelation of point p_i on frame I divide the autocorrelation map according to the first partition represented on Fig. 2. All the μ_{ik}, σ_{ik} and $b_{ik}(J)$ are computed over these sets.

6 Experimental validation

This section aims at validating the proposed framework throuhg simple experiments on two successive frames of a video. The tests focus on frame registration and not on tracking, which involves too many other parameters. The experiment is set up as follows: given two successive video frames I and J, given the position of the target in I (mask), the method should infer the translational displacement of the target with a limited computational power (= complexity constraint) C. The method that is evaluated here is the one described in 5.2. A set of 12 pairs of frames, with different kind of manually tagged targets (people) is used as validation set. The displacement of the target between two frames is computed as the median displacement of several manually matched points pairs.

The performances are measured by comparison with a method that chooses a points set randomly and then assigns the vertical matching metric to half of the points and the horizontal one to the second half.

Fig. 3 (left) compares both methods in terms of averaged motion error (in pixels) for several complexity constraints $C = [2, 4, 6, 9, 12]$, the complexity unit being the matching of one point. The motion errors have been computed over the whole validation set and the tests were repeated to generate this averaged error measure. Because the search region is a square of side equal to 80 pixels, a mean error superior to 30 pixels is not better than a random guess of the displacement.

In order to further reduce the complexity, we have tested the framework with the same two matching metrics m_1 and m_2, but reducing the search window to a vertical (horizontal for m_2) line. A single distance line is thus computed to approximate the whole distance map. Surprisingly, the framework is able to select points and associated metrics such that few tens of points are sufficient to predict the target displacement sufficiently well (with a matching complexity for each point that is considerably reduced). Fig. 3 (right) shows the results using this particular metrics set. Again, the complexity unit corresponds to the matching of one point.

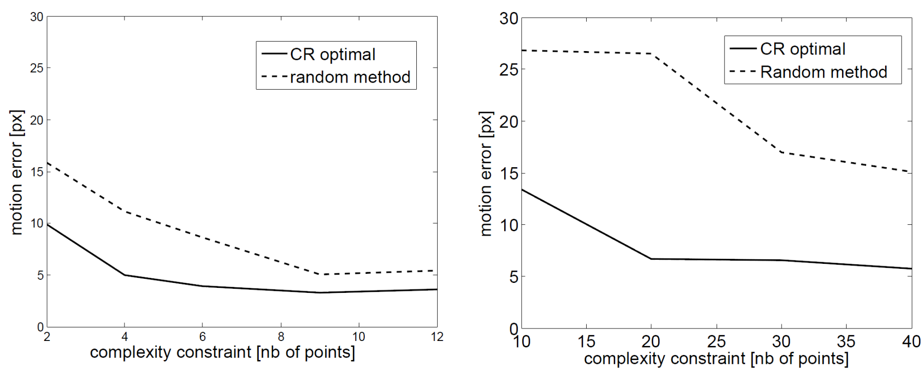

Fig. 3. Adapting the matching metric to the point neighborhood and selecting only reliable points allows infering a correct displacement at a lower computational cost

These results demonstrate the benefit of using the presented framework, particularly when the cost constraint is critical. Indeed, in the first approach, 4 points are sufficient for estimating the displacement with the CR-optimal method against 9 with the random method. In the second test case, 20 points generally suffice for a good motion estimation with the proposed method, while 30 points are needed with the random approach to perform significantly better than a simple guess on the displacement.

Interestingly, the results are even better for estimation the motion of a car between two pairs of frames, which is not surprising considering the number of visual structures of the vehicles. On the contrary, the performances are degraded when the method is applied on sequences where the target undergoes severe deformations compared with the target size.

7 Conclusion

We have presented a formulation of the tracking problem within a points matching framework. We showed that it was possible to optimize the reliability of the target displacement estimation, under some complexity constraint, by adapting the point matching strategy to each point regarding its neighborhood content. The approach differs from classical methods in that points are not selected based on some intuitive criterion, but are chosen so as to minimize the probability of erroneous target global displacement inference. We have shown the validity of our approach on a practical example, in which the combination of a few points and associated "cheap" matching metrics were sufficient to infer the targets displacements correctly between pairs of consecutive video frames.

References

1. Bay, H., Ess, A., Tuytelaars, T., Van Gool, L.: Speeded-up robust features (surf). Computer vision and image understanding 110(3), 346–359 (2008)
2. De Neyer, Q., Sun, L., Chaudy, C., Parisot, C., De Vleeschouwer, C.: Demo: Point matching for ptz camera autotracking. In: ICDSC. IEEE (2012)
3. Denman, S., Fookes, C., Sridharan, S.: Improved simultaneous computation of motion detection and optical flow for object tracking. In: Digital Image Computing: Techniques and Applications, 2009. DICTA'09. pp. 175–182. IEEE (2009)
4. Harris, C., Stephens, M.: A combined corner and edge detector. In: Alvey vision conference. vol. 15, p. 50. Manchester, UK (1988)
5. Hu, W.C.: Adaptive template block-based block matching for object tracking. In: Intelligent Systems Design and Applications, 2008. ISDA'08. Eighth International Conference on. vol. 1, pp. 61–64. IEEE (2008)
6. Lalonde, M., Foucher, S., Gagnon, L., Pronovost, E., Derenne, M., Janelle, A.: A system to automatically track humans and vehicles with a ptz camera. In: Defense and Security Symposium. International Society for Optics and Photonics (2007)
7. Lowe, D.: Object recognition from local scale-invariant features. In: ICCV. pp. 1150–1157 (1999)
8. Martello, S., Toth, P.: Knapsack problems: algorithms and computer implementations. John Wiley & Sons, Inc. (1990)
9. Papanikolopoulos, N.P., Khosla, P.K., Kanade, T.: Visual tracking of a moving target by a camera mounted on a robot: A combination of control and vision. Robotics and Automation, IEEE Transactions on 9(1), 14–35 (1993)
10. Sun, L., De Neyer, Q., De Vleeschouwer, C.: Multimode spatiotemporal background modeling for complex scenes. In: EUSIPCO (2012)
11. Veenman, C.J., Reinders, M.J.T., Backer, E.: Resolving motion correspondence for densely moving points. Pattern Analysis and Machine Intelligence, IEEE Transactions on 23(1), 54–72 (2001)
12. Xie, Y., Lin, L., Jia, Y.: Tracking objects with adaptive feature patches for ptz camera visual surveillance. In: Pattern Recognition (ICPR), 2010 20th International Conference on. pp. 1739–1742. IEEE (2010)
13. Zhang, Y., Liang, Z., Hou, Z., Wang, H., Tan, M.: An adaptive mixture gaussian background model with online background reconstruction and adjustable foreground mergence time for motion segmentation. In: Industrial Technology, 2005. ICIT 2005. IEEE International Conference on. pp. 23–27. IEEE (2005)