STANDALONE EVALUATION OF DETERMINISTIC VIDEO TRACKING

Juan C. SanMiguel[1], Andrea Cavallaro[2], José M. Martínez[1]

[1]Universidad Autónoma of Madrid (Spain) [2]Queen Mary University of London (United Kingdom)

ABSTRACT

We present an approach for performance evaluation of deterministic video trackers without ground-truth data. The proposed approach detects if a tracker is correctly operating over time using two main steps. First, it transforms the output of the localization step into a distribution of the target state, which emulates a multi-hypothesis tracker. Then, the uncertainty of such distribution is estimated to determine the time instants when the tracker is stable. A time-reversed analysis is used to identify tracker recovery after unsuccessful operation. The proposed approach is demonstrated on the well-known MeanShift tracker. The results over a heterogeneous dataset show that the proposed approach outperforms the related state-of-the-art methods in presence of tracking challenges such as occlusions, illumination and scale changes, and clutter.

Index Terms— performance evaluation without ground-truth, visual tracking, uncertainty estimation

1. INTRODUCTION

Video tracking faces many challenges related to geometric (pose, scale, occlusions) and photometric (clutter, appearance, illumination) factors [1], which lead to tracking failure (i.e. when the tracker loses the position of the target). Standalone evaluation operates without the need of ground-truth data allowing to detect tracking failures and recoveries after failure (i.e. when the tracker lock back on the target after a time interval of failure).

Standalone evaluation approaches for multi-hypothesis tracking have proven their superior performance as compared to single-hypothesis ones [2][3][4]. Multi-hypothesis approaches require estimating the posterior distribution of the tracked target and cannot be applied directly to evaluate *deterministic (single-hypothesis)* tracking. Some works addressed this limitation through adaptations of deterministic tracking. For example, [2][5] applied the time-reversibility constraint to MeanShift tracking [6] as the spatial overlap between the target estimations of the generated trajectory and its reversed version. For template-based tracking [7], [8] proposed a probabilistic view of single-hypothesis tracking and [9] estimated the uncertainty of the posterior distribution by a Gaussian fitting process on the tracker correlation surface. However, these approaches exhibit limitations related to error accumulation [5], computational cost [2] or applicability to low complexity videos only [8][9].

In this paper, we present a standalone performance evaluation approach that adapts a multi-hypothesis strategy [4] to *deterministic* tracking by converting its single-hypothesis localization process into a distribution of the target state which emulates a multi-hypothesis tracker. Then we estimate the uncertainty of such distribution to detect its temporal stability and use time-reverse analysis for checking tracker recovery after failure as in [4]. We refer to the proposed approach as Deterministic Adaptive Reverse Tracking Evaluation (DARTE). We demonstrate the proposed approach with the MeanShift tracker [6] and compare the results with related state-of-the-art approaches. The scheme of DARTE is depicted in Fig. 1.

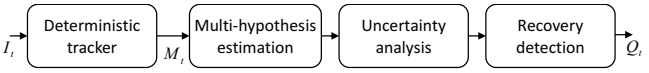


Fig. 1. Scheme of the proposed approach for standalone evaluation of *deterministic* video trackers. I_t: input video sequence, M_t: output of the localization step of the tracker, Q_t: the tracker evaluation result (*successful*, *unsuccessful*).

This paper is organized as follows: Section 2 and 3 define, respectively, the multi-hypothesis distribution estimation and the tracking evaluation of the proposed approach; Section 4 presents the experimental results and Section 5 concludes this paper.

2. MULTI-HYPOTHESIS ESTIMATION

Let x_t be the tracked state generated through a deterministic localization process on a n-dimensional matrix M_t that defines where the target is more likely to be located. In this work, we focus on the case when the state is only composed of the target position and therefore, $M_t(u, v)$ is a 2D surface obtained as:

$$M_t(u, v) = f(I_t, x_{t-1}, \beta) \tag{1}$$

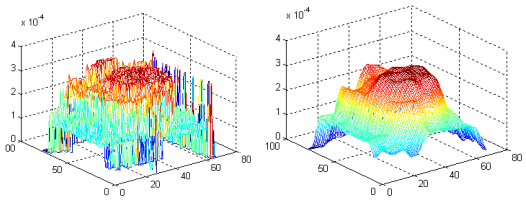
where the pair (u, v) defines each 2D position of $M_t(u, v)$, I_t is the video frame at time t, x_{t-1} is the tracked state at time $t-1$, β is the model of the target and $f(\cdot)$ is the process to generate $M_t(u, v)$ according to a similarity function (e.g., the histogram-based color similarity map of the MeanShift tracker [6] or the sum of squared differences of the template-based tracker [7]).

The aim of the proposed approach is to evaluate $M_t(u, v)$ for detecting when the algorithm is following the target (*successful*) or locked on background (*unsuccessful*). $M_t(u, v)$ is converted into a distribution of the target state, $p(x_t/z_{1:t})$, where $z_{1:t}$ are the observations up to time t, to emulate the output of a multi-hypothesis tracker, thus allowing the use of [4]. We evaluate the tracking data by analyzing the distribution uncertainty and using an additional tracker in reverse direction to check tracker recovery after a failure.

For estimating the multi-hypothesis distribution of the target state $p(x_t/z_{1:t})$ from $M_t(u, v)$, we start from the probabilistic view of deterministic tracking [8], which demonstrates that $p(x_t/z_{1:t})$ is proportional to the sampling of $M_t(u, v)$:

$$p(x_t/z_{1:t}) \approx \sum_{i=1}^{N} \omega_t^i \delta(x_t - x_t^i), \tag{2}$$

Work partially supported by the Spanish Government (TEC2011-25995 EventVideo), by the Consejería de Educación of the Comunidad de Madrid, and by The European Social Fund.



(a) (b)

Fig. 2. Example of the (a) 2D surface $M(u,v)$ for target localization and (b) its smoothed version $M_S(u,v)$ using a Gaussian kernel ($h = 9$ and $\sigma = 3$) for the MeanShift tracker [6].

where ω_t^i, x_t^i and N are, respectively, the weights, locations and the number of samples obtained from $M_t(u,v)$ at time t. However, [8] does not describe the extraction process for the samples and weights.

We propose to develop a sampling strategy for $M_t(u,v)$ to obtain the locations, x_t^i, and weights, ω_t^i, of the samples in order to estimate the multi-hypothesis distribution required for the standalone evaluation of tracking data. For notation simplicity, we have omitted the time index t from the following equations of this subsection.

First, we reduce the effect of impulsive noise in the video frames (consequently in $M(u,v)$) by using a 2D Gaussian low-pass filter:

$$M_S(u,v) = M(u,v) * K_G(h,\sigma), \tag{3}$$

where K_G is the Gaussian kernel with h size and σ standard deviation. An example of this noise reduction is shown in Fig. 2. Then, we create a confidence map, $C(u,v)$, that represents high (low) probability of target location with values close to 1 (0):

$$C(u,v) = f\left(\frac{M_S(u,v) - min\left(M_S(u,v)\right)}{max\left(M_S(u,v)\right)}\right), \tag{4}$$

where $f(\cdot)$ is a function that maintains ($f(x) = x$) or reverses ($f(x) = 1 - x$) if $M_S(u,v)$ indicates the similarity of the video frame with the target model with, respectively, high (e.g., high color similarity of MeanShift [6]) or low (e.g., low sum of squared differences of template matching [7]) values. Observe that the argument of $f(\cdot)$ in (4) defines the scaling of the $M_S(u,v)$ values to the range $[0,1]$. Thus, $f(\cdot)$ assures that $C(u,v)$ values close to 1 indicate high similarity between the video frame and the target model.

After extracting $C(u,v)$, we obtain a sampled map that defines the multi-hypothesis distribution, $C_P(u,v)$, as:

$$C_P(u,v) = C(u,v) \cdot s(u,v), \tag{5}$$

where $s(u,v)$ is the sampling signal determined by the sampling strategy defined as:

$$s(u,v) = \sum_{(u^i,v^i)\in\Lambda_s} \delta(u-u^i, v-v^i), \tag{6}$$

where $x^i = (u^i, v^i)$ is the i^{th} sample of the multi-hypothesis distribution derived by the sampling structure Λ_s.

For defining Λ_s, we assume that $M(u,v)$ represents the search area composed of the 2D estimated target location, $M_1(u,v)$, and the rest of the search area, $M_2(u,v)$. Hence, we are interested in measuring the amount of information relevant to the tracking evaluation task that is included in $M_1(u,v)$ and $M_2(u,v)$. Moreover, we want to study whether the estimated target center and its surroundings allow to evaluate the tracker or the whole information in $M_1(u,v)$ is required for this task. We propose the following sampling strategies for covering the previously mentioned aspects: (i) all the locations of $M(u,v)$ (S_0); (ii) around the peak of $M(u,v)$ as proposed by [6](S_1); (iii) all the locations of $M_1(u,v)$ and one out of four locations of $M_2(u,v)$ (S_2); (iv) all the locations of $M_1(u,v)$ (S_3); (v) all the locations of the log-polar transform [7] of $M(u,v)$ (S_4) and (vi) all the locations of the polar transform [7] of $M(u,v)$(S_5).

The sample weights of the multi-hypothesis distribution, ω^i, are extracted from $C_P(u,v)$ considering that each sample can have different importance (e.g., depending on its distance to the estimated target center) by using a weighting kernel:

$$C_W(u,v) = C_P(u,v) \cdot K(u,v), \tag{7}$$

where $K(u,v)$ is a non-negative real-value function that assigns a value in the range $[0,1]$ for each (u,v). For choosing $K(u,v)$, we consider the Uniform (U) and Epanechnikov (E) kernels [1]. The former assigns an equal weight to all samples whereas the latter gives a weight inversely proportional to the distance between each sample and the center (in our case, we use the maximum value for S_1 sampling and the estimated target center for S_0, S_2, S_3, S_4 and S_5 sampling). Finally, we obtain the weights ω^i by normalizing $C_W(u,v)$:

$$\omega^i = C_W(u^i,v^i) / \sum_{u,v} C_W(u,v). \tag{8}$$

3. TRACKING EVALUATION

After estimating the multi-hypothesis distribution, we compute its spatial uncertainty at each time t as:

$$S_t = \sqrt[d]{det(\Sigma_t)}, \tag{9}$$

where Σ_t is the covariance matrix of $p(x_t/z_{1:t})$ [9], $det(\cdot)$ is the determinant of a matrix and d is the number of dimensions of x_t. We consider that the deterministic tracker only estimates the 2D location of the target center maintaining its size constant ($d = 2$).

For identifying when the tracker is stable (i.e., it is following the target), we study the changes of S_t within a time window of length λ. We compute two relative variations of uncertainty for the change of $S_{t-\lambda}$ with respect to S_t and vice-versa as defined in [4]. The former indicates low-to-high uncertainty changes whereas the latter represents high-to-low uncertainty changes. Two time window lengths are used for considering short-term and long-term changes (λ_1 and λ_2). As a result, four signals are computed by combining the two relative variations and the two window lengths. Then, they are thresholded for detecting the uncertainty transitions with three thresholds (τ_1, τ_2 and τ_3) as proposed in [4]. Finally, these detections are combined by means of a finite-state machine to decide the tracker condition: focused on the target, scanning the video frame for the target or locking on the target after a tracking failure [4].

Then, we use time-reversed analysis to check the tracker recovery when it focuses on an object after unsuccessful operation as it might be on a distractor (background objects with features similar to those of the target). This analysis is based on applying a tracker in reverse direction from this recovery instant until a reference point (the last time instant when the tracker was successful) [4]. Effective tracker recovery after failure is determined by thresholding (with τ_4) the spatial overlap between the tracker to be evaluated and the reverse tracker at the reference point. Note that the time-reversed analysis is required as the uncertainty is only able to determine if the tracker is following an object that might be the target or a distractor.

Dataset	Target	Size	Characteristics
CAVIAR	P1 – P4	384x288	IC, SC, C
PETS2001	P5 – P7	768x576	SC, O, C
PETS2009	P8 – P14	768x576	O, C
VISOR	F1 – F2	352x288	SC, C, O

Table 1. Summary of the evaluation dataset (Key. SC: Scale Changes. IC: Illumination Changes. O: Occlusions. C: Clutter.)

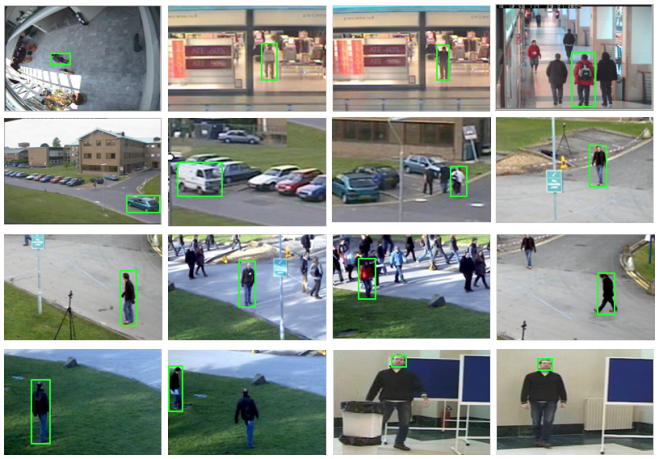


Fig. 3. Target initializations used in the evaluation dataset. (From top-left to bottom-right) pedestrians: P1, P2, P3, P4, P5, P6, P7, P8, P9, P10, P11, P12, P13 and P14. Faces: F1 and F2.

Finally, we combine the tracker condition and the time-reversed analysis using another finite-state-machine [4] to determine the tracker status between *successful* and *unsuccessful*. The former corresponds to the instants when the tracker is focused on the target (initial status of the tracker or after a correct recovery from tracking failure). The latter describes the case when the tracker is scanning the video frame or is focused on a wrong target (after an incorrect recovery from a tracking failure).

4. EXPERIMENTAL RESULTS

We evaluate the proposed sampling strategies and the overall approach (DARTE) for the widely used MeanShift tracker [6] on sequences from PETS2001[1], CAVIAR[2], PETS2009[3] and VISOR[4] datasets (Table 1). The target initializations are shown in Fig. 3. We use ROC analysis to measure the performance of the standalone evaluation (i.e., the detection of successful tracker operation) as the similarity between the obtained values and a ground-truth temporal segmentation. This segmentation defines the successful (unsuccessful) case when the spatial overlap between estimated and ground-truth target location is higher (lower) than 30%. For the the change detection analysis of S_t, we empirically defined τ_1 for each test and derived the other two ($\tau_2 = -\tau_1$ and $\tau_3 = \tau_1/2$). We used the values $\lambda_1 = 10$ and $\lambda_2 = 40$ for the time window length. For checking tracker recovery, we set the value $\tau_4 = 0.8$. For the smoothing kernel K_G, we used $h = 9$ and $\sigma = 3$.

[1]http://www.cvg.rdg.ac.uk/PETS2001/
[2]http://homepages.inf.ed.ac.uk/rbf/CAVIARDATA1/
[3]http://www.cvg.rdg.ac.uk/PETS2009/
[4]http://imagelab.ing.unimore.it/visor/

Sampling strategy	S_0	S_0	S_1	S_2	$\mathbf{S_3}$	S_3	S_4	S_4	S_5	S_5
W. Kernel	U	E	U	U	**U**	E	U	E	U	E
Param τ_1	0.12	0.13	0.15	0.18	**0.15**	0.15	0.99	0.90	0.12	0.16
AUC	0.719	0.787	0.753	0.802	**0.867**	0.832	0.808	0.815	0.755	0.786

Table 2. DARTE results for the MeanShift tracker (Key. AUC: Area Under the Curve. S_i: Sampling strategy as defined in Sec. 2. U: Uniform. E: Epanechnikov.)

Approach	AUC	Execution time (s)		
		mean	max	min
DARTE	0.867	0.077 ± 0.024	24.406	0.003
TIM [5]	0.576	0.852 ± 0.115	1.100	0.450
FBF [2]	0.826	35.305 ± 18.120	95.920	0.550
ENT [8]	0.666	0.005 ± 0.002	0.034	0.001
FSU [9]	0.709	0.004 ± 0.003	0.096	0.002

Table 3. Comparison of DARTE with state-of-the-art approaches for the MeanShift tracker. (Key. AUC: Area Under the Curve. TIM: frame-by-frame reverse-tracking [5]. FBF: full-length reverse-tracking [2]. ENT: entropy [8]. FSU: spatial uncertainty [9].)

Moreover, we compare DARTE with related standalone tracking evaluation approaches based on frame-by-frame reverse-tracking using template matching (TIM) [5], full-length reverse-tracking using the same trackers for forward and reverse analysis (FBF) [2], entropy of $M(u, v)$ (ENT) [8] and spatial uncertainty of $M(u, v)$ (FSU) [9].

The results for the proposed sampling strategies S_i and weighting kernels are listed in Table 2. The use of the Gaussian kernel K_G is justified by the improvement achieved for the sampling strategy S_0, the Uniform weighting kernel and the threshold $\tau_1 = 0.12$, obtaining an Area Under the Curve (AUC) of 0.719 and 0.655 for, respectively, with and without smoothing. Sampling around the maximum value of $M(u, v)$ (S_1) got the worst results demonstrating that the location of this maximum value lacks smoothness over time as it might be affected by noise, appearance changes and distractors. Sampling in the search area (S_0, S_1, S_2, S_4 and S_5) is less robust as it may contain distractors. Hence, S_3 presented the best results as it is restricted to target location. Among the weighting kernels, the Uniform kernel is preferable to the Epanechnikov one when no data of the search area are contained in the extracted samples.

The comparison with the selected state-of-the-art approaches is summarized in Table 3. TIM got low results due to its adaptation to tracking failures as the tracker tends to focus on distractors and therefore, TIM compares wrong target estimations with wrong reverse analysis. FBF obtained high performance. However, it is affected by the drift of the target estimation and, therefore, the accuracy of the reverse analysis is reduced. Its main limitation is the high computational cost with exponential dependency on the number of frames. For ENT, as tracking degrades and clutter appears, the unimodality of the posterior distribution is converted into multimodality. Thus, ENT had low performance. FSU showed that using the uncertainty of $M(u, v)$ only is not accurate as it solely indicated that the tracker was focused on an object (that could be either the target or a distractor). DARTE improved the selected approaches solving the above mentioned problems whilst achieving a bounded execution time.

An standalone evaluation example for F1 target is shown in Fig. 4. As the ground-truth error signal indicates, the tracker failed four times being not capable of estimating the correct target location due to occlusions with the blue blackboards (the first and third ones), an

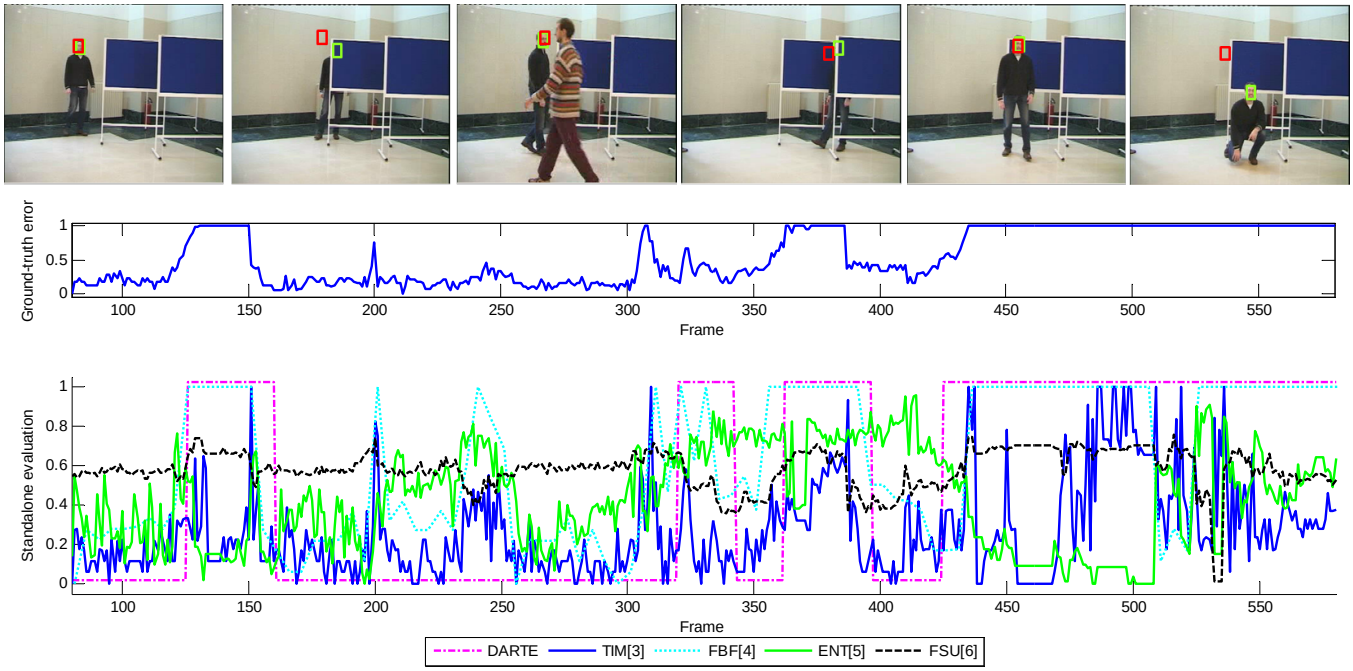


Fig. 4. Ground-truth error and standalone evaluation results for F1 target of sequence *VISOR_1* (frames 100, 125, 300, 375, 425 and 450) with the MeanShift tracker. Ground-truth data and tracking results are represented as green and red rectangles, respectively. The ground-truth error is measured as the spatial overlap between estimated and ground-truth target. (Key. TIM: frame-by-frame reverse-tracking [5]. FBF: full-length reverse-tracking [2]. ENT: entropy [8]. FSU: spatial uncertainty [9].)

occlusion with a similar moving target (the second one) and quick target movement (the fourth one). TIM showed its ability to detect quick changes of target position with high values (first frames of each tracking error). However, it demonstrated its adaptation to tracking failures by having low values for the first, second and fourth tracking errors (when the tracker was focused on background objects). Thus, it had low performance. FBF presented good results correctly indicating the tracking failures. However, it failed by detecting no tracking error for frames 510-525 and two tracking errors at frames 200 and 245. ENT obtained low performance showing lower values for the first and fourth errors whereas high values for the second and third errors. Moreover, it also presented high (frames 200-250) and low (frames 250-300) values for the successful tracking case (between frames 200 and 250). FSU also presented low performance having high and low values for successful and unsuccessful tracking. DARTE demonstrated its superior performance detecting the four tracking errors. However, a delay for identifying the second failure was observed due to the required amount of uncertainty change for detecting that the tracking data is not stable (controlled by τ_1).

5. CONCLUSIONS

We have presented an approach for standalone performance evaluation of *deterministic* trackers. The proposed approach is based on estimating a multi-hypothesis posterior distribution from the data provided by deterministic tracking, analyzing its uncertainty for detecting stable tracking data and using the time-reversibility constraint for checking tracker recovery after losing the target. The proposed approach was validated on the widely used MeanShift tracker over standard sequences. The results showed that the proposed approach outperforms the selected state-of-the-art approaches.

As future work, we will focus on automatic thresholding to detect changes in the uncertainty of tracking data and the use of the proposed approach for other deterministic trackers.

6. REFERENCES

[1] E. Maggio and A. Cavallaro. *Video Tracking: Theory and Practice*. Wiley, 2011.

[2] H. Wu, A. Sankaranarayanan, and R. Chellappa. Online empirical evaluation of tracking algorithms. *IEEE Trans. Pattern Anal. Mach. Intell.*, 32(8):1443–1458, Aug. 2010.

[3] J.C. SanMiguel, A. Cavallaro, and J.M. Martínez. Evaluation of on-line quality estimators for object tracking. In *Proc. of IEEE ICIP*, pages 825–828, 26-29 Sept. 2010.

[4] J.C. SanMiguel, A. Cavallaro, and J.M. Martínez. Adaptive online performance evaluation of video trackers. *IEEE Trans. Image Process.*, 21(5):2812 –2823, May 2012.

[5] R. Liu, S. Li, X. Yuan, and R. He. Online determination of track loss using template inverse matching. In *Proc. of VS*, pages 1–8, 17 Oct. 2008.

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

[7] R. Brunelli. *Template Matching Techniques in Computer Vision: Theory and Practice*. Wiley, 2009.

[8] V. Badrinarayanan, P. Perez, F. Le Clerc, and L. Oisel. On uncertainties, random features and object tracking. In *Proc. of IEEE ICIP*, pages 61–64, 16-19 Sept. 2007.

[9] K. Nickels and S. Hutch. Estimating uncertainty in ssd-based feature tracking. *Image Vis. Comput.*, 20(1):47–58, Jan. 2002.