

Real-Time Descriptorless Feature Tracking^{*}

Antonio L. Rodríguez, Pedro E. López-de-Teruel, and Alberto Ruiz

¹ Dpto. Ingeniería y Tecnología de Computadores, University of Murcia (Spain)

² Dpto. Lenguajes y Sistemas, University of Murcia (Spain)

arodriguez@ditec.um.es, {pedroe,aruiz}@um.es

Abstract. This paper presents a simple and efficient estimator of long-term sparse optical flow. It is supported by a novel approach to feature tracking, essentially based on global coherence of local movements. Expensive invariant appearance descriptors are not required: the locations of salient points in successive frames provide enough information to create a large number of accurate and stable tracking histories which remain alive for significantly long times. Hence, wide-baseline matching can be achieved both in extremely regular scenes and in cases in which corresponding points are photometrically very different. Our experiments show that this method is able to robustly maintain in real time hundreds of trajectories in long video sequences using a standard computer.

1 Introduction

Robust detection and tracking of a large number of features along video sequences is a fundamental issue in many computer vision systems. Applications such as visual navigation, 3D reconstruction, visual SLAM or augmented reality, among many others, commonly base their operation on precise landmark tracking. Physical points are usually chosen for this task, as there are many well known efficient algorithms for detecting them from scratch (see [1] for a good review) as well as for tracking them in time (based mainly in the local photometric structure of the image around the detected point, such as in the classical approach [2]). Another reason is their superior geometrical relevance, which allows to get more interimage constraints. Using the 2D position of the tracked points along the sequence, the 3D structure of the original landmarks can be inferred, taking advantage of well established multiple view geometry results [3] [4] and on-line bayesian approaches [5] [6].

More recently, new multiscale interest point detectors and associated descriptors for matching have also been developed, perhaps being SIFT [7] and MSER [8] the most known and widely used. These are clearly more powerful for wide baseline matching, but are also more computationally expensive and hence less amenable for real time systems running in standard low cost off-the-shelf hardware. (Considerable research effort is being devoted to obtain efficient implementations in alternative computing platforms like GPUs [9] or FPGAs [10]). In

^{*} This work has been supported by the Spanish MEC and European FEDER funds under grants “TIN2006-15516-C04-03” and Consolider “CSD2006-00046”.

any case, their multiscale invariance makes these features more appropriate for object recognition and classification under very different viewpoints.

In this paper, however, we are interested in continuous tracking, where the key issues are (a) precise localization of points at the finest possible scale, (b) a fast and robust detection/tracking procedure and (c) the ability to track large number of points along the maximum possible number of frames. Point matching algorithms are classically based on comparisons of local image patches (whether using the original, normalized, filtered, gradient, or other kind of more or less complex preprocessing of each patch). This adds some computational load to the tracking system which does not scale well with the number of features. Thus, they are commonly used for off-line video processing applications [3], but are less amenable for real time continuous tracking. Anyway, when largely simplified [6], or when used only on a small and well conditioned set of selected points [5], such methods are remarkably successful.

In contrast, we focus on pure 2D location to improve efficiency and scalability in the number of tracked points¹. We completely discard the local photometric structure of the environment of each point, exploiting continuity and global coherence of the movement to guide matching. The proposed method uses a *detection stage* based on aggregated saliency of the Determinant of Hessian (DoH) operator [1], followed by the estimation of a coarse *holistic* feature motion model based on a variation [11] of RANSAC [12]. Our experiments show that this *descriptorless* approach is viable, efficient and scalable, allowing for long-term tracking of 3D points whose 2D appearance changes along the sequence.

The paper is organized as follows. Section 2 gives an overview of the proposed method. Then, sections 3 and 4 respectively describe in detail the two main stages, saliency detection and robust tracking. In section 5 we present performance measurements in realistic scenarios, which justify the validity of the approach. Finally, section 6 provides the main conclusions of this work.

2 General Overview of the Approach

The proposed feature tracker works in three stages:

- First, salient points are detected in the input image to obtain an initial set of features suitable to be tracked. Any kind of point detector could in principle be used in this stage, but for better performance the detector should be repeatable and stable, and must obtain accurate point locations. This is important for our *descriptorless* approach, as the posterior matching process will be based only in position information. In order to get a good balance between speed and accuracy, we will use a simple saliency operator based on local maxima of aggregated image nonlinearity, as measured by the DoH operator [1], which shows remarkable repeatability. This detector will be further discussed in Sec. 3.

¹ The system described in [6] also focuses on scalability, but rather in the map of tracked 3D features and the associated uncertainty management procedure than in the process of detection and tracking of 2D points itself.



Fig. 1. Illustration of the alignment process. Left and center: salient points and a rectangle indicating the best metric transformation relating two frames. (For clarity we only show a few points and very distant, non successive frames). Right: Green points aligned and superimposed on the red points. Observe that most points are closely aligned, while for some others there is no matching.

- In a second stage we estimate the 2D metric motion model which best aligns the points detected in successive frames. This is a clearly incorrect model for the 2D projection of most movements in a real 3D scene. Even so, a 2D projective model for the local movement of image features is approximately correct if the camera centers are close in successive images. If the angles between optical axes are also small, such homography approximately reduces to a metric transformation. Therefore, these approximations are acceptable for reasonably continuous interframe movements.

Note that due to point detection failures and the limitations of the simplified motion model, in many cases some of the detected points will not have a matching counterpart in the previous image. Some kind of sample consensus estimation [12] is required. To allow for real-time computation, only a reduced set of the most salient points is used at this stage (much like in the PROSAC technique described in [11]). A simple motion predictor assuming constant velocity guides and bounds the search. Section 4 describes this robust estimator in detail.

- In the last stage we compute the largest set of point correspondences consistent with the estimated metric transformation. Figure 1 illustrates point alignment for two sample images.

3 Salient Point Detector

Our point detector is based on the Determinant of the Hessian (DoH) filter. For an input image $I(x, y)$, the value of the response image is defined as follows:

$$H(x, y) = \begin{vmatrix} \frac{\partial^2 I(x, y)}{\partial^2 x} & \frac{\partial^2 I(x, y)}{\partial x \partial y} \\ \frac{\partial^2 I(x, y)}{\partial x \partial y} & \frac{\partial^2 I(x, y)}{\partial^2 y} \end{vmatrix} \quad (1)$$

The determinant is strongly positive on blob-like regions [1,7], and negative on saddle points. Both cases indicate significant non-linearity in the locality of a point and are equally important for feature detection. We are interested in regions which, due to the presence of a blob, a corner, a textured zone, or similar

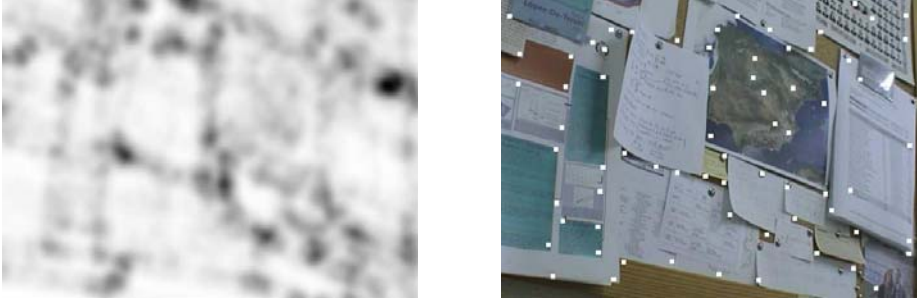


Fig. 2. Illustration of the saliency operator. Left: output of the smoothed absolute value of the DoH ($\sigma_{post}=6$). Darker zones correspond to locally nonlinear regions. Right: strong local maxima of the operator indicate stable and repeatable features.

situations, cannot be well approximated linearly. Poorly localized structures like flat gray level zones and simple straight edges must be discarded. In order to detect strongly salient nonlinear regions we take a local average of the absolute value of the DoH response in (1). We apply a previous gaussian filter to the input image with σ_{pre} to reduce noise and select the working scale. In practice $\sigma_{pre} \approx 2.0$ or 4.0 is acceptable in most cases, and in principle the algorithm is not very sensitive to this value.

After obtaining the absolute DoH response image, we estimate the aggregated local saliency by a second gaussian filter, this time using a larger σ_{post} . The effect of this parameter is grouping small textured zones in a unique maximum response point. Thus, increasing σ_{post} will hopefully make the position of the local maxima of the response more stable, possibly at the cost of diminishing the number of totally tracked points. In practice, σ_{post} should be chosen such that the positive and negative contiguous DoH peaks in a corner are joined in just one output peak. Values of σ_{post} around 6.0 or 8.0 are in principle adequate for small/medium scale point detection, though this parameter can be tuned to work at varying levels of detail (more on this on the experiments section). Figure 2 illustrates our saliency operator on a sample image.

As we will see in the next section, the alignment process requires detected points being sorted by relevance. The smoothed absolute value of the DoH filter response at each point, which is closely related to detection repeatability in successive frames, can be directly used for this task.

4 Robust Mapping Estimation

Simplified metric 2D motion model. Under the global metric model assumed for feature motion, the locations of two corresponding points $(x, y, 1)$ and $(x', y', 1)$ are related by the following expression:

$$\begin{pmatrix} x' \\ y' \\ 1 \end{pmatrix} = \begin{pmatrix} s \cos(\alpha) & -s \sin(\alpha) & \Delta x \\ s \sin(\alpha) & s \cos(\alpha) & \Delta y \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix} \quad (2)$$

This model has four parameters: global scale s , rotation α , and displacements Δx and Δy . They are completely determined from just two point correspondences.

PROSAC motion model estimation. Robust estimation proceeds as follows. In order to increase the probability of finding a valid motion model earlier in the search procedure, we initially select only the (hopefully) most repeatable points from each image. In our case, we use the N_{max} points with greater absolute value of the local maxima of the DoH. A naive initial approach would result then in a maximum of $N_{max} \times N_{max}$ candidate pairs², though in practice, as we will see shortly, we will reduce this number by more than an order of magnitude using a simple and efficient heuristic.

To get an initial guess for interimage correspondence, we assume that the camera performs a smooth movement. Therefore, the metric mapping to be estimated for the current image will be similar to that of the previous one³. Given a candidate point correspondence $(x, y) \mapsto (x', y')$, and the current metric motion model $E^{(t)} = (z^{(t)}, \alpha^{(t)}, \Delta x^{(t)}, \Delta y^{(t)})$, we define the following distance:

$$\mathcal{D}\left(E^{(t)}, (x, y) \mapsto (x', y')\right) = \left\| \begin{pmatrix} x' \\ y' \\ 1 \end{pmatrix} - \begin{pmatrix} z^{(t)} \cos(\alpha^{(t)}) & -z^{(t)} \sin(\alpha^{(t)}) & \Delta x^{(t)} \\ z^{(t)} \sin(\alpha^{(t)}) & z^{(t)} \cos(\alpha^{(t)}) & \Delta y^{(t)} \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix} \right\|$$

Instead of using every possible matching in the search, we can use only the matchings between each point p_0 in the previous image and the n points (where n is a constant value) closer to its estimated location $E^{(t)}p_0$ in the current image. This speeds-up the search considerably, because the number of candidate matchings is reduced from $N_{max} \times N_{max}$ to $n \times N_{max}$. Our tests indicate that a value of $n = 4$ is enough to obtain the same results as using the whole $N_{max} \times N_{max}$ candidate matchings set in the PROSAC search⁴.

The $n \times N_{max}$ candidates are then sorted according to the following simple heuristic value:

$$\mathcal{L}\left(E^{(t)}, (x, y) \mapsto (x', y')\right) = e^{-\mathcal{D}\left(E^{(t)}, (x, y) \mapsto (x', y')\right)}$$

A high value of \mathcal{L} for a given matching indicates that the correspondence is more likely to be valid, and should be used earlier in the sample consensus search.

In each step, two possible correspondence pairs are taken from the list of sorted candidates to compute a new tentative metric movement model $E^{(t+1)}$.

² For example, a value of $N_{max} = 50$ leads to 2500 candidate matchings.

³ A more elaborate Kalman filter estimator could also be used, possibly taking into account angular and linear velocities. In practice, our simple constant velocity predictor works remarkably well even for moderate camera accelerations, and our tests indicate that the Kalman filter does not produce a significant improvement in the tracker accuracy.

⁴ For $N_{max} = 50$ and $n = 4$, this reduces the number of candidates matching from 2500 to only 200.

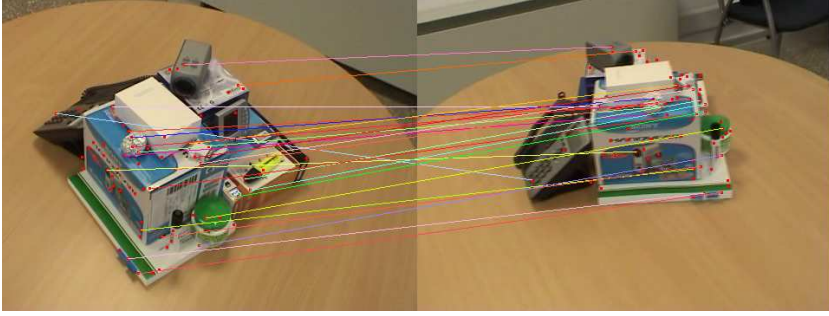


Fig. 3. Wide baseline matching after continuous tracking of 100 frames. For clarity only a small number of correspondences are shown. Observe the very different local photometric structure of corresponding points in most matchings.

This model is evaluated against the remaining correspondences in the list. If the model explains (up to a given distance threshold τ) the motion of a sufficient number (N_{min}) of features, the search finishes and the model is considered to be valid. Again, any sensible approximate choice of these parameters will work without affecting too much the behaviour of the algorithm (say, for example, $\tau \approx 10$ pixels, and $N_{min} \approx 0.5N_{max}$, depending also on the quality of the optics and the clarity of the images obtained by the camera). Observe that local feature descriptors are not required for matching, even if the interframe displacements are larger than the typical distances between features in a given frame. Global motion coherence is sufficient to estimate an initial set of valid correspondences just from position information.

This is a straightforward application of PROSAC [11], a variation of the RANSAC robust estimation technique which uses heuristic information to significantly increase the probability of finding a valid model earlier in the search. This is mandatory in real-time applications.

Global set of matches and model refinement. The estimated motion model is finally used to find the largest set of valid correspondences in the whole list of points (i.e., not limited to the N_{max} most salient ones). We consider a correspondence $p_i \mapsto q_j$, (with p_i in the first image, and q_i in the second) to be valid if (a) the closest point to $E^{(t+1)}p_i$ in the second image is q_j , (b) the closest point to $E^{(t+1)-1}q_j$ in the first image is p_i , and (c) the distance between $E^{(t+1)}p_i$ and



Fig. 4. Sample frames taken from test videos. Left to right: *floor-texture.avi*, *paper-mountain.avi*, *table-1.avi*, *table-2.avi*, *template-2.avi*.

the q_j is smaller than a given threshold (we can use τ again in this step). The resulting matches can be used to reestimate the metric interframe motion model.

Continuous operation and matching hysteresis. Only features that are tracked through a sufficient number of frames are considered as correctly matched. This way, remaining outliers due to spurious features are finally discarded, while all the correctly tracked inliers remain stable even if their local image patches have radically changed (see Fig. 3).

An additional advantage of this approach is that it allows for easy recovery of lost features due to occlusion or temporary detection failure. If a feature is not detected in a given frame, the tracker keeps on estimating its location at each new frame using the global motion model. If it appears again early in the video sequence, tracking continues as if the point was never lost. Otherwise, after a given number of frames (UF_{max} , for maximum number of unmatched frames) tracking for this point is utterly finished.

5 Experimental Results

Robustness and efficiency of the proposed tracking algorithm have been evaluated in several realistic scenarios and operating conditions. Figure 4 shows a few sample frames taken from five test videos available at [13]. Unless otherwise stated, we used the following parameter values: in the point detection stage, the final corner response image was thresholded with an absolute value of 1000, to quickly eliminate low-response points. To obtain a feature point list with a reasonable size, it will contain at most 300 of the points with best corner response

Table 1. Performance of different stages of the algorithm. Central columns values are in milliseconds. The last column indicates the mean number of points per frame used in the matching stage.

	Size	Detection	Alignment	Matching	#points
floor-texture.avi	320x240	10.871	0.575	5.931	299.115
	480x360	21.218	0.641	5.655	300
	640x480	37.596	0.824	5.062	300
paper-mountain.avi	320x240	8.961	0.493	2.483	138.869
	480x360	17.716	0.516	5.566	277.777
	640x480	32.268	0.513	6.069	300
table-1.avi	320x240	8.24	0.537	2.766	152.401
	480x360	18.443	0.547	6.007	293.754
	640x480	32.227	0.528	5.989	300
table-2.avi	320x240	9.166	0.527	2.327	131.038
	480x360	18.414	0.597	5.383	262.102
	640x480	32.014	0.571	6.38	299.86
template-2.avi	320x240	8.71	0.487	2.931	149.431
	480x360	18.28	0.506	5.531	256.965
	640x480	32.629	0.553	6.156	285.374

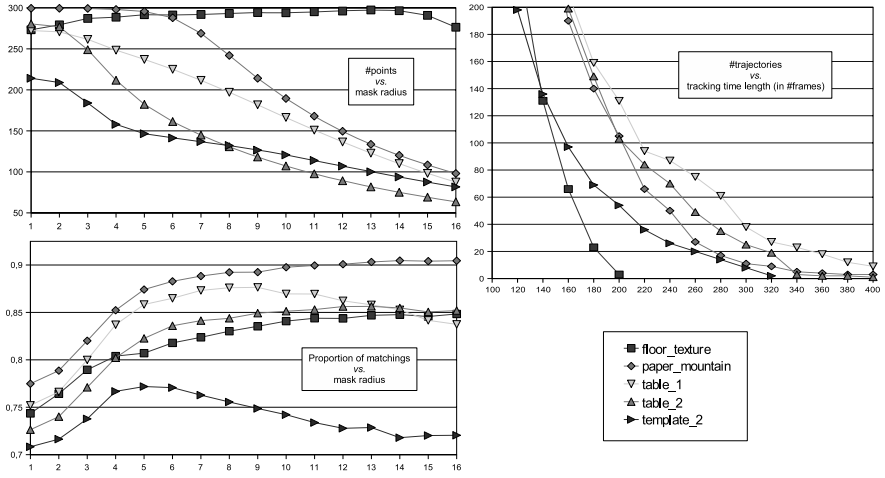


Fig. 5. Analysis of dynamic behaviour of the tracker: (Top left) σ_{pre} vs. mean number of points per frame. (Bottom left) σ_{pre} vs. mean proportion of matched points between consecutive frames. (Right) Total number and time lengths of the tracking trajectories.

image. The UF_{max} parameter for point hysteresis is set to 2 frames in the tests. The parameters σ_{pre} , σ_{post} and τ were set to 4.0, 8.0 and 8.0 respectively. Finally, the values $N_{max} = 30$, $N_{min} = 15$ achieve a good compromise between accuracy and computing time in the PROSAC stage.

All tests were performed on a 2.4 GHz Intel processor, using optimized low-level image processing routines from the Intel Integrated Performance Primitives (IPP) library.

Computing time performance: Table 1 shows the mean execution time per frame of the main computation stages, for three different typical input image sizes. Even for the largest 640×480 size, in every case the total computing time keeps below 40 ms (25 fps). The execution time also depends on the total number of points, so very textured videos, such as *floor-texture.avi*, show slightly slower processing rates. Anyway, the low level computation of saliency takes most of the computing time, so for smaller sizes (480×360 and 320×240), the detection stage becomes much faster. The posterior alignment and matching time depends on the total number of features rather than image size (though obviously these magnitudes are implicitly related).

Influence of σ_{post} : Obviously, this parameter has also a direct influence on the total number of points extracted in each frame. Figure 5 illustrates how smaller values of σ_{post} tend to increment the number of points (Fig. 5, top left), at the cost of making the detector less repeatable⁵. This clearly affects the proportion of matched points in successive frames (Fig. 5, bottom left).

⁵ The X axis in top and bottom left graphs of Fig. 5 actually corresponds to the mask radius in pixels, equal to $2.5\sigma_{pre}$.

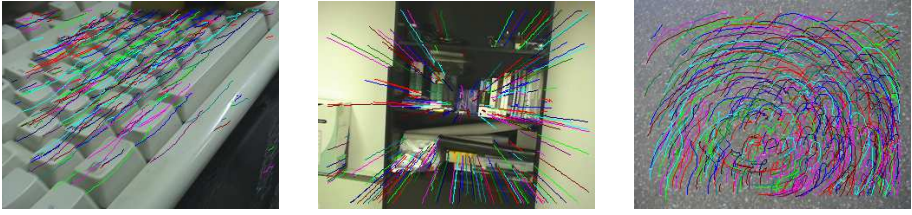


Fig. 6. Tracking examples (for clarity only a few trajectories are displayed). Left: Translation. Center: Zoom. Right: Rotation. By exploiting motion coherence we can track a large number of features in extremely regular textured scenes.

Tracking evaluation: Figure 5 (right) plots the number of tracked points vs. tracking time length (measured as consecutive frames). The algorithm obtains hundreds of correct trajectories during long time periods even in difficult scenes with large uniformly textured zones (*paper-mountain.avi*, *floor-texture.avi*). Note that classical approaches based on local patch descriptors are prone to fail in such situations. The shortest tracking trajectories were obtained in the *floor-texture.avi* and *template-2.avi* test videos, where, in the worst case, more than 100 points were tracked during more than 140 frames. Longer trajectories are obtained in sequences which focus on a specific object (*paper-mountain.avi*, *table-1.avi* and *table-2.avi*). The best results are obtained in the *table-1.avi* test, where most objects never get out of sight, nor are occluded by other objects. More than 60 points are tracked for more than 280 frames, and up to 10 points for more than 400 frames. Figure 6 shows some snapshots of the tracking algorithm in action. We encourage the reader to download from [13] recorded real-time demonstrations of the dynamic behavior of this method.

6 Conclusion

This work demonstrates that appearance descriptors are not required for long-term, stable estimation of sparse optical flow. Global motion coherence is sufficient to remove feature matching ambiguity, even in scenes with undifferentiated texture. This is an advantageous alternative for wide-baseline matching, since invariant keypoint characterization is achieved just by spatio-temporal continuous history.

The proposed tracking method is remarkably efficient in computation time: our experiments show that hundreds of points can be tracked during dozens of seconds in video sequences with rich camera movements. This approach opens up new possibilities for applying in real-time many geometric reconstruction algorithms based on interframe relationships.

References

1. Schmid, C., Mohr, R., Bauckhage, C.: Evaluation of interest point detectors. *International Journal of Computer Vision* 37(2), 151–172 (2000)
2. Shi, J., Tomasi, C.: Good features to track. *Proc. of IEEE Computer Vision and Pattern Recognition Conference*, 593–600 (1994)
3. Pollefeys, M., VanGool, L.: Visual modeling: from images to images. *Journal of Visualization and Computer Animation* 13, 199–209 (2002)
4. Hartley, R., Zisserman, A.: *Multiple View Geometry in Computer Vision*, 2nd edn. Cambridge University Press, Cambridge (2003)
5. Davison, A.J., Molton, N.D.: MonoSLAM: Real-time single camera SLAM. *IEEE Trans. on PAMI* 29(6), 1052–1067 (2007)
6. Rosten, E., Drummond, T.: Fusing points and lines for high performance tracking. In: *Proc. of the IEEE ICCV conference*, pp. 1508–1515 (2005)
7. Lowe, D.: Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision* 20, 91–110 (2003)
8. Matas, J., Chum, O., Urban, M., Pajdla, T.: Robust wide baseline stereo from maximally stable extremal regions. In: *Proc. of the BMVC*, pp. 384–393 (2002)
9. Heymann, S., Maller, K., Smolic, A., Froehlich, B., Wiegand, T.: SIFT implementation and optimization for general-purpose GPU. In: *Proc. of Int. Conf. in Central Europe on Computer Graphics, Visualization and Computer Vision* (2007)
10. Se, S., Nge, H., Jasiobedzki, P., Moyung, T.: Vision based modeling and localization for planetary exploration rovers. In: *Proc. of Int. Astronautical Congress* (2004)
11. Chum, O., Matas, J.: Matching with PROSAC - Progressive sample consensus. In: *Proc. of the IEEE CVPR*, pp. 220–226 (2005)
12. Fischler, M.A., Bolles, R.C.: Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM* 24(6), 381–395 (1981)
13. PARP Group homepage (2009), <http://perception.inf.um.es/tracker/>