Multi-target and multi-camera object detection with monte-carlo sampling

Giorgio Panin, Sebastian Klose, Alois Knoll

Technische Universität München, Fakultät für Informatik
Boltzmannstrasse 3, 85748 Garching bei München, Germany
{panin,kloses,knoll}@in.tum.de

Abstract. In this paper, we propose a general-purpose methodology for detecting multiple objects with known visual models from multiple views. The proposed method is based Monte-Carlo sampling and weighted mean-shift clustering, and can make use of any model-based likelihood (color, edges, etc.), with an arbitrary camera setup. In particular, we propose an algorithm for automatic computation of the feasible state-space volume, where the particle set is uniformly initialized. We demonstrate the effectiveness of the method through simulated and real-world application examples.

1 Introduction

Object detection is a crucial problem in computer vision and tracking applications. It involves a global search over the feasible state-space, and requires to cope with an unknown number of targets, possible mutual occlusions, as well as false measurements, arising from background clutter.

Using multiple cameras can greatly improve the detection results in terms of precision and robustness, since the joint likelihood will be much more focused on real targets, and mutual occlusions from a given view will be solved by the others. Moreover, multiple cameras constrain the state-space of visible objects to a smaller volume, where a target appears in all visual fields. This reduces the search space of a great amount, and therefore facilitates the detection process.

For this purpose, a typical *bottom-up* approach usually consists of sampling image features (e.g. segmenting color *blobs*) and matching them between cameras, in order to perform a 3D triangulation and object localization: however, this approach requires to explore all possible combinations of data that can be associated to similar targets, possibly in presence of missing detections and false alarms, as well as partial occlusions, which can make the problem of an intractable complexity.

In a *top-down* approach, instead, a detection task can be seen as a global optimization of a multi-modal *likelihood* function in state-space, which presents strong local maxima around each target (detected by the optmization method), as well as smaller peaks around false measurements. This optimization problem involves generating and testing a number of state-space *hypotheses*, by projecting

the relevant model features on each camera view, and comparing them locally with the image measurements.

When two targets are too close with respect to the covariance of the measurement noise, the related peaks merge to some extent, and are not anymore distinguishable by the search method. Therefore, the measurement covariance by itself sets a limit to the state-space *resolution* of the detector.

Evolutionary and Genetic Algorithms are well-known in the literature, in order to cope with such multi-modal optimization problems [9]; however, their computational complexity limits the application field, particularly when a real-time (or near real-time) performance is required, such as object tracking.

In order to approach the problem from a general point of view, not restricted to a particular form of the likelihood, or a given camera configuration, we choose instead a Monte-Carlo based strategy, followed by unsupervised clustering of state hypotheses, according to the respective likelihoods. This approach has the advantage of neither requiring any prior assumption about the number of targets, nor about the form of the likelihood, provided that a significant local maximum is present around each target state.

The paper is organized as follows: Section 2 describes the general clustering strategy, based on kernel representation and weighted mean-shift; Section 3 the introduces the uniform sampling strategy for multiple camera views, on the joint vieweing volume; Section 4 provides simulated and experimental results, and Section 5 concludes the work with proposed future developments.

2 The particle-based detector

In order to detect targets, we basically look for local maxima (or *modes*) of a given likelihood, provided by any visual property of each target, and a suitable matching strategy between model and image features. In general, this function can integrate multiple visual cues, as well as data from multiple cameras. Such a general formulation, together with an arbitrary number and relative location of targets, makes the estimation problem of a complex and nonlinear nature.

Therefore, we approach the problem by means of a general and flexible method, such as Monte-Carlo sampling. In particular, we represent our likelihood through a discrete *particle set* (s^i, w^i), where s_i are state hypotheses, weighted by their likelihood w_i. This representation is well-known in a tracking framework [7], and can cope with nonlinear and multi-modal distributions.

In absence of any prior information about the possible location of targets, the particle set is initialized with uniform distribution, covering the feasible state-space volume where targets can be viewed by the multi-camera setup (Sec. 3).

Each peak of the likelihood will provide a cluster of high-weighted particles around it, and therefore a weighted *state-space clustering* algorithm can be run, in order to identify them. However, if the likelihood peaks are too large and partially overlapping, the clusters will overlap as well, and the algorithm will fail to separate them properly.

In a computer vision application, the width of the likelihood modes depends on the modality used (edges, color, etc.), and on the related covariance. This parameter can be externally set (or internally computed), and it reflects the uncertainty of the measurement process in feature- or state-space: a high-resolution measurement has low covariance, with narrow peaks well-located around the targets, but also many local maxima in the neighborhood; on the other hand, a low-resolution measurement will have a higher covariance, larger and less localized peaks.

In order to identify the modes, we need a smooth representation that can be locally optimized, such as a *kernel-based* representation [2][4]. More in detail, if x is a one dimension variable, a weighted kernel density is represented by

$$p\left(x|\,\theta\right) = \frac{1}{N}\sum_{i=1}^{N}\frac{w_i}{h}k\left(\frac{x-x_i}{h}\right) \tag{1}$$

In this formula, k is the kernel, which has a maximum value in $x = 0$, and quickly decays in a neighborhood of the origin; h is called *bandwidth*, and regulates the width of the kernel around each point x_i. The number of *modes* N is also a parameter of this distribution, overall represented by the set of values

$$\theta = (N, h, x_1, ..., x_N, w_1, ..., w_N) \tag{2}$$

A typical choice for the kernel is the Gaussian distribution

$$k\left(x\right) = \frac{1}{\sqrt{2\pi}}e^{-\frac{x^2}{2}} \tag{3}$$

for which $h = \sigma$ is the standard deviation, so that (1) represents a Mixture of Gaussians. In a multi-dimensional space, the kernel representation generalizes to

$$p\left(\mathbf{x}|\,\theta\right) = \frac{1}{N}\sum_{i=1}^{N}\frac{w_i}{\det H}\mathcal{K}\left(H^{-1}\left(\mathbf{x}-\mathbf{x}_i\right)\right) \tag{4}$$

where the multi-variate kernel $\mathcal{K}$ is obtained as the product of univariate ones

$$\mathcal{K}\left(\mathbf{x}\right) = \prod_{d=1}^{D}k\left(x_d\right) \tag{5}$$

with D the space dimension. In the Gaussian case, $\Sigma = HH^T$ is the *covariance matrix* of the multi-variate kernel.

Concerning the clustering method, in order to keep the most general setting, we make use of unsupervised clustering, through the weighted Mean-Shift algorithm [2]. Mean-shift is a kernel-based, non-parametric and unsupervised clustering method, that finds local maxima of the kernel density by gradient ascent, starting from each sample point, and assigns to the same cluster all paths that converge to the same peak; therefore, it simultaneously finds the number and location of modes, and assigns the sample points to each cluster as well.

By restricting the attention to isotropic kernels ($H = hI$), the density can be locally optimized by computing the weighted mean-shift vector

$$\mathbf{m}_h(\mathbf{x}) = \left[\frac{\sum_{i=1}^{n} w_i g\left(\left\| \frac{\mathbf{x}-\mathbf{x}_i}{h} \right\|^2 \right) \mathbf{x}_i}{\sum_{i=1}^{n} w_i g\left(\left\| \frac{\mathbf{x}-\mathbf{x}_i}{h} \right\|^2 \right)} - \mathbf{x} \right] \tag{6}$$

where $g = -k'$ its first derivative of the kernel, and afterwards updating the position $\mathbf{x} \rightarrow \mathbf{x} + \mathbf{m}_h$. The iteration is stopped when the update vector becomes smaller than a given threshold: $\|\mathbf{m}_h\| < \epsilon$.

Chosing the correct bandwidth h can be critical, in order to ensure that the correct number of particle clusters will be found. For our purposes, we simply choose h proportional to the minimum distance between detectable targets in state-space (resolution of the detector), which is of course application-dependent: for example, if the detected targets are small objects, the minimum distance will be smaller than for people detection.

During mean-shift clustering, it still may happen that small, spurious clusters of a few sample points are detected. These clusters are stationary points (where the mean-shift gradient is zero) but usually located on non-maxima, such as saddle points. Therefore, they are removed by a simple procedure: if a cluster center, located on a local maximum, is perturbed by a small amount, and the mean-shift algorithm is run again from this location, then it will converge again to the same point. Otherwise, the cluster center must be located on a saddle point.

3 Redundant multi-camera setup: sampling from the view-volume intersection

In a multi-camera context, we need first to initialize the particle set with a uniform distribution in 3D space. This requires defining the *sampling volume* for this distribution, in particular concerning the positional degrees of freedom (x, y, z translation).

In general, we consider here *redundant* multi-camera settings (Fig. 1), as opposed to *complimentary* ones. In a redundant configuration, the fields of view overlap to a large extent, so that the object can simultaneously be seen from all cameras, at any pose. This has the advantage of a more informative measurement set, which allows 3D tracking of complex objects. By contrast, a complimentary setup consists of almost non-overlapping camera views, where the object to be tracked can be completely seen only by one camera at a time.

In particular, when dealing with a redundant configuration, we need to sample hypotheses uniformly from the subset of state-space configurations that are visible from all cameras. This requires computing the joint *viewing volume* of m cameras. For this purpose, each camera provides 6 *clipping planes*, which overall define a truncated pyramid: 4 lateral planes defined by the 4 image sides, and the focal length, while the two frontal planes define the minimum and maximum depth of detectable objects.

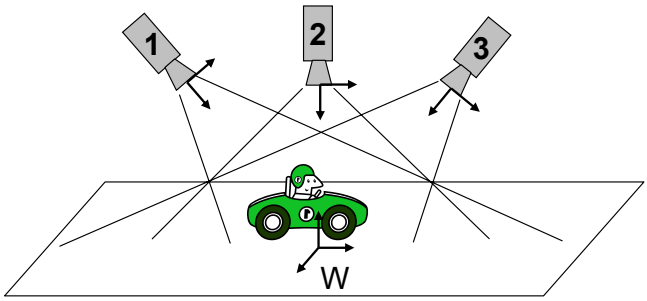

Fig. 1. Redundant, multi-camera configuration for object tracking.

These planes can be expressed (in camera-centered or world-centered coordinates) by means of 3 points in space. For example, the left clipping plane contains the upper-left and lower-left corners of the image, plus the camera center. In camera coordinates, then we have

$$^{c}\mathbf{x}_{u,l} = (-\frac{r_x}{2}, -\frac{r_y}{2}, f) \tag{7}$$

$$^{c}\mathbf{x}_{l,l} = (-\frac{r_x}{2}, \frac{r_y}{2}, f) \tag{8}$$

$$^{c}\mathbf{c} = (0, 0, 0)$$

These points can be transformed to world coordinates, by applying the respective camera transformation matrix $T_{W,c}$

$$^{W}\mathbf{x} = T_{W,c} \cdot ^{c}\mathbf{x} \tag{9}$$

Therefore, the left clipping plane of camera c, $\pi_{c,1}$ is given in homogeneous coordinates by the null-space of the (3×4) matrix [6] (dropping the reference frame W)

$$\pi = null\left(\begin{bmatrix} \mathbf{x}_{u,l}^T \\ \mathbf{x}_{l,l}^T \\ \mathbf{c}^T \end{bmatrix}\right) \tag{10}$$

so that all visible points from camera c must lie in the half-space defined by

$$\pi_{c,1}^T \mathbf{x} \leq 0 \tag{11}$$

where the sign of π can be chosen, for example, in order to make sure that the image center $(0, 0, f)$ (expressed in world coordinates) is contained in the half-space. The same procedure can be applied to the other clipping planes in a similar way.

If we denote by $\pi_{c,i}, i = 1, ..., 6$ the world-related planes of camera c, its viewing polyhedron is defined by the homogeneous inequalities

$$A_c\mathbf{x} \leq \mathbf{0};\ A_c \equiv \begin{bmatrix} \pi_{c,1}^T \\ \dots \\ \pi_{c,6}^T \end{bmatrix} \tag{12}$$

and finally, the overall intersection is given by the convex polyhedron, defined by

$$A\mathbf{x} \leq \mathbf{0};\ A \equiv \begin{bmatrix} A_1 \\ \dots \\ A_C \end{bmatrix} \tag{13}$$

This equation could be used in principle to directly select a uniform sample of visible points inside it. For this purpose, most popular methods in the literature refer to Markov-Chain Monte-Carlo (MCMC) strategies, starting from the well-known work [5]. However, due to its computational complexity and the presence of several parameters in the algorithm, in the present work we propose a simpler approach, that consists in uniformly sampling from the 3D *bounding box* of the polyhedron, and discarding all samples which do not satisfy (13). This will produce uniformly distributed points, at the price of discarding many samples, and therefore requiring a longer (and less predictable) time before reaching the desired number of valid points.

In order to compute the bounding box of the polyhedron, we also need to explicitly compute its vertices in 3D space, from the implicit formula (13). This is known as the *vertex enumeration* problem, and can be solved via the *primal-dual* method of [1].

A final note concerns the choice of the two main parameters for our algorithm (namely, the kernel size and the number of hypotheses), for which we employ the following criterion:

- The kernel width h determines the resolution of our detector, since two likelihood peaks closer than h lead to a single, detected mode in the mean-shift optimization.
- The number of hypotheses n determines the spatial density of the sample, which depends on the kernel size h: we need to make sure that at least one sample point falls into any sphere of radius h, in order for the mean-shift algorithm to work and not getting stuck into zero-density regions. Therefore, if $V_S(h)$ is the volume of a sphere of radius h, and V_B is the volume of the bounding box for sampling (which is larger than the polyhedron volume), we can choose $n = V_B/V_S$.

4 Applications and experimental results

In Fig. 2, we show an example result of the sampling procedure, applied to a 3-camera configuration. The three viewing volumes (indicated with different colors) intersect in the central polyhedron, which is filled by uniformly distributed points. Its bounding box is also shown in black.

As a first experiment, we run the proposed system on a simulated scenario: a set of randomly chosen targets provide "virtual measurements", by generating for each target o a measurement z_o around the true state $\bar{s}_o$, plus Gaussian noise v_o

$$z_o = \bar{s}_o + v_o \tag{14}$$

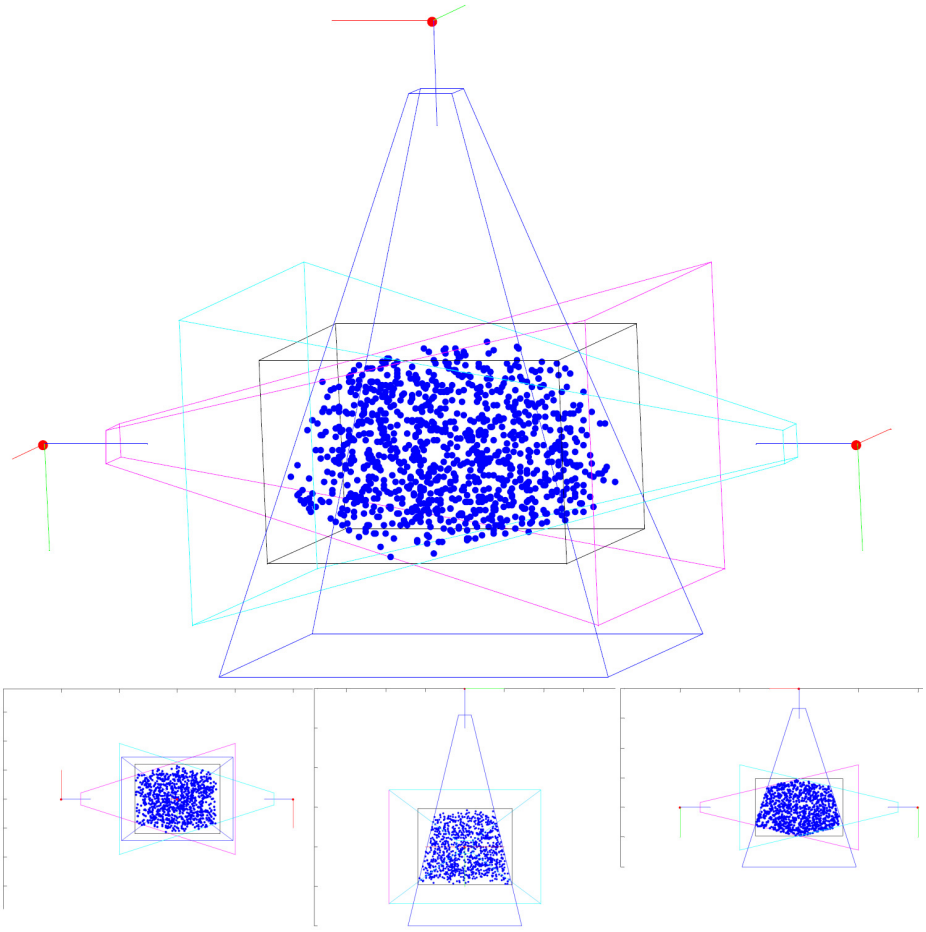

Fig. 2. Uniform sample from the joint viewing volume of a 3-camera configuration.

with the same covariance matrix V for all targets. This model corresponds to a Mixture of Gaussians likelihood

$$P\left(z|\, s\right) \propto \sum_{o} exp(-\frac{1}{2}\left(z_o - s\right)^T V^{-1}\left(z_o - s\right)) \tag{15}$$

for any state hypothesis s within the joint volume. The state here is represented by 3D position, $s = (x, y, z)$.

Four targets are selected at random within the viewing volume. In this example, all targets are separated in space by more than $100mm$, and the kernel size is $h = 30$, so that the detector has no difficulties in distinguishing them. A set of $n = 2000$ sample points is drawn within the volume, and their likelihood values

are computed. After performing mean-shift clustering, the detected modes are shown with different colors on the right side of Fig. 3.

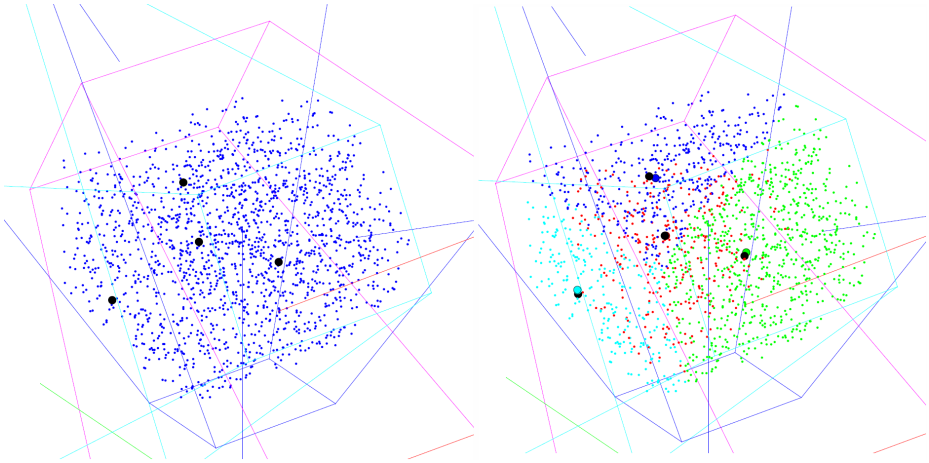

Fig. 3. Result of the simulated experiment, with 4 targets and a Gaussian likelihood on each target. Left: true targets (black dots) and the uniform Monte-Carlo set. Right: result of weighted mean-shift clustering, and respective cluster centers (the red cluster center is not visible, because almost coincident with the true target position).

As it can be seen, the detected modes are in the correct number, and their location is in a good accordance with the real target positions.

Subsequently, we tested the system on real camera images. As image likelihood, we compute the Bhattacharyya distance between color histograms, often used for object tracking [4][8]

$$B\left(q, p\left(s\right)\right) = \left[1 - \sum_{b} \sqrt{q_b p_b}\right]^{\frac{1}{2}} \tag{16}$$

where q is a reference histogram, collected from an image of the object, and $p(s)$ is the observed histogram, from image pixels underlying the projected object area, under pose hypothesis s (Fig. 4, left side). The sum is performed over $(D \times D)$ histogram bins ($D = 10$ is a common choice in Hue-Saturation space).

On a multi-camera setup, by assuming independence of the camera measurements, the corresponding likelihood is

$$P\left(z|s\right) \propto \prod_{c} \exp\left(-\frac{B^2\left(q, p_c\left(s\right)\right)}{2\sigma^2}\right) \tag{17}$$

where $p_c(s)$ is the image histogram at pose s, projected on camera c, and σ^2 is the measurement noise covariance (the same for all cameras and targets).

In Fig. 4, we can see the detection result for 4 real targets. The object model is given by a yellow sphere of radius $65mm$, and the reference histogram is collected from a single image of the object. On the left side of the picture, we can see the 3 camera images with superimposed projections of the estimated target locations; on the right side, the 3D positions of the detected targets in the common viewing volume, together with the particle clusters after mean-shift optimization, are shown.

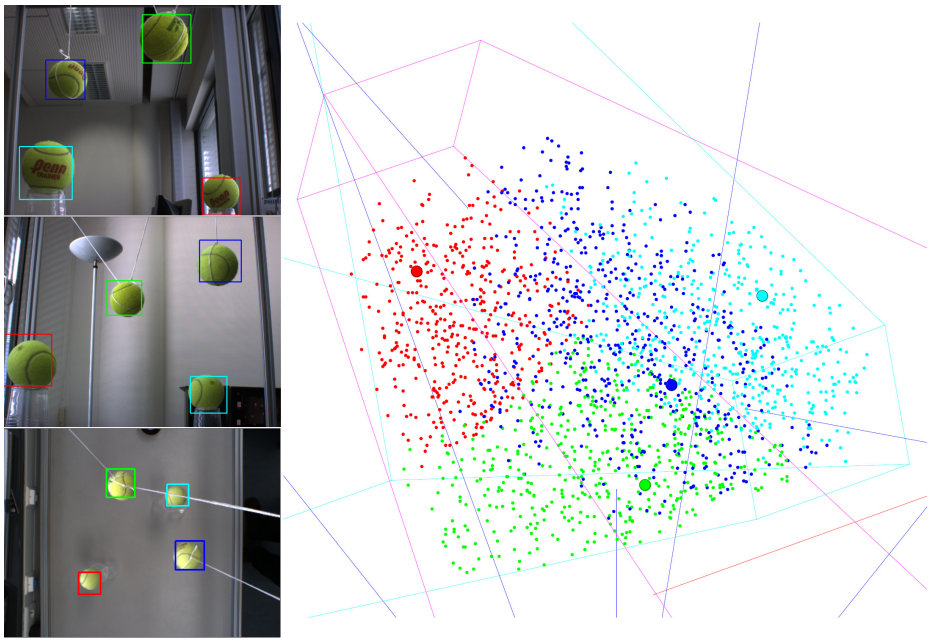

Fig. 4. Left: Camera images, with re-projection of the detected targets. Right: detected targets in 3D space, and particle clusters after mean-shift optimization.

5 Conclusion and future work

We presented a Monte-Carlo methodology for generic multi-camera, multi-target detection. The proposed method can be applied to a variety of likelihood functions in computer vision, as well as to generic, calibrated camera setups.

One limitation of the proposed system is the number of targets that can simultaneously be detected, still limited to a few units: a maximum of 7-8 targets have successfully been detected with the simulated experiment of Sec. 4 which, as explained at the end of Sec. 3, depends on the spatial resolution desired (i.e. the kernel bandwidth h).

A possible improvement of the system may use an adaptive version of mean-shift algorithm [3], where the bandwidth parameter is selected and modified according to the data points, in order to give the best clustering results.

6 Acknowledgements

This work is partly supported by the German Research Council (DFG), under the excellence initiative cluster *CoTeSys - Cognition for Technical Systems* [1] within the project *ITrackU (Image-based Tracking and Understanding).*

References

1. David Bremner, Komei Fukuda, and Ambros Marzetta. Primal-dual methods for vertex and facet enumeration (preliminary version). In *SCG '97: Proceedings of the thirteenth annual symposium on Computational geometry*, pages 49–56, New York, NY, USA, 1997. ACM.
2. Dorin Comaniciu and Peter Meer. Mean shift: A robust approach toward feature space analysis. *IEEE Trans. Pattern Anal. Mach. Intell.*, 24(5):603–619, 2002.
3. Dorin Comaniciu, Visvanathan Ramesh, and Peter Meer. The variable bandwidth mean shift and data-driven scale selection. *Computer Vision, IEEE International Conference on*, 1:438, 2001.
4. Dorin Comaniciu, Visvanathan Ramesh, and Peter Meer. Kernel-based object tracking. *IEEE Trans. Pattern Anal. Mach. Intell.*, 25(5):564–575, 2003.
5. Martin Dyer, Alan Frieze, and Ravi Kannan. A random polynomial-time algorithm for approximating the volume of convex bodies. *J. ACM*, 38(1):1–17, 1991.
6. R. I. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision.* Cambridge University Press, ISBN: 0521540518, second edition, 2004.
7. M. Isard and A. Blake. Condensation – conditional density propagation for visual tracking. *International Journal of Computer Vision (IJCV)*, 29(1):5–28, 1998.
8. Patrick Pérez, Carine Hue, Jaco Vermaak, and Michel Gangnet. Color-based probabilistic tracking. In *ECCV '02: Proceedings of the 7th European Conference on Computer Vision-Part I*, pages 661–675, London, UK, 2002. Springer-Verlag.
9. Gulshan Singh and Kalyanmoy Deb, Dr. Comparison of multi-modal optimization algorithms based on evolutionary algorithms. In *GECCO '06: Proceedings of the 8th annual conference on Genetic and evolutionary computation*, pages 1305–1312, New York, NY, USA, 2006. ACM.

[1] http://www.cotesys.org