DEFORMABLE MULTIPLE-KERNEL BASED HUMAN TRACKING USING A MOVING CAMERA

Li Hou[※], *Wanggen Wan*
College of Communication and Information Engineering,
Shanghai University,
Shanghai, 200444, China
{houli, wanwg}@shu.edu.cn

Kuan-Hui Lee, Jenq-Neng Hwang
Dept. Electrical Engineering,
Box 352500
University of Washington,
Seattle, WA, 98195, USA
{ykhlee, hwang}@uw.edu

Greg Okopal, *James Pitton*
Applied Physics Laboratory,
Box 355640
University of Washington,
Seattle, WA, 98195, USA
{okopal, pitton}@apl.uw.edu

ABSTRACT

In this paper, we propose an innovative human tracking algorithm, which efficiently integrates the deformable part model (DPM) into the multiple-kernel based tracking using a moving camera. By representing each part model of a DPM detected human as a kernel, the proposed algorithm iteratively mean-shift the kernels (i.e., part models) based on color appearance and histogram of gradient (HOG) features. More specifically, the color appearance features, in terms of kernel histogram, are used for tracking each body part from one frame to the next, the deformation cost provided by DPM detector is further used to constrain the movement of each body kernel based on the HOG features. The proposed deformable multiple-kernel (DMK) tracking algorithm takes advantage of not only low computation owing to the kernel-based tracking, but also robustness of the DPM detector. Experimental results have shown the favorable performance of the proposed algorithm, which can successfully track human using a moving camera more accurately under different scenarios.

Index Terms—human tracking, kernel-based tracking, deformable part model.

1. INTRODUCTION

Nowadays, the development of intelligent surveillance system has attracted significant attention. Human tracking is one of the major topics in intelligent video surveillance systems. By tracking human in the videos, it is possible to collect their trajectories for high level analytics and applications, for example, human counting, people flow estimation, criminal tracking, and so on.

Human tracking can be regarded as a specific category of video object tracking, which has been extensively developed and discussed [1], [2]. According to their tracking schemes, they can be roughly divided into two categories:

1) *Feature-based*: Most traditional methods utilize Kalman filter to express a target as a point in the frame, and the previous target state is used to make the association between targets and the point. Among the tracking algorithms, kernel-based object tracking are popular because of its fast convergence speed and relatively low computation. The idea of the kernel-based tracking is to minimize the difference between the target and the candidate appearance models constructed by spatially weighting the object with a kernel function in the color histogram calculations [3], [4]. Based on the kernel-based tracking framework, many improved methods have been proposed for human tracking [5]–[8]. In [9], the authors generalize the *constrained multiple-kernel* (CMK) tracking by adaptively adjusting kernels' widths and weightings according to their similarity, so as to improve the reliability in case of occlusion. However, the performance of the kernel-based tracking may become degraded due to the fact that fast changing color information during the tracking with moving cameras.

2) *Detection-based*: Recently, many approaches track objects based on object detection because of the robustness and effectiveness of detectors [10]–[13]. One of the most popular detectors is the *deformable part model* (DPM) [12], which uses a root model and several part models to describe different partitions of an object. The part models are spatially associated with the root model according to the predefined geometrical configuration, so as to precisely depict the object. By applying a detector to each frame of video sequence, the tracking scheme becomes a task to associate the detected objects with each other frame-by-frame. Such *tracking-by-detection* scheme is widely used in tracking human either in a static camera or a moving camera [14]–[18]. Instead of performing detected object association individually frame-by-frame, many approaches [19]–[21] have been proposed to formulate the association of detected object problem as a minimum-cost flow network problem. These methods globally optimize the trajectories of all objects, instead of locally optimizing for each object. Although these detection-based methods have good performance and show the robustness and effectiveness, the high computational cost due to pyramid scanning for different scales during the detection is always considered as a serious drawback of the detection-based methods.

In this paper, we propose an innovative deformable multiple-kernel (DMK) tracking algorithm, which efficiently combines the multiple-kernel based tracking and the object

※ Li Hou is also associated with School of Information Engineering, Huangshan University, Huangshan, 245041, China

detector, to overcome the issues mentioned above. More specifically, we integrate the DPM into the constrained multiple-kernel tracking, by regarding each part model as a kernel. The deformation costs of the part models are considered as the constraints to bind the kernels with each other into the most appropriate configuration. Given a detected object to be tracked, the proposed algorithm first mean-shift the kernels, which correspond to various parts of the DPM model, based on spatially weighted color histogram features, to the new locations in the next frame. These parts are then mean-shifted again based on HOG features of the current frame with respect to the corresponding part models which are pre-trained by the DPM detector. Instead of constraining kernels based on some pre-defined geometrical relationship as used in the CMK tracking [9], the proposed algorithm utilizes the deformation cost, which is statistically inferred from the pre-trained DPM detector, to restrict the movement of each kernel (i.e., part model). These two steps are alternatively operated to accurately locate the human target in each frame.

The rest of the paper is organized as follows. In Section 2, the technical overview of the algorithms adopted in our proposed DMK scheme, including mean-shift tracking and DPM, is presented. Section 3 depicts the details of the proposed DMK tracking framework and the integration of the DPM reconfiguration into the CMK tracking. The experimental results are shown in Section 4, followed by the conclusion in Section 5.

2. ADOPTED ALGORITHMS

2.1. Mean-Shift Tracking

The idea of mean-shift tracking [3] is to iteratively compute the closest candidate whose feature sample distribution is most similar to the target's ones. In the candidate model, a feature sample corresponding either to the color or texture appearance at location $\mathbf{x}$ has its associated metric weight $w(\mathbf{x})$, which defines the measurement of similarity between the feature sample and the target model. Generally, the mean-shift tracking method moves the object centroid by the mean-shift vector $\delta\mathbf{x}$. In other words, given the initial object location $\mathbf{x}_0$, the new object location $\mathbf{x}_1$ can be calculated by $\mathbf{x}_1 = \mathbf{x}_0 + \delta\mathbf{x}$, where the mean-shift vector $\delta\mathbf{x}$ is:

$$\delta\mathbf{x} = \frac{\sum_{i=1}^{N} k\left(\mathbf{x}_i - \mathbf{x}_0\right) w(\mathbf{x}_i)\left(\mathbf{x}_i - \mathbf{x}_0\right)}{\sum_{i=1}^{N} k\left(\mathbf{x}_i - \mathbf{x}_0\right) w(\mathbf{x}_i)}, \ \mathbf{x} \in \mathbb{R}^2 . \qquad (1)$$

Note that the denominator is used for normalization; $\{\mathbf{x}_i\}_{i=1\ldots N}$ is the pixel locations of the candidate model, and $k(\bullet)$ is a symmetric kernel with bandwidth h'.

2.2. Deformable Part Model (DPM)

The DPM [12] represents an object by a so-called star model, which comprises a coarse root model and several part models (or filters). The root model approximately denotes the entire object, while the part models denote smaller parts of the object with higher resolution. In DPM, a model for an object with M parts can be defined by an $(M + 2)$-tuple: $(F_0, P_1, \ldots, P_M, b)$, where F_0 is a root filter, P_j is a model for the j^{th} part, and b is a real-valued bias term. Each part model P_j is defined by a 3-tuple: (F_j, v_j, d_j), where F_j is a filter for the j^{th} part, v_j defines a 2-D "anchor" position for the j^{th} part relative to the root position, and d_j is a 4-D vector which contains coefficients of a quadratic function. The filters are expressed in a form of concatenated vectors, i.e., an $m_j \times n_j$ filter is represented as a 1-D vector with dimension of $(m_j \times n_j)$. The quadratic function defines a deformation cost for each possible offset of the part relative to the anchor position. If a part moves away from the centroid of the filter, the deformation cost becomes larger.

To detect a specific object in an image, the image is first down-sampled and up-sampled like a pyramid, and then the matching score of each object hypothesis ranged by a sliding window is computed to decide how likely the object is. The matching score is defined by:

$$\text{score}(\mathbf{x}_0, \ldots, \mathbf{x}_{n_p}, l) = \sum_{j=0}^{M} F_j \cdot \phi(\mathbf{x}_j, l) - \sum_{j=0}^{M} d_j \cdot \phi_d(d\mathbf{x}_j) + b , \quad (2)$$

where $\phi(\mathbf{x}_j, l)$ is the j^{th} concatenated HOG vectors at the l^{th} level of the pyramid, $\phi_d(d\mathbf{x}) = (dx, dy, dx^2, dy^2)$, is the $\phi_d(d\mathbf{x}) \in \mathbb{R}^4$ deformation feature, $d\mathbf{x}_j = \mathbf{x}_j - (2\mathbf{x}_0 + v_j)$, $d\mathbf{x}_j \in \mathbb{R}^2$ is the displacement of the j^{th} part relative to its anchor, and the bias term b is used to determine which viewing perspective component of the model is used.

3. DEFORMABLE MULTIPLE-KERNEL TRACKING

Inspired by the CMK tracking [9], which describes an object by multiple kernels that are bound together by several proper constraints, in our DMK tracking we regard each part model as a kernel and use deformation costs to restrict the movements of the kernels during the tracking. Therefore, the proposed algorithm iteratively mean-shift the kernels based on weighted color histogram and HOG, so as to take advantage of not only low computation owing to the kernel-based tracking but also the robustness of the DPM detector.

3.1. Tracking Scheme

Fig. 1 shows the flowchart of our proposed DMK scheme. Given an object of interest to be tracked, either by detectors or users' manual identifications, we first define the whole object as the root kernel. From the size (in terms of pixel) of the root kernel ($h_0 \times w_0$), we choose the viewing perspective component of the model whose (m_0 / n_0) is closest to (h_0 / w_0). Then, the part kernels are placed according to the "anchor" position relative to the root kernel. Each part kernel's size ($h_j \times w_j$) should be proportional to the size of the root kernel ($h_0 \times w_0$), such that the aspect ratio of the part

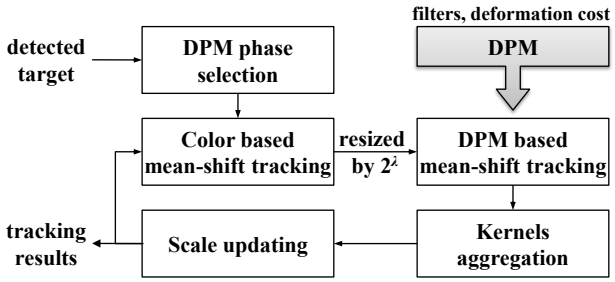


Fig. 1. Tracking scheme in each frame for deformable multiple-kernel (DMK) tracking.

kernel size to the root kernel size is equal to that of the part filter size to the root filter size, i.e.,

$$h_j / h_0 = m_j / m_0 \text{ , and } w_j / w_0 = n_j / n_0 . \quad (3)$$

Once the kernels are determined, the proposed algorithm searches the local maximum of both color and HOG similarity. First, we compute the color based (i.e., spatially weighted color histogram) mean-shift vector $\delta \mathbf{x}_j^{color}$ for the j^{th} kernel, based on Eq. (1), and then shift this kernel by $\delta \mathbf{x}_j^{color}$. This step is iteratively processed until the maximum iterations T^{color} is reached. Second, we compute the DPM based mean-shift vector $\delta \mathbf{x}_j^{dpm}$, and then shift the j^{th} kernel by $\delta \mathbf{x}_j^{dpm}$, iteratively until the maximum iterations T^{dpm} is reached.

Since background subtraction cannot be applied in the moving camera scenarios, no explicit segmentation masks can be created, we thus create an ellipse mask for each kernel when calculating either the weighted color or HOG histograms. When calculating $\delta \mathbf{x}_j^{color}$, we use the K-L distance as the measurement of the similarity, i.e., $w(\mathbf{x}_i)$ introduced in Eq. (1). Also roof kernel [8] for spatial weighting is adopted by $k(\bullet)$. Furthermore, $\delta \mathbf{x}_j^{color}$ is weighted by the similarity of the j^{th} kernel $simi_j(\mathbf{x}_j)$. As for $\delta \mathbf{x}_j^{dpm}$, the measurement of the similarity at $\{\mathbf{x}_i\}_{i = 1...N}$ is based on calculating matching score between the HOG within an ($h_j \times w_j$) block centered at each $\mathbf{x}_i$ and the corresponding DPM filter. Since the HOG models are trained in a specific scale level, the frame should be resized to the same scale level, so as to well match the HOG. Thus, the proposed algorithm determines the number of scale level λ by:

$$\lambda = \min\left\{\log_2\left(w_0 / \left(n_0 \times s_{cell}\right)\right), \log_2\left(h_0 / \left(m_0 \times s_{cell}\right)\right)\right\}, \quad (4)$$

where s_{cell} is the cell size pre-defined in the DPM. The size of the frame is converted (down-sampled or up-sampled depending on the sign of λ) by a factor 2^{λ}, so that the HOG extracted from the target can be well matched with that of DPM. Therefore, to compute the mean-shift vector, we consider the matching score at each $\mathbf{x}_i$ as $w(\mathbf{x}_i)$ in Eq. (1). For each (say the j^{th}) kernel, the matching score at each $\mathbf{x}_i$ location can be defined as

$$w_j(\mathbf{x}_i) = F_j^* \cdot \phi^*(\mathbf{x}_i, \lambda) - d_j^* \cdot \phi_d^*(d\mathbf{x}_{i,j}) , \quad (5)$$

where $(\cdot)^*$ represents the normalized vector, F_j and d_j are provided by DPM, and $d\mathbf{x}_{i,j} = \mathbf{x}_{i,j} - (2\mathbf{x}_0 + v_j)$ is the displacement of the j^{th} part kernel relative to the root kernel.

As shown in Eq. (5), if the second term is larger than the first term, $w_j(\mathbf{x}_{i,j})$ is negative $\delta \mathbf{x}_j^{dpm}$ and cannot be correctly computed. To avoid this situation, we normalize $\delta \mathbf{x}_j^{dpm}$ by the summation of the $|w_j(\mathbf{x}_{i,j})|$, i.e.,

$$\delta \mathbf{x}_j^{dpm} = \frac{\sum_{i=1}^{N_j} k\left(\mathbf{x}_{i,j} - \mathbf{x}_0\right) w_j(\mathbf{x}_{i,j}) \left(\mathbf{x}_{i,j} - \mathbf{x}_0\right)}{\sum_{i=1}^{N_j} k\left(\mathbf{x}_{i,j} - \mathbf{x}_0\right) \left| w_j(\mathbf{x}_{i,j}) \right|} . \quad (6)$$

Again, $\delta \mathbf{x}_j^{dpm}$ is further weighted by the color similarity of the j^{th} kernel $simi_j(\mathbf{x}_j)$. The reason we use color similarity here is that color is more distinctive to tell one object from another, especially when occlusion occurs.

3.2. Kernels Aggregation

After both color based and DPM based mean-shift, multiple part kernels are aggregated to determine the newly tracked position of an object, i.e., the center of the target:

$$c_{final} = \alpha \cdot c_{root} + (1-\alpha) \cdot c_{part} , \; c_{final} \in \mathbb{R}^2 , \quad (7)$$

where $c_{root} = \mathbf{x}_0 = (x_0, y_0)$ is the center of the root kernel, and c_{part} is defined by

$$c_{part} = c_{prev} + \left(\sum_{j=1}^{M} simi_j(\mathbf{x}_j) \cdot \left(c_j - c_{prev}\right)\right) \Big/ \left(\sum_{j=1}^{M} simi_j(\mathbf{x}_j)\right), \quad (8)$$

where previous target's center c_{prev}, and c_j is the relative object's center derived from $\mathbf{x}_j$, i.e., $c_j = (\mathbf{x}_j - v_j)/2$, v_j denotes the anchor of the j-th part kernel. The weight $simi_j(\mathbf{x}_j)$ reflects how well the candidate features match the target features; a higher $simi_j(\mathbf{x}_j)$ refers to higher confidence of the kernel. $\alpha \in [0,1]$ is a parameter to balance the importance between the root kernel and part kernels. We choose $\alpha = 0.6$ in this paper.

3.3. Scale Issue

The proposed algorithm updates the scale (size) of the target if the target is moving toward or away from the camera. We adopt the scale updating mechanism in [9], which utilizes derivative of the density estimator $f(h')$ with respect to the kernel bandwidth h'. Hence, the proposed algorithm applies the change of the scale $\triangle s = -\beta \cdot \nabla f(h')$, i.e.,

$$s(t) = (1 + \Delta s) \cdot s(t-1) , \quad (9)$$

where β is the step size and has an empirical value of $\beta = 9000$ in this paper.

4. EXPERIMENTAL RESULTS

Our simulation scenarios are mainly in tracking a specific person in a moving platform (camera), where traditional background subtraction is not suitable for the foreground extraction. The experiment settings for the case of recorded videos and the real-time moving platform are separately described in this section. All the videos associated with the simulations reported in this paper can be viewed from our website[1].

[1] website: http://allison.ee.washington.edu/kuanhuilee/dmkt

TABLE I Average Error (Pixel)

Dataset	Method [3]	Method [9]	Proposed
BAHNHOF_1	64.83	59.50	27.50
BAHNHOF_2	60.24	50.41	36.28
JELMOLI_1	67.69	62.35	37.73
JELMOLI_2	78.55	69.08	45.70
UWMDR 1	136.2	117.3	90.48
UWMDR 2	115.8	153.6	72.65
UWCP_1	424.2	129.8	32.99

4.1. Tracking in Recorded Videos

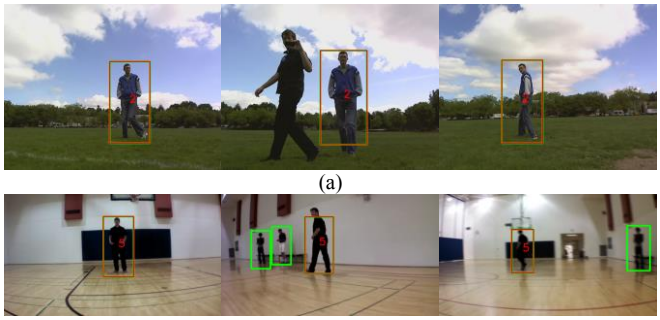
To demonstrate the performance, we test several video sequences in the ETH Mobile Scene (ETHMS) dataset [15] and our own recorded videos. The targets are manually selected in the beginning and then DMK is applied to continuously track them. For the mean-shifting tracking, we choose $T_{color} = 5$ and $T_{dpm} = 3$. Fig. 2 shows the typical visual results of the DMK tracking, which tracks people well in case of moving cameras. To further evaluate the performance, we use the pixel error, which is defined as the average distance of the objects' corners (left, top, right, and bottom of the bounding box) between the simulation results and ground truth, provided by ETHMS and our manually labeling. To demonstrate the performance, we compare two kernel-based tracking methods, one is the single kernel mean-shift [3] and the other is the constrained 2-kernel mean-shift tracking [9]. Table I shows the average error of these competing algorithms on the tested datasets, where the results show the robustness and performance improvement of our proposed method. This is because both methods [3] and [9] search the target only based on the color similarity, but the proposed method further search the target based on HOG similarity, i.e., DPM similarity. In the case of UWCP_1, where the target is severely occluded by other pedestrians, the proposed method shows the effectiveness and the robustness of handling the occlusion issues.

4.2. Tracking in Real-Time Moving Platforms

The moving platforms used in our experiments are self-built mobile robot "Patsy" and AR Drone 2.0 [22], both operated under the Robot Operating System (ROS). The Pasty is equipped with a 1.6GHz CPU and WiFi connection. In order to achieve better real-time performance, we use another laptop computer with a 3.0GHz CPU to remotely compute the tracking algorithm. Additionally, we apply video stabilizer for the drone to stabilize the recorded vibrating video frames during the flight. Both systems using DMK tracking allow either the Patsy or AR Drone to continuously follow the tracked human. As for the parameters used in DMK tracking, we choose $T_{color} = T_{dpm} = 3$. Fig. 3 shows the visual results from the perspective views of the moving platforms. More demo videos are also available on our website[1].

(a) (b) (c) (d) (e)

Fig. 2. Visual tracking results of tracking specific people in the tested dataset: (a) BAHNHOF, (b) JELMOLI, (c) UWMDR_1, (d) UWMDR_2, (e), and (f) UWCP .



(a) (b)

Fig. 3. Visual tracking results from the perspective views of the moving platforms: (a) Patsy, and (b) AR drone 2.0.

5. CONCLUSION

This paper proposes an innovative DMK tracking algorithm, which efficiently integrates the DPM into the multiple-kernel tracking. Multiple kernels alternatively search the local optimal based on color and DPM information, and kernels are bound with each other owing to the deformation costs. The proposed algorithm takes advantage of not only low computation owing to the kernel-based tracking, but also robustness of the DPM detector, so as to successfully track objects more accurately.

6. REFERENCES

[1] A. Yilmaz, O. Javed, and M. Shah, "Object tracking: A survey," *ACM Comput. Surveys*, vol. 38, no. 4, 2006.

[2] D. Gerónimo, A. M. López, A. D. Sappa and T. Graf, "Survey of pedestrian detection for advanced driver assistance systems," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 32, no. 7, pp. 1239–1258, Jul. 2010.

[3] D. Comaniciu and P. Meer, "Mean shift: A Robust Approach toward Feature Space Analysis," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 24, no. 5, pp. 603–619, May 2002.

[4] D. Comaniciu, V. Ramesh, and P. Meer, "Kernel-based object tracking," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 25, no. 5, pp. 564–577, May 2003.

[5] A. Yilmaz, "Object tracking by asymmetric kernel mean shift with automatic scale and orientation selection," *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, pp. 1–6, Jun. 2007.

[6] B. Martinez, L. Ferraz, X. Binefa, and J. Diaz-Caro, "Multiple kernel two-step tracking," *Proc. IEEE Int'l. Conf. Image Processing*, pp. 2785–2788, 2006.

[7] Z. Fan, Y. Wu, and M. Yang, "Multiple collaborative kernel tracking," *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, pp. 502–509, Jun. 2005.

[8] G.D. Hager, M. Dewan, and C. V. Stewart, "Multiple kernel tracking with SSD," *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, vol. 1, pp. 790–797, 2004

[9] C.-T. Chu, J.-N. Hwang, H.-I Pai, and K.-M. Lan, "Tracking human under occlusion based on adaptive multiple kernels with projected gradients," *IEEE Trans. Multimedia.*, vol.5, no.7, pp. 1602–1615, Nov. 2013.

[10] N. Dalal and B. Triggs, "Histograms of Oriented Gradients for Human Detection," *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, vol. 1, pp. 886–893, 2005.

[11] B. Leibe, A. Leonardis, and B. Schiele, "Robust Object Detection with Interleaved Categorization and Segmentation," *Int'l J. Computer Vision*, vol. 77, no. 1–3, pp. 259–289, May 2008.

[12] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan, "Object Detection with Discriminatively Trained Part-Based Models," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 32, no.9, pp. 1627–1645, Sep. 2010.

[13] J. Wu, N. Liu, C. Geyer, and J. M. Rehg, "C4: A Real-Time Object Detection Framework," *IEEE Trans. Image Processing*, vol. 22, no.10 , pp. 4096–4106, Oct. 2013.

[14] B. Leibe, N. Cornelis, K. Cornelis, and L. VanGool, "Dynamic 3D scene analysis from a moving vehicle," *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, Jun. 2007.

[15] A. Ess, B. Leibe, K. Schindler, and L. VanGool, "Robust multiperson tracking from a mobile platform," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 31, no. 10, pp. 1831–1846, Oct. 2009.

[16] H. Cho, P.E Rybski, A. Bar-Hillel, and W. Zhang, "Real-time pedestrian detection with deformable part models," *Proc. IEEE Intelligent Vehicles Symp*, pp. 1035–1042, Jun. 2012.

[17] L. Zhang and L. van der Maaten, "Improving object tracking by adapting detectors," *Proc. IEEE Int'l Conf. Pattern Recognition*, Jun. 2014.

[18] H. Zhang, S. Cai and L. Quan, "Real-time object tracking with generalized part-based appearance model and structure-constrained motion model," *Proc. IEEE Int'l Conf. Pattern Recognition*, Jun. 2014.

[19] L. Zhang, Y. Li, and R. Nevatia, "Global data association for multi-object tracking using network flows," *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, Jun. 2008.

[20] H. Pirsiavash, D. Ramanan, and C. C. Fowlkes, "Globally-optimal greedy algorithms for tracking a variable number of objects," *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, Jun. 2011.

[21] J. Berclaz, F. Fleuret, E. Türetken, and P. Fua, "Multiple object tracking using k-shortest paths optimization," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 33, no.9, pp. 1806–1819, Sep. 2011.

[22] *AR Drone 2.0*, link: http://ardrone2.parrot.com/