SELF-ORGANIZED AND SCALABLE CAMERA NETWORKS FOR SYSTEMATIC HUMAN TRACKING ACROSS NONOVERLAPPING CAMERAS

Chun-Te Chu, Kuan-Hui Lee, Jenq-Neng Hwang
Department of Electrical Engineering,
Box 352500, University of Washington,
Seattle, WA 98195, U.S.A.
{ctchu, ykhlee, hwang}@uw.edu

ABSTRACT

We present a self-organized and scalable multiple-camera tracking system that tracks human across the cameras with nonoverlapping views. Given the GPS locations of uncalibrated cameras, the system automatically detects the existence of camera link within the camera network based on the routing information provided by Google Maps. The connected zones in any pair of directly-connected cameras are identified based on the feature matching between the camera's view and Google Street View. The camera link model is further estimated by an unsupervised learning scheme. Finally, multiple-camera tracking is performed. Thanks to the unsupervised pairwise learning and tracking in our system, the camera network is self-organized, and our proposed system is able to be scaled up efficiently when more cameras are added into the network.

Index Terms— self-organization, scalable, camera network, multiple-camera tracking, camera link model

1. INTRODUCTION

Tracking objects across multiple cameras has recently attracted a lot of interests in the video surveillance community. Due to the limited field of view (FOV) of a single camera, a target's information is no longer available once the target leaves the view of the camera. Hence, a surveillance system is required to have multiple networked cameras covering a wide range of area. One of the major challenges of tracking multiple people across uncalibrated cameras with nonoverlapping (disjoint) views is to re-identify the same people. Some researchers aim to come up with distinctive features of human [1][2][3], such as SIFT, SURF, covariance matrix, etc. The re-identification is done based on the assumption that these kinds of features are invariant under different cameras' views. However, due to different perspectives and illuminations, the human appearance varies dramatically under different cameras' views. On the contrary, we focus on solving the tracking problem based on systematically building the links between cameras [4]. If there exists a path allowing people traveling between two cameras without passing through any other camera, we call they are *directly-connected* cameras, and a link exists between them. The relationship between a particular pair of *entry/exit zones* in two directly-connected cameras can be characterized by a *camera link model*. The entry/exit zone is defined as the area that people tend to enter in or leave from within the camera's view. The camera link model enables us to utilize a particular feature, which may not be invariant, under different cameras [4]. For instance, due to different lighting conditions and camera color responses, the same object may appear in different colors under different views. The brightness transfer function (BTF) [5] is applied to compensate the deviation between different color models of the cameras. BTF is included in the model. After the camera link models are estimated between pairs of cameras, they can be utilized to compute the similarity between the people in different cameras and to track the objects across the cameras [4].

Given a camera network consisting of multiple cameras, two pieces of information are required before the camera link model estimation can be performed: (i) The system needs to identify which pairs of cameras have link models between them, i.e., which pairs are directly connected. Wrong links or redundant links deteriorate the tracking performance easily, due to the increased searching range resulting in reduced recall rate and increased false positives, not to mention the exponentially increased computational complexity. (ii) To our observation, the link actually only connects two entry/exit zones in a pair of directly-connected cameras; that is, if a person is traveling between two cameras, he/she will likely leave from one particular zone and enters into the other. Hence, the training data used in camera link model estimation (and the subsequent re-identification tracking) should only include the observations happening in these two specific zones in order to avoid too many outliers. Therefore, to identify which specific zones are linked together is another critical issue. In this paper, we propose a systematic method that performs the camera link identification by incorporating the information from Google Maps and Google Street View.

Fig. 1 shows the overview of our proposed system. First of all, the camera link identification, including link existence detection and connected zones identification, is performed based on the incorporation of Google Maps. After that, the system automatically estimate the camera link model based on the training data. Finally, the model is utilized for tracking objects across multiple cameras with nonoverlapping field of views.

Makris [6] exploited the statistical consistency of the training data in order to identify the link and build the link model. However, the presence of outliers was not considered, so the accuracy of the estimation drops significantly if the outliers exist [7]. In Gilbert's work [8], the links were learned based on an incremental scheme. In their block-based entry/exit zone formulation, the identification of the link required large amount of training data, which reduced the scalability of the camera network size. Javed [9] presented a multiple-camera tracking system which combined the temporal and color features. All the learning in this design was under human supervision. The link identification between cameras is assumed known in the beginning. In [10], a method was proposed to discover and remove the "weak link" which was defined as the redundant link between two

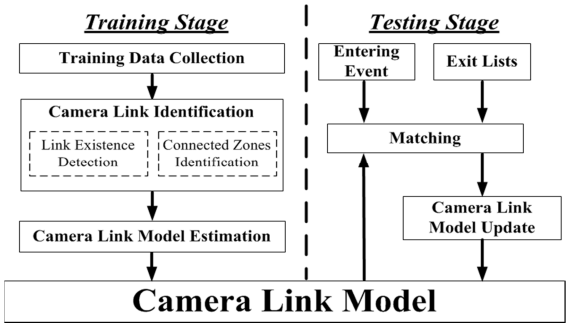

Figure 1. System Overview

cameras that are not directly connected; that is, all the paths between two cameras pass through at least one other camera. Several temporal criterions were designed for identifying the valid link. However, these temporal criterions suffered from the presence of outliers in the training data.

Our aim in this paper is to present a scalable and self-organized multiple-camera tracking system which automatically and unsupervisedly identifies the links between cameras, estimates the camera link model, and tracks objects across the cameras. The only prior information is the user specified GPS locations of the cameras. Specifically, in addition to the camera link model estimation in [4], the camera link identification is proposed and added in the system which makes it self-organized and enhances the scalability significantly.

This paper is organized as follows: The proposed camera link identification, including link existence detection and connected zones identification, is introduced in Section 2. Section 3 describes the estimation of the camera link model and tracking across multiple cameras. The experimental results are shown in Section 4, followed by the conclusion in Section 5.

2. CAMERA LINK IDENTIFICATION

The camera link model between each pair of directly-connected cameras has been shown to be effective in tracking human across multiple cameras [4][9][10]. Before the camera link model is estimated, the prior knowledge is to identify which pairs of cameras within the camera network should possess a camera link model. In this section we introduce how our system detects the existence of links given the GPS locations of the cameras. The camera link identification includes the following two modules: link existence detection and connected zones identification.

2.1. Link Existence Detection

Given the locations of the cameras, which are easily obtained when setting up the cameras in the environment, we are able to access the routing information provided by Google Maps. The routing information contains the possible routes between any two locations. If there exists one route that connects two cameras without passing by another camera, we recognize them as directly-connected, and there should be a link between them. If all the routes between two cameras pass by at least one other camera, we recognize the link should not exist between the two cameras. Fig. 2 shows an example of the routing associated with three cameras denoted by C_0, C_1, and C_2, whose locations are shown in Fig 2(a). To our observation, in practice people tend to follow the similar paths due to the presence of available pathway, obstruct, and shortest route, so it is reasonable to utilize the estimated paths from Google Maps as the routing information. Fig. 2(b), (c) and (d) show the shortest routes between each pair of cameras. Since the route between C_1 and C_2 passes C_0, the system only detect the links between C_0 and C_1 and C_0 and C_2.

2.2. Connected Zones Identification

There may be several entry/exit zones within a camera's view. The link between two directly-connected cameras actually only connects one zone each in these two cameras. So far, we can only know the existence of the link from the link existence detection without knowing the specific zones that are connected together. If we can know the connected zones, we only need to collect the training data, the exit and entry observations, from the associated zones so as to reduce large number of outliers in the training data during the estimation of camera link model and also obtain better accuracy when tracking the objects.

First of all, all the zones within each camera are detected in an unsupervised manner by using the Gaussian Mixture Model based on the collected entry/exit measurements [10]. Then, we match the camera's view with the panoramic images automatically retrieved from Google Street View to estimate the principal orientation of the camera. The scheme of the street view matching is described as follows: (i) given a GPS location, the system can access the images from Google Street View with different viewing angles θ, pitches φ, and foveation f. (ii) perform feature point matching between the images and camera's view. (iii) identify the image with the maximum number of the matched points, and the corresponding viewing angle θ offers the principal orientation of the camera. Fig. 3 shows an illustration of the scheme, where the camera's view highly matches one of the panoramic images. Moreover, the direction of the route is provided by Google Maps according to the GPS locations. Therefore, given the principal orientations, the direction of the route, and the detected entry/exit zones, we can estimate the two zones that are connected together.

3. CAMERA LINK MODEL ESTIMATION AND MULTIPLE-CAMERA TRACKING

3.1. Camera Link Model Estimation

After the camera links are identified, the training data, i.e., observations happening within the connected zones, is collected automatically through the single camera tracking module [11]. For each link between a pair of directly-connected cameras, denote the training data, two observation sets collected from a pair of entry/exit zones, as $\mathbf{X}$ and $\mathbf{Y}$ representing the exit and entry observations, respectively:

$$\mathbf{X} = [\mathbf{x}_1 \quad \cdots \quad \mathbf{x}_{N_1}],\ \mathbf{Y} = [\mathbf{y}_1 \quad \cdots \quad \mathbf{y}_{N_2}] \qquad (1)$$

where $\mathbf{x}_i$ and $\mathbf{y}_j$ are exit and entry observations, and N_1 and N_2 are the numbers of the observations. Each observation contains the exit or entry time stamp, color and texture information of the object. In order to build the camera link model between the cameras, the goal of the estimation process is to automatically identify the correspondences between two sets, i.e., to find the $(N_1+1)\times(N_2+1)$ correspondence matrix $\mathbf{P}$. The entry P_{ij} in $\mathbf{P}$ is set to 1 if $\mathbf{x}_i$ corresponds to $\mathbf{y}_j$; otherwise, it is set to 0. The $(N_1+1)th$ row and the $(N_2+1)th$ column represent the outlier entries, i.e., an exit observation from one camera never enters in the other, or an entry observation in one camera is not from the other. The problem

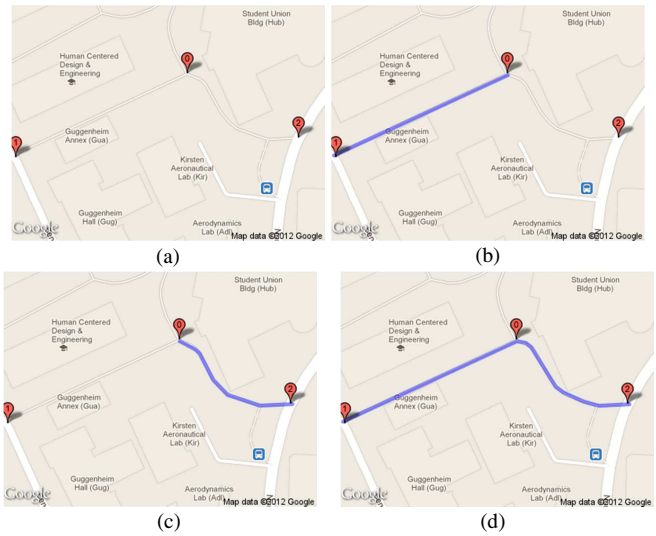

Figure 2. An example of the routing associated with camera 0, 1, and 2. (a) The locations of three cameras. The shortest route between (b) camera 0 and 1, (c) camera 0 and 2, (c) camera 1 and 2.

can be written as a constrained minimization integer programming problem:

$$\hat{\mathbf{P}} = \arg\min_{\mathbf{P}} J(\mathbf{P}) \tag{2}$$

$$\text{s.t. } P_{ij} \in \{0,1\} \quad \forall\, i \le N_1 + 1, j \le N_2 + 1 \tag{3}$$

$$\sum_{i=1}^{N_1+1} P_{ij} = 1 \;\; \forall\, j \le N_2, \qquad \sum_{j=1}^{N_2+1} P_{ij} = 1 \;\; \forall\, i \le N_1 \tag{4}$$

where $J(\cdot)$ is the objective function to be minimized. The constraint equations (3) and (4) enforce the one-to-one correspondence (except for the outliers). The objective function $J(\cdot)$ comprised several cost functions. Each cost function stands for the distance between a pair of exit and entry observations associated with one feature, e.g., time, color, texture, where the camera link model is applied before computing the distance. We adopt an innovative EM-like algorithm to sequentially solve the matrix **P** and the camera link model in each iteration. Deterministic annealing is also employed in our estimation process to obtain the optimal solution [4].

The information obtained from Google Maps not only provides the routes between two locations but also gives an estimation of the traveling time between them. It enables us to pose a good initial state of the matrix **P** before the estimation process starts. Denote the values of the traveling time estimated by Google Maps as s_i, $i = 1{\sim}K$. Note that K may be greater than one if there exist multiple alternative routes between two cameras. We can have

$$P_{ij} = \begin{cases} 0, & y_j^t - x_i^t \le 0, \;\; \forall i = 1{\sim}N_1, j = 1{\sim}N_2 \\ \frac{1}{K}\sum_{m=1}^{K} exp\left(-\frac{(y_j^t - x_i^t - s_m)^2}{2\sigma^2}\right), & otherwise \end{cases} \tag{5}$$

where y_j^t and x_i^t represent the time stamps in the observations $\mathbf{y}_j$ and $\mathbf{x}_i$. Since the traveling time is always positive if two cameras have nonoverlapping area, if the entry time stamp is smaller than the exit time stamp ($y_j^t - x_i^t \le 0$), it is not possible for them to be a matched pair, hence P_{ij} is set as 0. Otherwise, we assume it takes people roughly the estimated amount of time s_i to move from one camera to the other, so P_{ij} is set as the likelihood based on a parzen window built by s_i. By incorporating this information as prior knowledge to the estimation process, it enables the system to reach the convergence with fewer iterations than are required in [4].

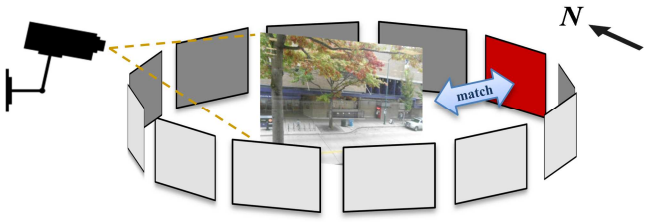

Figure 3. The estimation of principal orientation of the camera.

3.2. Tracking Objects Across Multiple Cameras

In the testing phase, each camera C_i, $i = 1{\sim}N_C$ maintains an exit list $\boldsymbol{L}_{i,k}$ for each entry/exit zone k. It consists of the observations of the people who have left the FOV from zone k within T_{max} seconds from now.

$$\boldsymbol{L}_{i,k} = \{O_{i,k}^1, O_{i,k}^2 \ldots O_{i,k}^{Nc_{i,k}}\} \tag{6}$$

Whenever a person enters a camera's view, the system finds the best match among the people in the exit lists corresponding to the linked zones of its directly-connected cameras. Based on the camera link model, the matching score between two objects is computed as the weighted sum of negative distances:

$$score = -\sum_{i=1}^{N_{feature}} \alpha_i \times feature\_dist_i \tag{7}$$

where α_i, which is obtained in the estimation stage, is the weight for the distance corresponding to the feature i. Note that the camera link model is applied to calculate the distances with respect to different features. If the highest score is higher than certain threshold, the label handoff is performed; otherwise, we will treat it as a new person within the camera network. The re-identification results are further used to update the camera link model. In the implementation, we consider transition time, color, and texture as our features [4]. We adopted the method in [11] for all the object detection and tracking within a camera.

4. EXPERIMENTAL RESULTS

In this section, we will first show the results of the connected zones estimation. After that, we present the tracking results of our self-organized scalable multiple-camera tracking system.

4.1. Connected Zone Identification

To obtain the information from Google Maps, we implement a user interface by using Google Maps APIs 3.0. In the street view matching, we adopt SIFT feature [12] as our point matching algorithm. We divide the viewing angle θ into 24 segments, i.e., $\theta = 15° \times k$, $k = 0, 1, 2, \ldots, 23$. For each angle, we retrieve a set of 9 images which are the combination of 3 different pitches φ and 3 different foveation f ($\varphi = -20, -10, 0$; $f = 80, 100, 120$). The camera's view is matched to this set of images, and the cumulated number of the matched points is used as the degree of matching for this angle. The image resolution we used is 640×480.

Fig. 4 shows one of our deployed cameras. Fig. 4(a) is the camera's view, and the entry/exit zones are marked as red ellipses. Four panoramic images with different θ from Google Street View are shown in Fig. 4(b). Fig. 4(c) shows the result of the SIFT feature matching, where the red dot block is the ground truth of the principle orientation obtained manually. One can see that the number of the matched points are relatively high from 105° to 120°, which is close to the ground truth. Since $\theta = 0°$ stands for

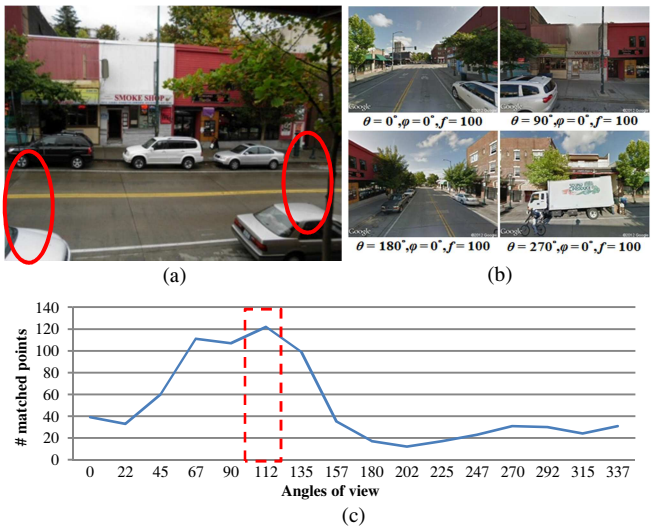

Figure 4. (a) Camera's view. Red ellipses are the entry/exit zones. (b) Four panoramic images from Google Street View. (c) Result of the SIFT feature matching. Red dot block is the ground truth of the camera orientation.

the orientation toward north, the principal orientation of the camera is estimated as toward East, and the left entry/exit zone is at North side of the view while the right one is at South side of the view. Fig. 5 shows the results from another deployed camera. Similarly, the principal orientation of the camera is estimated as toward South, and the left entry/exit zone is at East side of the view while the right one is at West side of the view. We tried 13 cameras, and all of their principal orientations can be determined well through the matching against Google Street View.

Given the route direction from Google Maps, we can then estimate that the link connects the left entry/exit zone of the view in Fig. 4 and the left entry/exit zone of the view in Fig. 5. Fig. 6 shows the connected zones.

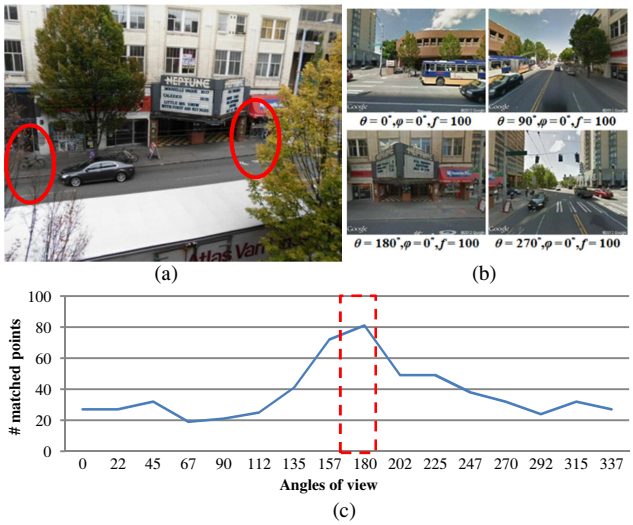

Figure 5. (a) Camera's view. Red ellipses are the entry/exit zones. (b) Four panoramic images from Google Street View. (c) Result of the SIFT feature matching. Red dot block is the ground truth of the camera orientation.

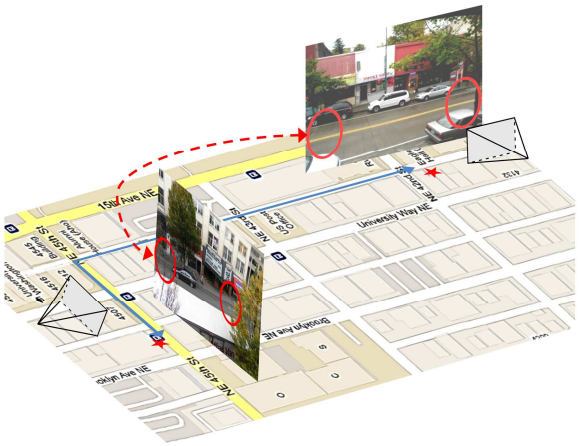

Figure 6. Connected zones identification. Red stars denote the camera locations. Red ellipses are the entry/exit zones. The connected zones, linked by red line, can be identified based on the principal orientations and the direction of the route (shown as the blue line).

4.2. Camera Link Model Estimation and Tracking

We set up four cameras around the department building, and the FOVs of the cameras are spatially disjointed. Cameras do not need any calibration in advance. The only prior knowledge is the GPS locations of the cameras which is accessible in real world practice. By incorporating the estimated traveling time from Google Maps (eq. (5)), the number of the required iteration in the camera link model estimation process drops about 11% compared to [4]. In our testing video, there are 188 people appearing in the deployed camera network, and our system achieves 76.9% re-identification accuracy defined as the fraction of the people being correctly labeled. Note that after the GPS locations are specified, all the estimation and tracking processes are fully automatic.

Since our system is based on the pairwise learning and tracking scheme, the system can be scaled up easily. Here we present a simple scenario to illustrate the scalability of the system. Assume there are N_C cameras, C_i $i = 1 \sim N_C$, already in the network, and we would like to add one camera C_{N_C+1} in the network. Providing the new camera's location, the system automatically identifies the links and the connected zones between C_{N_C+1} and the other cameras. After that, the camera link model estimation is performed pairwisely for those newly created links. By applying the models, tracking across multiple cameras is carried out within this new camera network. Following the similar manner, the camera network can be scaled up without human intervention.

5. CONCLUSION

We propose a scalable multiple-camera tracking system. By providing the GPS locations of uncalibrated cameras and incorporating with Google Maps and Google Street View, our system automatically identifies the camera links within the camera network, estimates the camera link models for pairwise zones, and performs multiple-camera tracking. The pairwise learning and tracking scheme enables the system to be self-organized and be scaled up efficiently.

12. REFERENCES

[1] M. Farenzena, L. Bazzani, A. Perina, V. Murino, and M. Cristani, "Person re-identification by symmetry-driven accumulation of local features," *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, pp. 2360-2367, 2010.

[2] S. Bak, E. Corvee, F. Bremond, and M. Thonnat, "Person re-identification using Haar-based and DCD-based signature," *IEEE Intl. Conf. on Advanced Video and Signal Based Surveillance*, 2010.

[3] M. Bauml and R. Stiefelhagen, "Evaluation of local features for person re-identification in image sequences," *IEEE Intl. Conf. on Advanced Video and Signal Based Surveillance*, Sep, 2011.

[4] C. Chu, J. Hwang, J. Yu and K. Lee, "Tracking across nonoverlapping cameras based on the unsupervised learning of camera link models," *ACM/IEEE Intl. Conf. on Distributed Smart Cameras,* 2012.

[5] T. D'Orazio, P. Mazzeo, and P. Spagnolo, "Color brightness transfer function evaluation for nonoverlapping multi camera tracking," *ACM/IEEE Intl. Conf. on Distributed Smart Cameras,* pp. 1-6, 2009.

[6] D. Makris, T. Ellis, and J. Black, "Bridging the gaps between cameras," *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, vol. 2, pp. 205-210, 2004.

[7] C. Chu, J. Hwang, Y. Chen, and S. Wang, "Camera link model estimation in a distributed camera network based on the deterministic annealing and the barrier method," *Proc. IEEE Conf. on ASSP*, pp. 997-1000, March, 2012.

[8] A. Gilbert and R. Bowden, "Tracking objects across cameras by incrementally learning inter-camera colour calibration and patterns of activity," *Proc. ECCV*, pp. 125-136, 2006.

[9] O. Javed, K. Shafique, Z. Rasheed, and M. Shah, "Modeling inter-camera space-time and appearance relationships for tracking across non-overlapping views," *Computer Vision and Image Understanding,* pp. 146-162, 2008.

[10] K. Chen, C. Lai, P. Lee, C. Chen and Y. Hung,, "Adaptive learning for target tracking and true linking discovering across multiple non-overlapping cameras," *IEEE Trans. on Multimedia,* vol. 13, pp. 625-638, 2011.

[11] C. Chu, J. Hwang, S. Wang and Y. Chen, "Human tracking by adaptive Kalman filtering and multiple kernels tracking with projected gradients," *ACM/IEEE Intl. Conf. on Distributed Smart Cameras,* Aug, 2011.

[12] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *Intl. Journal of Computer Vision*, vol. 60, no. 2, pp. 91–110, 2004.