Learning to Track and Identify Players from Broadcast Sports Videos

Wei-Lwun Lu, Jo-Anne Ting, James J. Little, Kevin P. Murphy

Abstract—
Tracking and identifying players in sports videos filmed with a single pan-tilt-zoom camera has many applications, but it is also a challenging problem. This article introduces a system that tackles this difficult task. The system possesses the ability to detect and track multiple players, estimates the homography between video frames and the court, and identifies the players. The identification system combines three weak visual cues, and exploits both temporal and mutual exclusion constraints in a Conditional Random Field. In addition, we propose a novel Linear Programming Relaxation algorithm for predicting the best player identification in a video clip. In order to reduce the number of labeled training data required to learn the identification system, we make use of weakly supervised learning with the assistance of play-by-play texts. Experiments show promising results in tracking, homography estimation, and identification. Moreover, weakly supervised learning with play-by-play texts greatly reduces the number of labeled training examples required. The identification system can achieve similar accuracies by using merely 200 labels in weakly supervised learning, while a strongly supervised approach needs a least 20000 labels.

Index Terms—sports video analysis, identification, tracking, weakly supervised learning

1 Introduction

1.1 Motivation

Intelligent sports video analysis systems have many commercial applications and have spawned much research in the past decade. Recently, with the emergence of accurate object detection and tracking algorithms, the focus is on a detailed analysis of sports videos, such as player tracking and identification.

Automatic player tracking and identification has many commercial applications. From the coaching staff's point of view, this technology can be used to gather game statistics in order to analyze their competitors' strength and weakness. TV broadcasting companies also benefit by using such systems to create star-camera views – video streams that highlight star players. Since both tasks are currently performed by human annotators, automating these processes would significantly increase the production speed and reduce cost.

This article is thus devoted to developing a system to automatically track and identify players from a single broadcast video taken from a pan-tilt-zoom camera. The only input of the system is a single broadcast sports video. The system will automatically localize and track players, recognize the players' identities, and estimate their locations on the court. As opposed to most existing techniques that utilize video streams taken by multiple cameras [1], the proposed system has two advantages: (1) The system works with a single uncalibrated pan-tilt-zoom camera. This avoids the expensive process of installing a calibrated camera network in the stadium, which is usually impractical for amateur sports fields. (2) The system has the ability to analyse existing sports video archive taken by uncalibrated moving cameras (*e.g.* sports videos in YouTube). To the best of our knowledge, the proposed system is the first one that is able to track and identify players from a single video, and we believe it could benefit a wider range of users.

- *W.L. Lu is with Ikomed Technologies, Vancouver, BC, Canada.*
- *J.J. Little is with the Department of Computer Science, University of British Columbia, Vancouver, BC, Canada.*
- *K.P. Murphy is with Google Research, Mountain View, CA, USA.*
- *J.A. Ting is with Bosch Research, Palo Alto, CA, USA.*

1.2 Challenges

Tracking and identifying players in broadcast sports videos is a difficult task. Tracking multiple players in sports videos is challenging due to several reasons: (1) The appearance of players is ambiguous – the players of the same team wear uniforms of the same colors; (2) Occlusions among players are frequent and sometimes long; (3) Players have more complicated motion pattern, and they do not have a fixed enter/exit locations, as opposed to the pedestrians in surveillance videos.

Identifying players is even a harder problem. Most existing systems can only recognize players in close-up views, where facial features are clear, or jersey numbers are visible. However, in frames taken from a court view, faces become blurry and indistinguishable, and jersey numbers are broken and deformed. In addition, since players constantly change their poses and orientations, both faces and jersey numbers are only visible in limited cases. Colors are also weak cues because players on the same team have identical jersey color, and many have very similar hair and skin colors.

One possible solution for player identification is to train a classifier that combines multiple weak cues, as we

proposed in [2]. However, [2] requires a large amount of labeled training data, and acquiring those labels is time-consuming. A learning strategy that requires fewer labeled examples is thus preferable.

1.3 Related work

Reviewing all relevant tracking literature is beyond the scope of this article (see [3] for a survey). Here, we discuss the most relevant ones for tracking sport players. Single target trackers such as Boosted Particle Filter (BPF) [4], [5] or mean-shift [6], [7] can be used to track players. However, these trackers lack a global view of the scene and thus usually fail to track players when occlusion occurs. Another strategy is to first detect players and then link detections into tracks. This technique requires an affinity function between detections, which can be defined by heuristics [8], [9], or learnt from training data [10]. Then, one can use either Linear Programming [11], or MCMC [12] to search for the best way to link detections into tracks. The algorithm proposed in this article resembles [5], but we also borrow ideas from [11].

Previous player identification systems in the sports domain have focused on video streams filmed with close-up cameras. These systems relied on recognizing either frontal faces [13], [14] or jersey numbers [15], [16]. Unfortunately, these systems only apply to close-up and frontal-view images where facial features are clear and jersey numbers are visible. Recently, [1] introduced a system that recognizes players from 8 stationary cameras, but it still requires 2 cameras with close-up views. From the best of our knowledge, our previous work [2] is the first one that tracks and identifies players in a single broadcast sports video filmed from the court view.

For learning a classifier with fewer numbers of labeled data, a rich literature can be found in the field of semi-supervised learning (see [17] for a complete survey). For learning from video streams, one solution is the crowd-sourced marketplace [18], but it still requires human labour. An alternative solution is the use of *weak labels*. A typical source of weak labels are the captions/subtitles that come with movies, which specify what or who is in the video frame. Such weakly labeled data is often cheaply available in large quantities. However, a hard correspondence problem between labels and objects has to be solved. An early example of this approach is [19], which learnt a correspondence between captions and regions in a segmented image. [20], [21], [22], [23] learnt a mapping between names in subtitles of movies to appearances of faces, extracted by face detection systems. Others have also attempted to learn action recognition systems from subtitles [24], [25], [26] or a storyline [27]. In this article, we adopt a similar approach that utilizes the play-by-play text to train a player identification system. Play-by-play text has been previous used as *features* in sports tactic analysis [28], but this article is the first attempt on using play-by-play text to train a player identification system. In section 6.3.3, we also show that our proposed new method gives better results than using one of the current best existing methods [20] for learning from weak labels.

00:42.3 [LAL 51-29]	**Bynum Slam Dunk Shot: Made (5 PTS) Assist: Bryant (5 AST)**
00:35.8	Bynum Foul : Personal (1 PF)
00:35.8	Bryant Substitution replaced by Blake
00:31.1	Brown Foul : Personal (1 PF)

Fig. 2. The *play-by-play* of a basketball game, which shows the time and people involved in important events.

1.4 Contributions

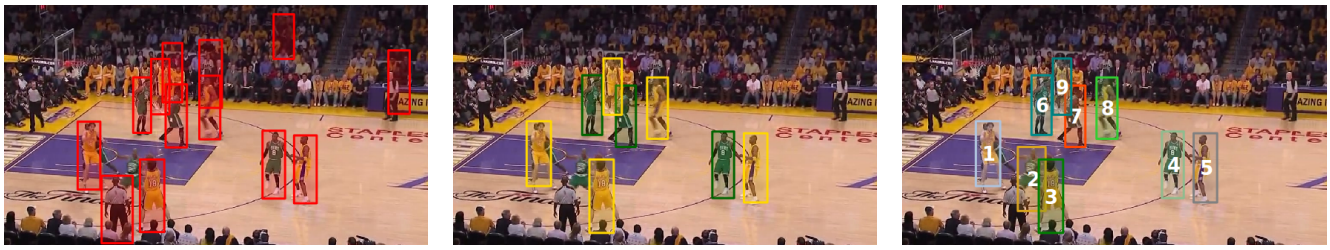
This article introduces an intelligent system that tracks and identifies players in broadcast sports videos filmed by a single pan-tilt-zoom camera. The contribution of this article is two-fold. Firstly, this article presents a complete solution for tracking and identifying players in broadcast sports videos. The system possesses the ability to detect and track multiple players, recognizes players by using weak visual cues, and estimates the homography between video frames and the court. Secondly, the article presents a weakly supervised learning algorithm that greatly reduces the number of labeled data needed to train an identification system. We also introduce a new source of weak labels – the play-by-play text, which is widely available in sports videos.

Compared to our previous work [2], the new system has a better identification accuracy owing to a better inference algorithm that relies on a Linear Programming Relaxation formulation. In addition, the number of labeled images is also greatly reduced from 20000 to 200, owing to weakly supervised learning and the use of the play-by-play text as the weak labels.

2 Video Pre-processing

2.1 Shot segmentation

A typical broadcast sports video consists of different kinds of shots: close-ups, court views, and commercials. Since this article focuses on video frames taken from the court view (see Figure 1 for examples), the first step is to segment videos into different shots. We achieve this by utilizing the fact that different shots have distinctive color distributions. For example, close-up views are dominated by jersey and skin colors, while in court views, colors of court and spectators prevail. Specifically, we train a Hidden Markov Model (HMM) [30] for shot segmentation. The emission probability is modelled by a Gaussian Mixture Model (GMM) where features are RGB color histograms, and the transition probability is formulated to encourage a smooth change of shots. We then run the Viterbi algorithm [30] to find the optimal configuration of the graphical model in order to segment the video into different shots.



(a) detection (b) team classification (c) tracking

Fig. 1. (a) Automatic player detection generated by the DPM detector [29]. (b) Automatic team classification. Celtics are in green, and Lakers are in yellow. (c) Automatic tracking by associating detections into tracklets. Numbers represent the track ID not player identities.

2.2 Play-by-play processing

People keep the logs of important events for most sports games. The log is called the *play-by-play*, which is usually available in real-time during professional games and can be freely downloaded from the Internet. Figure 2 shows an example of play-by-play text downloaded from the NBA website. We see that it shows event types (e.g., "Dunk", "Substitution"), player names (e.g., "Bryant", "Bynum"), and timestamps (e.g., "00:42.3").

In this article, we only focus on player identities, rather than actions. Since the play-by-play provides the names of the starting players and the substitution events (see the second log in Figure 2), we can use a finite-state machine to estimate the players on the court at any given time. To assign a game time-stamp to every video frame, we run an optical character recognition (OCR) system [31] to recognize the clock numbers showing on the information bar overlaid on the videos. The OCR system has nearly perfect accuracy because the clock region has a fixed location, and the background of the clock is homogeneous.

3 Player Tracking

This paper takes a *tracking-by-detection* approach to track sports players in video streams. Specifically, we run an player detector to locate players in every frame, and then we associate detections over frames with player tracks.

3.1 Player detection

We use the Deformable Part Model (DPM) [29] to automatically locate sport players in video frames. The DPM consists of 6 parts and 3 aspect ratios and is able to achieve a precision of 73% and a recall of 78% in the test videos. Figure 1(a) shows some DPM detection results in a sample basketball video. We observe that most false positives are generated from the spectators and referees, who have similar shapes to basketball players. Moreover, since the DPM detector applies non-maximum suppression after detection, it may fail to detect players when they are partially occluded by other players.

3.2 Team classification

In order to reduce the number of false positive detections, we use the fact that players of the same team wear jerseys whose colors are different from the spectators, referees, and the other team. Specifically, we train a logistic regression classifier [32] that maps image patches to team labels (Team A, Team B, and other), where image patches are represented by RGB color histograms. We can then filter out false positive detections (spectators and referees) and, at the same time, group detections into their respective teams. Notice that it is possible to add color features to the DPM detector and train a player detector for a specific team [33]. However, [33] requires a larger labeled training data, while the proposed method only needs a handful examples.

After performing this step, we significantly boost precision to 97% while retaining a recall level of 74%. Figure 1(b) shows some team classification results.

3.3 Player tracking

We perform tracking by associating detections with tracks and use a one-pass approach similar to [5]. Starting from the current frame, we assign detections to existing tracks. To ensure the assignment is one-to-one, we use bi-partite matching where the matching cost is the Euclidean distances between centers of detections and predictive locations of tracks. Specifically, let $C_{i,j}$ be the cost of associating the i-th track to the j-th detection. We compute the cost function by $C_{i,j} = ||\widehat{\mathbf{s}}_i - \mathbf{d}_j||$, where $\widehat{\mathbf{s}}_i = [\hat{x}_i, \hat{y}_i, \hat{w}_i, \hat{h}_j]^T$ is the *predicted* bounding box of the i-th track, and $\mathbf{d}_j = [x_j, y_j, w_j, h_j]^T$ is the j-th bounding box generated by the detector. As shown in Figure 3, the relationship between time and player's locations is approximately linear in a short time interval. Therefore, we compute the predictive bounding box at time t by (we drop subscript i for simplicity): $\widehat{\mathbf{s}} = t\mathbf{a}_t + \mathbf{b}_t$, where both $\mathbf{a}_t$ and $\mathbf{b}_t$ are both 4×1 vectors. We estimate $\mathbf{a}_t$ and $\mathbf{b}_t$ by utilizing the trajectory of the i-th track in the previous T frames. Specifically, we optimize the following least-square system with respect to $\mathbf{a}_t$ and $\mathbf{b}_t$:

$$\min_{\mathbf{a}_t, \mathbf{b}_t} \sum_{k=1}^{T} (1-\alpha)^k ||(t-k)\mathbf{a}_t + \mathbf{b}_t - \mathbf{s}_{t-k}||^2 \qquad (1)$$

where $\mathbf{s}_{t-k}$ is the track's estimated bounding box at time $t-k$, and $0 < \alpha < 1$ is a small positive constant. Equation 1 can be thought of fitting a linear function to map time t to a bounding box $\mathbf{s}_t$[1].

After assigning detections to existing tracks, the next step is to update the state estimate of players. The state vector we want to track at time t is a 4-dimensional vector $\mathbf{s}_t = [x, y, w, h]^T$, where (x, y) represents the center of the bounding box, and (w, h) are its width and height, respectively. We use a linear-Gaussian transition model: $p(\mathbf{s}_t|\mathbf{s}_{t-1}) = \mathcal{N}(\mathbf{s}_t|\mathbf{s}_{t-1}, \sigma_d^2\mathbf{I})$, and a linear-Gaussian observation model: $p(\mathbf{d}_t|\mathbf{s}_t) = \mathcal{N}(\mathbf{d}_t|\mathbf{s}_t, \sigma_e^2\mathbf{I})$, where $\mathbf{I}$ is a 4×4 identity matrix, and σ_d and σ_e is the variance for the transition and observation model, respectively. Since both the transition and observation models are linear-Gaussians, we can update the current state $\mathbf{s}_t$ by using a Kalman Filter [34]. If there is no associated detection, we use Kalman Prediction [34] to fill the gap.

A new track is initialized for any unassigned detection. However, the track is dropped if it fails to have a sufficient number of detections associated with it after a short time. Existing tracks are also removed if they are close to the image boundaries, or if they fail to have any detections associated with them for a sufficient period of time (1 sec in our experiments).

Figure 3 shows results of tracking basketball players. Every dot in the graph represents the center of a bounding box, where different colors represent different tracklets. The tracking algorithm has a 98% precision with an better recall of 82%. The recall improves because the original detections are temporally sparse, and the tracking bridges the gap between disjointed detections by Kalman Prediction. For example, the tracking system successfully locates track #2 in Figure 1(c), while DPM fails to detect the players in Figure 1(a).

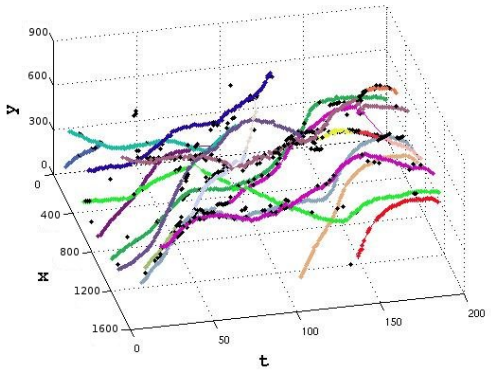


Fig. 3. The x-y-t graph of tracking results, where (x, y) is the center of a bounding box, and t is the time. Every dot in the graph represents a detected bounding box, where different colors represent different tracklets.

4 Player Identification

The next step is to automatically identify sports players. Face recognition is infeasible in this domain, because image resolution is too low even for human to identify players. Recognizing jersey numbers is possible, but still very challenging. We tried to use image thresholding to detect candidate regions of numbers, and run an OCR [31] to recognize them, as in [1]. However, we got very poor results because image thresholding cannot reliably detect numbers, and the off-the-shelf OCR is unable to recognize numbers on deformed jerseys. Frequent pose and orientation changes of players further complicate the problem, because frontal views of faces or numbers are very rare from a single camera view, as opposed to [1] that accessed to 8 cameras.

We adopt a different approach, ignoring face and number recognition, and instead focusing on identification of players as entities. We extract several visual features from the entire body of players. These features can be faces, numbers on the jersey, skin or hair colors. By combining all these weak features together into a novel Conditional Random Field (CRF), the system is able to automatically identify sports players, even in video frames taken from a single pan-tilt-zoom camera.

4.1 Graphical model

Given the tracking results, we construct a Conditional Random Field (CRF) for the entire video clip, as shown in Figure 4. The CRF consists of N feature nodes $\mathbf{x}_{t,d}$ that represent the observed feature vectors of the player detection d at time t, where N is the number of detections in a video clip. The CRF also has N identity nodes $y_{t,d}$ that represent the player identity of detection d at time t, whose values will be estimated given all observed $\mathbf{x}$. The feature node $y_{t,d}$ has $|\mathcal{C}|$ possible values, where $\mathcal{C}$ is a set of all possible player classes.

We first connect identity nodes $y_{t,d}$ to corresponding feature nodes $\mathbf{x}_{t,d}$. The node potential is defined as:

$$\psi_{\text{unary}}(y_{t,d}, \mathbf{x}_{t,d}) = p(y_{t,d}|\mathbf{x}_{t,d}, \boldsymbol{\theta}) \cdot p(y_{t,d}) \tag{2}$$

where $\mathbf{x}_{t,d}$ are feature vectors and $\boldsymbol{\theta}$ are parameters. We model $p(y_{t,d}|\mathbf{x}_{t,d}, \boldsymbol{\theta})$ as multi-class logistic regression:

$$p(y_{t,d} = k|\mathbf{x}_{t,d}, \boldsymbol{\theta}) = \frac{\exp(\boldsymbol{\theta}_k^T \mathbf{x}_{t,d})}{\sum_j \exp(\boldsymbol{\theta}_j^T \mathbf{x}_{t,d})} \tag{3}$$

Parameters $\boldsymbol{\theta}$ are trained in a discriminative way using either fully-labeled or weakly-labeled training data.

The prior probability $p(y_{t,d})$ expresses our initial belief of the presenting players. During testing, $p(y_{t,d})$ is set to be a uniform distribution over all possible players and hence this term can be ignored; that is, player identity is estimated only from the visual features. However, in training time, $p(y_{t,d})$ can be adjusted if some prior knowledge is available (see section 4.4 for details).

We then connect identity nodes $y_{t,i}$ and $y_{t+1,j}$ if they belong to the same track, where tracking is done by

1. Notice that we fit a linear model only when we have a sufficient number of training data. Otherwise, first-order auto-regression (i.e., $\widehat{\mathbf{s}} = \mathbf{s}_{t-1}$) is used.

using the algorithm in section 3. We use this edge to encourage temporal smoothness of identities within a track. The edge potential is defined as:

$$\psi_{\text{time}}(y_{t,i}, y_{t+1,j}) = \begin{cases} 1-\epsilon & \text{if } y_{t,i} = y_{t+1,j} \\ \epsilon & \text{otherwise} \end{cases} \quad (4)$$

where $0 \leq \epsilon \leq 1$ is a fixed parameter that reflects the tracking error. Setting ϵ to 0 forces all identity nodes y within a track to have the same estimated value. On the other hand, setting ϵ to a positive value allows identity nodes y to change values within a track. In our previous work [2], we set ϵ to a small positive value to account for tracking errors. However, here we set $\epsilon = 0$ because this simplifies the optimization problem and does not affect the identification accuracy in our experiments.

We also connect all pairs of identity nodes $y_{t,i}$ and $y_{t,j}$ if they appear in the same time t. We then introduce an edge potential that enforces mutual exclusion:

$$\psi_{\text{mutex}}(y_{t,i}, y_{t,j}) = \begin{cases} 1 & \text{if } y_{t,i} \neq y_{t,j} \\ 0 & \text{otherwise} \end{cases} \quad (5)$$

This potential specifies the constraint that a player can be appear only *once* in a frame. For example, if the i-th detection $y_{t,i}$ has been assign to Bryant, $y_{t,j}$ cannot have the same identity because Bryant is impossible to appear twice in a frame.

The joint posterior of the entire CRF is then:

$$\begin{aligned} p(\mathbf{y}|\mathbf{x}, \boldsymbol{\theta}) = & \left(\prod_{t=1}^{|\mathcal{T}|} \prod_{d=1}^{|\mathcal{D}_t|} \psi_{\text{unary}}(y_{t,d}, \mathbf{x}_{t,d}) \right) \\ & \cdot \left(\prod_{t=1}^{|\mathcal{T}|} \prod_{d=1}^{|\mathcal{D}_t|} \psi_{\text{time}}(y_{t,d}, y_{t+1,\text{succ}(t,d)}) \right) \\ & \cdot \left(\prod_{t=1}^{|\mathcal{T}|} \prod_{d=1}^{|\mathcal{D}_t|} \prod_{j \neq d} \psi_{\text{mutex}}(y_{t,d}, y_{t,j}) \right) \end{aligned} \quad (6)$$

where $\text{succ}(t, d)$ is the next node (if it exists) that is connected to $y_{t,d}$ in the track, $|\mathcal{D}_t|$ is the number of detections in frame t, and $|\mathcal{T}|$ is the total number of frames in the video clip.

4.2 Visual Features

The feature vector $\mathbf{x}_{t,d}$ consists of three different kinds of visual cues: maximally stable extremal regions (MSER) [35], SIFT features [36], and RGB color histograms. The MSER regions [35] are those stable segments whose colors are either darker or lighter than their surroundings. They are useful for detecting texts in natural scenes because text has often uniform color and high contrast. We first detect the MSER regions [35] and normalize them according to [37], as shown in Figure 5(b). For every MSER region, a 128-dimensional SIFT descriptor is computed and quantized into one of 300 visual words using a learnt codebook (the codebook is learnt using k-means clustering). The MSER representation of the image is a 300-dimensional bag-of-words bit vector, where a value of 1 indicates presence of the corresponding visual word and 0 otherwise.

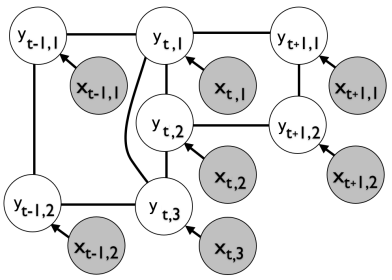


Fig. 4. Graphical model for training clips. $\mathbf{x}$ are detections. Mutex arcs exist between detection identities $\mathbf{y}$ in a frame. Temporal arcs exist between $\mathbf{y}$ nodes across frames.

The SIFT features [36] are stable local patches invariant to scale and affine transformation. We first detect the SIFT features, as shown in Figure 5(c). Then, we compute the SIFT descriptors and quantize them into 500 visual words. We use more visual words for SIFT because there are more SIFT features than the MSER regions.

Although colors are weaker features (players of the same team wear the same uniform), skin color may provide some information for player identification. To account for the colors of limbs, hair, etc., we also compute RGB color histograms from the image. For the RGB color histogram, we use 10 bins for each of the R, G and B channels. We treat the three colors independently, so the full histogram has in total 30 bins.

Figure 5 shows an example of the MSER and SIFT features. We can see that faces are blurred, while numbers can only be clearly seen in the last frame. Since we do not segment the player from the background, some MSER and SIFT features may be generated from the background. However, these features will not affect identification results because they are assigned lower weights in Equation 3 due to the use of L1 regularization.

4.3 Inference

We estimate the identities of players by maximizing the log posterior in Equation 6 with respect to the identity variables $\mathbf{y}$. We first represent the identity variable y by an auxiliary column vector $\mathbf{z}$, and we have $\mathbf{z} = [z_1 \dots z_C]^T$ where $z_c = 1$ if $y = c$, and 0 otherwise. Then, we re-write the joint posterior as a Gibbs distribution:

$$p(\mathbf{z}|\mathbf{x}, \boldsymbol{\theta}) = \frac{1}{Z(\boldsymbol{\theta})} \exp(E(\mathbf{z}, \mathbf{x})) \quad (7)$$

where $Z(\boldsymbol{\theta})$ is the normalization constant. $E(\mathbf{z}, \mathbf{x})$ is a log-linear energy function:

$$E(\mathbf{z}, \mathbf{x}) = \sum_{t=1}^{|\mathcal{T}|} \sum_{d=1}^{|\mathcal{D}_t|} \mathbf{f}_{t,d}^T \mathbf{z}_{t,d} + \mathbf{q}^T \mathbf{z}_{t,d} \quad (8)$$

$$+ \sum_{t=1}^{|\mathcal{T}|} \sum_{d=1}^{|\mathcal{D}_t|} \mathbf{z}_{t,d}^T \mathbf{G}\ \mathbf{z}_{t+1,\text{succ(t,d)}} \quad (9)$$

(a) raw image (b) MSER visual words (c) SIFT visual words

Fig. 5. (a) Raw image patches extracted from the tracking system. (b) Green ellipses represent the detected MSER regions [35]. (c) Red circles represent the detected SIFT interest points [36].

$$+\sum_{t=1}^{|\mathcal{T}|}\sum_{d=1}^{|\mathcal{D}_t|}\sum_{j\neq d}\mathbf{z}_{t,d}^T\mathbf{J}\ \mathbf{z}_{t,j} \tag{10}$$

We represent the feature potentials by a vector $\mathbf{f}_{t,d} = [f_{t,d,1}\dots f_{t,d,C}]^T$ where $f_{t,d,c} = \boldsymbol{\theta}_c^T\mathbf{x}_{t,d}$. The prior probability is represented by $\mathbf{q}$ where $\mathbf{q}(i) = \log p(y=i)$. The temporal constraints are encoded by a matrix $\mathbf{G}$, where $\mathbf{G}(i,i) = \log(1-\epsilon)$ and $\mathbf{G}(i,j) = \log(\epsilon)$ if $i \neq j$. The mutual exclusion constraints are represented by the matrix $\mathbf{J}$, where $\mathbf{J}(i,i) = -\inf$ and $\mathbf{J}(i,j) = 0$ if $i \neq j$.

We further simplified the problem by setting $\epsilon = 0$, and thus both Equation 9 and 10 become hard constraints. We then re-write the optimization problem as:

$$\max_{\mathbf{z}} \sum_{t=1}^{|\mathcal{T}|}\sum_{d=1}^{|\mathcal{D}_t|}\mathbf{f}_{t,d}^T\mathbf{z}_{t,d} + \mathbf{q}^T\mathbf{z}_{t,d} \tag{11}$$

subject to the following constraints:

$$\sum_{c=1}^{|\mathcal{C}|} z_{t,d,c} = 1 \qquad \forall t\in\mathcal{T}, d\in\mathcal{D}_t \tag{12}$$

$$z_{t,d,c} - z_{t+1,\text{succ}(t,d),c} = 0 \qquad \forall t\in\mathcal{T}, d\in\mathcal{D}_t, c\in\mathcal{C} \tag{13}$$

$$z_{t,d,c} + z_{t,j,c} \leq 1 \qquad \forall t\in\mathcal{T}, c\in\mathcal{C}, d\neq j \tag{14}$$

$$z_{t,d,c} \geq 0 \qquad \forall t\in\mathcal{T}, d\in\mathcal{D}_t, c\in\mathcal{C} \tag{15}$$

where $\mathcal{T}$ is a set of frames, $\mathcal{C}$ is a set of all possible players, and $\mathcal{D}_t$ is a set of detections at time t. Equation 12 ensures that there is exactly one variable in $\mathbf{z}_{t,d}$ being 1. Equation 13 ensures that both $z_{t,d,c}$ and $z_{t+1,\text{succ}(t,d),c}$ have the same value, and thus it enforces the temporal constraint. Equation 14 prevents both $z_{t,d,c}$ and $z_{t,j,c}$ being 1, which violates the mutual exclusion constraint. Equation 15 ensures that all $\mathbf{z}$ are non-negative.

Since solving the above optimization problem with respective to binary variables $\mathbf{z}$ is hard, we relaxed the problem and allowed $\mathbf{z}$ to take real values. We then see that Equation 11 becomes a Linear Programming (LP) problem with linear constraints in Equation 12–15. This problem can be efficiently[2] solved by standard optimization algorithms [38]. After solving the problem for real-valued $\mathbf{z}$, the player identity $y_{t,d}$ can be obtained by $y_{t,d} = \text{argmax}_c z_{t,d,c}$.

2. In our Matlab implementation, it takes about 3 seconds to perform inference for 1000 frames in a computer with four 2.8GHz CPUs. In the same time, our Loopy BP implementation takes more than 100 seconds.

Other inference algorithms are also possible. For example, in our previous work [2], we applied sum-product Loopy Belief Propagation (LBP) [39] to compute the marginal distribution $p(y_{t,d}|\mathbf{x})$, and then took the maximum independently to every marginal to generate a sub-optimal configuration. However, this approach sometimes produces a configuration that violates mutual exclusion constraints. One can also apply max-product BP [39], but the speed is much slower than the proposed LP formulation.

4.4 Learning

The goal of learning is to find the best parameters $\boldsymbol{\theta}$ in the feature potentials ψ_{feat}. In other words, we want to train a classifier $p(y_{t,d}|\mathbf{x}_{t,d},\boldsymbol{\theta})$ that maps feature vectors $\mathbf{x}_{t,d}$ to player class $y_{t,d}$ given some labeled training data.

If we can afford a large number of labeled training data as in [2], the most straightforward approach is *supervised* learning. We maximize the log-posterior of labeled training data with a L1 regularizer as in [40]:

$$\max_{\boldsymbol{\theta}} \sum_t\sum_d \log p(y_{t,d}|\mathbf{x}_{t,d},\boldsymbol{\theta}) - \alpha||\boldsymbol{\theta}||_1 \tag{16}$$

where α is a constant. The above optimization problem can be efficiently solved by the algorithm introduced by Schmidt *et al.* [41]. Here, we assumed that all training data is labeled, *i.e.*, every detected player $\mathbf{x}_{t,d}$ has a corresponding label $y_{t,d}$. The major problem of supervised learning is that it usually requires a large amount of labeled training data. For example, in [2], more than 20000 labeled training data is needed in order to train an accurate identification system. Unfortunately, labeled training data is very expensive to acquire.

Here we take a different approach. Starting with a small number of labeled training data, we use *semi-supervised* learning to train the identification system. We then take advantage of the *play-by-play* text that is available for most professional sports games to further reduce the number of labeled training data required.

The semi-supervised learning algorithm is a variant of Expectation-Maximization (EM) [42]. We start with a small number of randomly labeled training data. This is achieved by presenting random image patches of players to human annotators to label, where image patches are generated from training clips by the DPM detector.

Algorithm 1 EM for weakly supervised learning

estimate $\boldsymbol{\theta}_0$ by using labeled data $\mathbf{x}_{\mathcal{L}}$ and $y_{\mathcal{L}}$
$k = 0$
repeat
  $k = k + 1$
  $\hat{y}_{\mathcal{U}}$ = LinearProgramming($\mathbf{x}_{\mathcal{U}}$, $\boldsymbol{\theta}_{k-1}$) ▷ Eq. 11–15
  $\boldsymbol{\theta}_k$ = MultiLogitRegFit($y_{\mathcal{L}}$, $\mathbf{x}_{\mathcal{L}}$, $\hat{y}_{\mathcal{U}}$, $\mathbf{x}_{\mathcal{U}}$) ▷ Eq. 18
until convergence
return $\boldsymbol{\theta}_k$

We then compute the initial model parameters $\boldsymbol{\theta}_0$ by maximizing the log-posterior with a L1 regularizer, as in Equation 16. Then, in the first iteration, we predict the identities of the *unlabeled* training data by solving the LP problem in Equation 11 with the initial parameters $\boldsymbol{\theta}_0$. This is called the E-step because we compute the *expected* value of $y_{t,d}$ given the current model. Then, we optimize the log-posterior with *all* training data:

$$\begin{aligned} \max_{\boldsymbol{\theta}} \; & \sum_{u \in \mathcal{U}} \sum_{y_u} p(y_u | x_u, \boldsymbol{\theta}^{old}) \left[\log p(y_u | \mathbf{x}_u, \boldsymbol{\theta}) \right] \\ & + \sum_{l \in \mathcal{L}} \log p(y_l | \mathbf{x}_l, \boldsymbol{\theta}) - \alpha ||\boldsymbol{\theta}||_1 \end{aligned} \tag{17}$$

where $\mathcal{L}$ is the set of labeled data, $\mathcal{U}$ is the set of unlabelled data, and $\boldsymbol{\theta}^{old}$ are the parameters in the previous iteration. We approximate the summation over y_u by using the prediction $\hat{y}_u$ generated from LP, i.e.:

$$\max_{\boldsymbol{\theta}} \; \sum_{u \in \mathcal{U}} \log p(\hat{y}_u | \mathbf{x}_u, \boldsymbol{\theta}) + \sum_{l \in \mathcal{L}} \log p(y_l | \mathbf{x}_l, \boldsymbol{\theta}) - \alpha ||\boldsymbol{\theta}||_1 \tag{18}$$

Specifically, for labeled data, we use the groundtruth labels y provided by human annotators. For unlabelled data, we use the predicted label $\hat{y}$ computed in the E-step. The optimization problem can be efficiently solved by [41]. This is called the M-step because we *maximize* the log-posterior given expected labels generated from the E-step. We repeat this process until convergence to obtain the best parameters. Since our semi-supervised learning algorithm is a coordinate ascent algorithm, it is guaranteed to monotonically converge to a local maximum. Algorithm 1 summarizes the EM-based semi-supervised learning algorithm.

In the standard semi-supervised learning, we set the prior $p(\hat{y}_{t,d}) = \frac{1}{|\mathcal{C}|}$, where $|\mathcal{C}|$ is the number of all possible players. This means that the predicted label $\hat{y}_{t,d}$ has a uniform probability to be any of the $|\mathcal{C}|$ possible players. When the play-by-play text is available, we set:

$$p(\hat{y}_{t,d} = c) = \begin{cases} \dfrac{1}{|\mathcal{P}_t|} & \text{if } c \in \mathcal{P}_t \\ 0 & \text{otherwise} \end{cases} \tag{19}$$

where $\mathcal{P}_t$ is a set of player that appears in frame t, and $\mathcal{P}_t \subset \mathcal{C}$. We call this strategy *weakly supervised* learning because we are given additional constraints provided by the play-by-play text. Notice that the majority of data still remains unlabelled, *i.e.*, there is no one-to-one mapping between y_u and $\mathbf{x}_u$. Taking professional basketball games as an example, a team usually has 12 players in their roster, and therefore $|\mathcal{C}| = 12$. However, in any given time t, there are only 5 players on the court, and thus $|\mathcal{P}_t| = 5$. In the experiments, we will show that this play-by-play prior is *crucial* to train the identification system with very small number of labeled training data.

5 Homography Estimation

Knowing the players' locations on the court coordinates is needed for many applications such as the automatic collection of player statistics. In order to achieve this, a transformation between the image and court coordinates is required. Unfortunately, camera calibration algorithms [43], [44] are inapplicable due to the pan-tilt-zoom camera and the lack of an auxiliary plane in sports videos. We instead seek to compute the homography transformation between the image and the court plane, as in [45], [46], and [47].

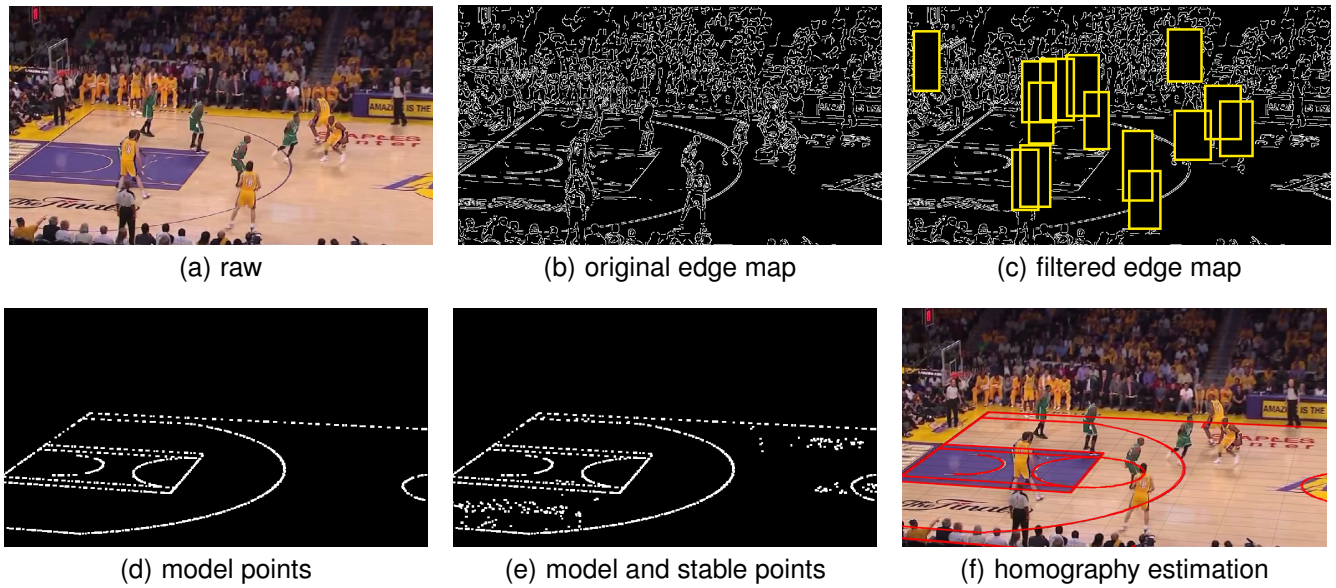
The relationship between a point $\mathbf{p} = [x, y]^T$ in the court coordinate and its corresponding point $\mathbf{p}' = [x', y']^T$ in the image coordinate can be specified by:

$$\mathbf{p}' = \frac{1}{h_7 x + h_8 y + 1} \begin{bmatrix} h_1 x + h_2 y + h_3 \\ h_4 x + h_5 y + h_6 \end{bmatrix} = f(\mathbf{p}; \mathbf{H}) \tag{20}$$

where $f(\mathbf{p}; \mathbf{H})$ is a nonlinear function, and $\mathbf{H} = [h_1 \dots h_8]$ is the homography. In order to obtain the pairs of correspondences $(\mathbf{p}, \mathbf{p}')$, the standard approach is to extract and match interest points such as SIFT features [36] in both images. However, in sports videos, most interest points are generated from players and spectators, but not from the court [47]. Using point correspondences of players and spectators to estimate the homography usually leads to unreliable results.

To tackle this problem, we apply a model-based approach inspired by [45]. Instead of matching interest points of two images, we match the court model with the *edges* of the video frames to establish correspondences. As shown in Figure 6(d), we construct a court model consisting of a point set $\mathcal{M} = [\mathbf{p}_1 \dots \mathbf{p}_n]$, where $\mathbf{p}_i = [x, y]^T$ is a 2D point. Since all professional courts of the same sport have identical dimensions, this model only has to be constructed once for a sport. The edges of the video frames are computed by the Canny detector [48], as shown in Figure 6(b). Since the edge detections contain many false responses from non-court objects, we further utilize the detection results in section 3.1 to remove edges caused by players, as shown in Figure 6(c).

This article adopts a variant of the Iterated Closest Points (ICP) [49] to estimate the homography. Firstly, we manually specify the first-frame homography $\mathbf{H}_1$ and transform the model such that $\mathcal{M}_1 = \mathbf{H}_1 \mathcal{M}$ (However, it is possible to use a semi-automatic approach to initialize the first-frame homography, as shown in [50]). Given the second frame, we use ICP to compute the frame-to-frame homography $\mathbf{H}_{12}$. Specifically, for every model point



(a) raw (b) original edge map (c) filtered edge map

(d) model points (e) model and stable points (f) homography estimation

Fig. 6. Estimating the homography in basketball videos. (a) A raw video frame. (b) Canny edges [48] of the frame. (c) Filtered Canny edges by dropping edges caused by players. (d) The original model points. (e) The original model point set augmented by stable points. (f) Homography estimation results by a variant of Iterated Closest Point (ICP) [49].

$\mathbf{p} \in \mathcal{M}_1$ that appears in the first frame, we search for the closest edge point $\mathbf{p}'$ in the second frame to establish the correspondence $(\mathbf{p}, \mathbf{p}')$. We drop correspondences with inconsistent edge normal directions to improve the robustness, as suggested in [45]. The homography is estimated by solving the following nonlinear least-square system [51]:

$$\min_{\mathbf{H}} \sum_i \left(f(\mathbf{p}_i; \mathbf{H}) - \mathbf{p}'_i\right)^2 \tag{21}$$

The optimization problem can be solved efficiently by the Levenberg-Marquardt algorithm [38]. Then, we transform the model points by the new homography. We iterate the process of finding closest points and least-square fitting until convergence to compute the final frame-to-frame homography $\mathbf{H}_{12}$. The second-frame homography can be thus derived by $\mathbf{H}_2 = \mathbf{H}_{12}\mathbf{H}_1$. Since camera motion is minor in two consecutive frames, ICP usually converges within 3-5 iterations. We repeat this process for all subsequent frames in a video clip. Figure 6(f) shows the homography estimation results, where red lines represent the transformed basketball model in a test video frame.

Sometimes, model points are sparse in video frames due to the camera's viewpoints or occlusions. To alleviate this problem, we augment the model point set by *stable points* that have a consistent homography as the court plane. These stable points are usually marks or logos on the court, which vary in different stadiums. Specifically, we construct an edge map $\mathcal{E}_t$ where $\mathcal{E}_t(\tilde{\mathbf{p}}) = 1$, for $\tilde{\mathbf{p}} = f^{-1}(\mathbf{p}', \mathbf{H}_t)$ where $f^{-1}(\cdot)$ is the inverse transformation that back-projects $\mathbf{p}'$ in the image coordinate to the court coordinate. If a point $\mathbf{p}'$ lies on the court plane, it will be always projected to the same position, and therefore we will have $\mathcal{E}_t(\tilde{\mathbf{p}}) = 1$ for many frames. Utilizing this fact, we maintain a score map $\mathcal{S}_t$ whose size is identical to $\mathcal{E}_t$, and update it according to:

$$\mathcal{S}_t = \alpha\mathcal{E}_t + (1-\alpha)\mathcal{S}_{t-1} \tag{22}$$

where $0 < \alpha < 1$ is a constant forgetting factor. A point is considered as *stable* if its score is higher than a threshold. Figure 6(e) shows the original model points and stable points of a professional basketball court. The stable points are automatically generated by Equation 22.

6 Results

6.1 Data

We used videos from the 2010 NBA Championship series (Los Angeles Lakers vs. Boston Celtics). The original videos consist of different kind of shots, and we only used the shots taken from the court view in this article. The training set consists of 153160 detected bounding boxes across 21306 frames. The test set has 20 video clips, consisting of 13469 frames with 90001 detections. The test clips varied in length, with the shortest at 300 frames and longest at 1400 frames. They also varied in level of identification difficulty. Labelling both training and test sets took us considerable effort (more than 300 hours). The size of this training data set is comparable or larger than others in the weakly labeled learning literature. For example, in previous work on high-resolution movies, [22] trained/tested on 49447 faces, and [20] trained on about 100000 faces.

6.2 Tracking evaluation

We use *precision* and *recall* to evaluate the performance of player detection, team classification, and multi-target

tracking. The DPM detector [29] had a moderate precision 73% and recall 78% in the test basketball dataset. DPM detected most players, but it also detected false positives such as referees and spectators. After team classification, the precision increases to 97% while it retains a recall level of 74% in the basketball dataset. The precision is significantly improved because we utilized jersey colors to discard false positives generated from referees and spectators, who wore clothes of different colors. The tracking algorithm has a 98% precision with an improved recall of 82% in the basketball dataset. This is because the original detections are temporally sparse, and tracking helps to bridge the temporal gap between disjointed detections.

We compare the proposed tracking algorithm (KF+DPM) with the Boosted Particle Filter (BPF) [4]. We use the same hockey dataset released by the authors [4], which consists of 1000 frames of a broadcast hockey game. Figure 7 shows the precision and recall of both the proposed algorithm and the BPF. The BPF has an average precision of 65.5% and an average recall of 50.8% over the 1000 frame hockey video (red lines). On the other hand, the proposed algorithm has a higher average precision of 91.8% and a higher average recall of 79.7% (blue lines). We also compare the performance by using the metrics proposed in [8]. As shown in Table 1, we compare the tracking results of KF+DPM and BPF by: Precision (PR), Recall (RC), False Alarm (FA), Mostly Tracked (MT), Mostly Lost (ML), and ID Switches (IDS). We can see that KF+DPM outperforms BPF in all aspects except ML. This is partially due to a better detector (DPM) which has a better precision and recall, and a more sophisticated motion model (Equation 1) that is able to resolve ID switches.

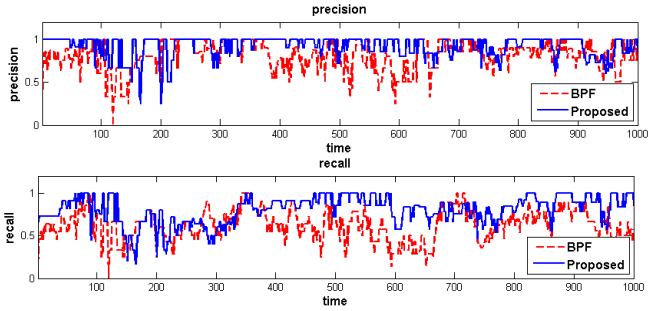


Fig. 7. Precision and recall of the proposed KF+DPM and BPF [4] on the hockey dataset (blue: KF+DPM, red: BPF).

method	PR	RC	FA	MT	ML	IDS
BPF [4]	65.5%	50.8%	2.3	16.0%	**14.7%**	37
KF+DPM	**91.8%**	**79.7%**	**0.6**	**56.3%**	20.8%	**27**

TABLE 1
Tracking results of the proposed KF+DPM tracker and BPF [4] on the hockey dataset, using the metrics in [8].

6.3 Identification evaluation

The identification system achieves an average accuracy of 85% for Lakers players and 89% for Celtics players. In addition, using weakly supervised learning greatly reduces the number of labels required to train the identification system (from 20000 to mere 200 labels for all players in a team). The following will provide a detailed analysis.

6.3.1 Comparison of features

We first compare the effectiveness of different features in Figure 8. In the experiments, we randomly choose 30000 labeled image patches, and then use the supervised learning approach in Equation 16 to train the identification system. Inference is performed by solving the linear programming problem in Equation 11–15.

Among the three appearance features, the SIFT bag of words has the strongest discriminative power, followed by the MSER bag of words and RGB color histograms. Colors are weak in this domain because players of the team wears uniforms of identical color, and many players have very similar skin and hair colors.

Since these three visual cues complement each other, combining the RGB color histograms, MSER, and SIFT yields the best results. For Lakers, the accuracy achieves 85%, while in Celtics, the accuracy becomes 89%.

6.3.2 Comparison of graphical models

We then compare the effectiveness of the graphical model in Figure 9. Similar to the previous experiments, we randomly choose 30000 labeled image patches, and then use the supervised learning to train the identification system. The IID model assumes that there is no connection between any identity nodes $y_{t,d}$. In other words, we identify players by only the feature potential in Equation 2, but without using the temporal and mutual exclusion potentials. The identification results are poor, having an accuracy about 50% for Lakers and 55% for Celtics. This demonstrates the challenges of identifying players from a single-view camera. Since players constantly change their poses and orientations, it is very difficult to identify players from a single image.

Adding temporal potentials to the model significantly boosts the performance. In Lakers, the accuracy increases to 81%, while in Celtics, the accuracy increases to 85%. Temporal edges in the graphical model are constructed by the tracking system. If the identification system has high confidence about the identity of even a single image in a track (*e.g.*, a frontal-view image of a player), this information can be passed forward and backward to the entire track, and helps identify images of obscure views.

Adding mutual exclusion potentials to the model slightly improves the accuracy. In Lakers, the accuracy increases to 85%, while in Celtics, the accuracy becomes 89%. Although the improvements are small, it is still necessary to include mutual exclusion constraints in order to prevent a duplicate of identities in a frame. In applications such as automatic collection of statistics and

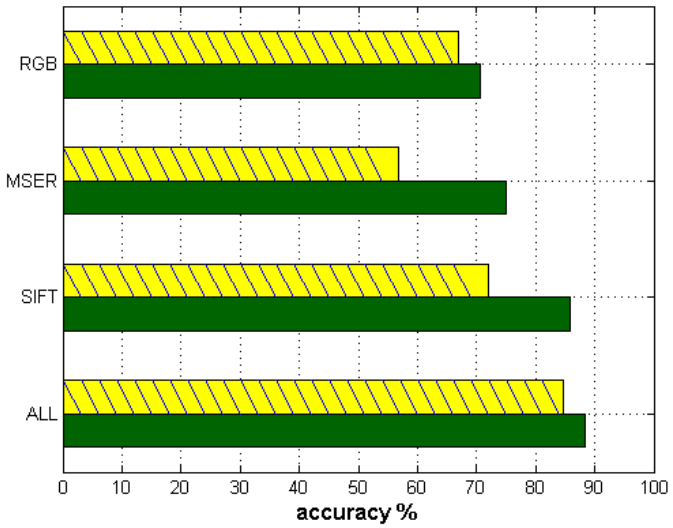


Fig. 8. Player identification results of different features for Lakers (yellow) and Celtics (green). We compare the effectiveness of RGB color histograms, MSER bag of words [35], SIFT bag of words [36], and a combination of all features (ALL).

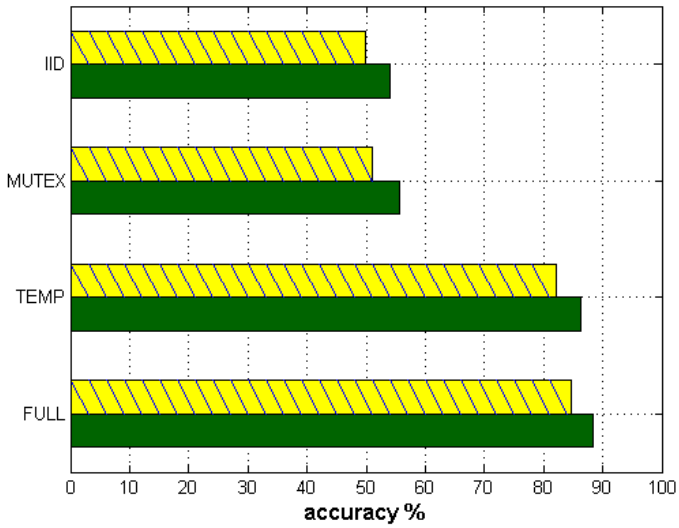


Fig. 9. Player identification results of different graphical models for Lakers (yellow) and Celtics (green): feature potentials only (IID), feature and mutex potentials (MUTEX), feature and temporal potentials (TEMP), and the full graphical model (FULL).

star-camera view, such duplications would significantly reduce the quality, and thus they should be avoided.

6.3.3 Comparison of learning strategies

Figure 10 compares different learning strategies for player identification. In the training dataset, we perform supervised [2], semi-supervised, and weakly supervised learning, with different number of labeled training data. Then, we test the learnt classifiers in the testing dataset while no labels (neither strong nor weak labels) are available. Since some algorithms have random initialization, we repeat the experiments for 10 times to compute the mean and standard deviation.

The *supervised learning* utilizes only labeled training data $y_{\mathcal{L}}$ and $\mathbf{x}_{\mathcal{L}}$ to train the model using Equation 16. We can observe that the identification accuracy in the testing dataset converges slowly with the increase of labeled training data[3]. In both Celtics and Lakers, accuracies converge after using more than 20000 labeled training examples.

The *semi-supervised* approach uses both labeled training data $y_{\mathcal{L}}$ and $\mathbf{x}_{\mathcal{L}}$, and unlabelled training data $y_{\mathcal{U}}$ and $\mathbf{x}_{\mathcal{U}}$. This approach uses the EM-based algorithm (Algorithm 1) to train the model parameters $\boldsymbol{\theta}$. Since play-by-play texts are not provided, we set the prior to an uniform distribution over all possible players, *i.e.*, $p(\hat{y}_{t,d}) = \frac{1}{|\mathcal{C}|}$. The accuracies of semi-supervised learning converge faster than the supervised one. Using only 2000 labeled examples, semi-supervised learning can achieve similar identification accuracies as the supervised one.

3. Since no strong/weak labels are used during testing, the accuracy will not achieve 100%.

The *weakly supervised* approach also uses both labeled and unlabelled training data, and it also applies the EM-based algorithm (Algorithm 1). However, the weakly supervised approach takes advantages of additional constraints provided by the play-by-play texts, and it uses the prior in Equation 19. We can observe that weakly supervised learning converges much faster than the semi-supervised one, and it can achieve similar accuracies by using merely 200 labeled examples. This is a significant reduction of labeled training data, compared with 2000 labels needed for semi-supervised learning, and 20000 labels required for supervised learning.

For comparison, we also show results of ambiguous label learning [20]. Ambiguous label learning assumes that every training image is associated with a set of labels, one of which is the correct label for the image. Since facial features in the original implementation of [20] are not suitable in this case, we replace them by the proposed visual features. We train the classifier using all our training images, with corresponding label sets provided by the play-by-play texts (the set size is 1 for labeled images and 5 for unlabelled images). After training, we use the same LP inference algorithm to identify players, instead of the IID approach used in [20]. We can see that ambiguous label learning performs better than the proposed EM-based algorithm while using a very small amount of labeled data (10–30 labels, or 1–3 labeled images per player). This is because ambiguous label learning has a more sophisticated mechanism to deal with weakly labeled data. However, after giving 30–50 labels (3–5 labeled images per player) to initialize the model in EM, the proposed weakly supervised approach quickly outperforms the ambiguous label learning. This

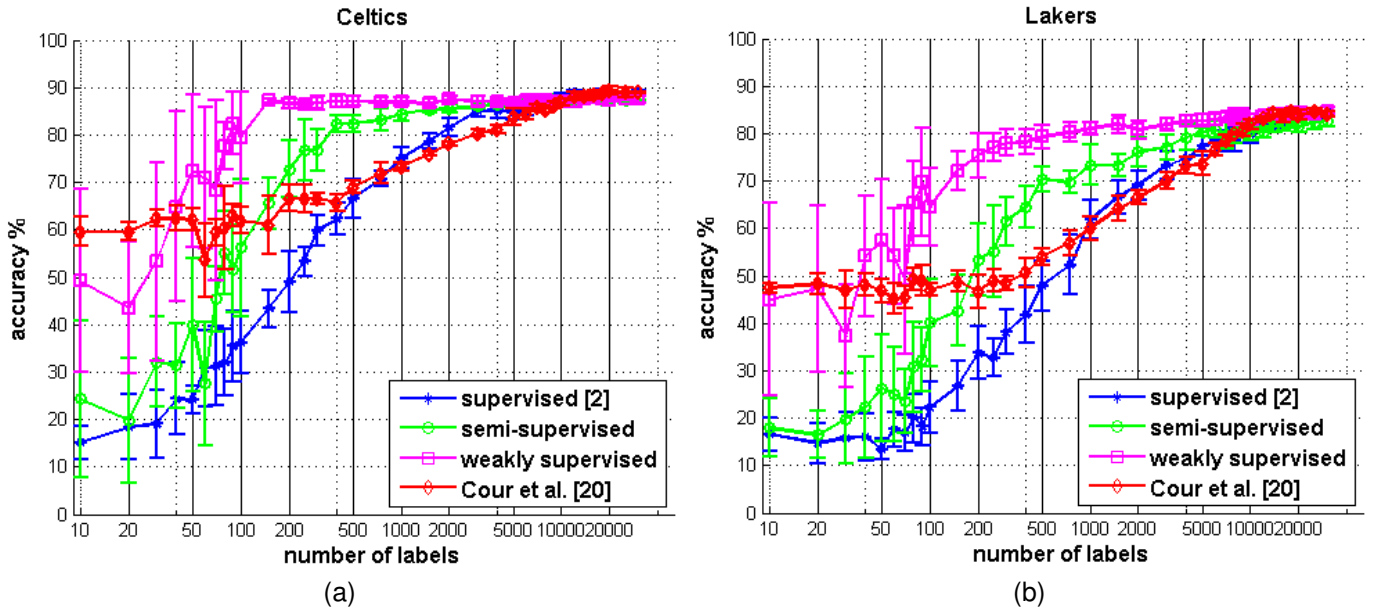


Fig. 10. Player identification results of different learning strategies. In both figures, we show identification results (mean and standard deviation) of strongly supervised, semi-supervised, weakly supervised learning, and Cour *et al.* [20], with different numbers of labeled training data. Notice that the x-axis is in log-scale. The *strongly supervised* approach uses only labeled training data in Equation 16. The *semi-supervised* approach uses both labeled and unlabeled data in Equation 17, but it uses an uniform prior over all possible players, *i.e.*, $p(\hat{y}_{t,d}) = \frac{1}{|\mathcal{C}|}$. The *weakly supervised* approach also uses both labeled and unlabeled data in Equation 17, but it uses the prior in Equation 19 provided by the play-by-play texts. (a) Identification results for Celtics. (b) Identification results for Lakers.

is because the proposed EM algorithm utilizes the temporal and mutual exclusion potentials to help deal with ambiguous images (e.g., profile views), while the ambiguous label learning classifies every image independently in the learning phase[4].

6.3.4 Comparison to the existing system [2]

Our previous work [2] reported a 85% identification accuracy for Lakers and 82% identification accuracy for Celtics players. Evaluating on the same dataset, the proposed system improves the identification accuracy to 87% for Lakers and 92% for Celtics. Since both systems use the same feature vectors, the boost is resulted from a better inference algorithm introduced in section 4.3. In addition, the proposed system also applies weakly supervised learning to greatly reduce the number of labelled training data required from 20000 to 200.

Figure 13 shows tracking and identification results on a basketball video. We see that the proposed system is able to track and identify multiple basketball players effectively. Please go to our website[5] for more results.

6.4 Homography evaluation

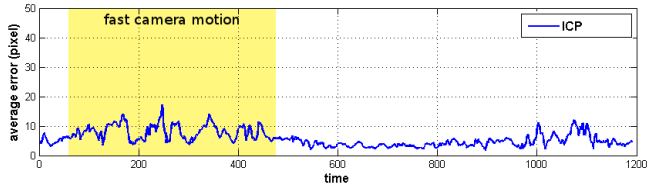
We measure the performance of homography estimation by computing the average distance between points transformed by the annotated homographies and points transformed by the estimated homographies. We test on 5969 frames with annotated homography.

The average error of homography is 7.32 pixels on 1280×720 images, or 13.65 cm on the basketball court ($23.65m \times 15.24m$), which is very small. Figure 11(a) shows the estimation errors of one selected test clip. We can see that errors are usually below 10 pixels, except the region between 200-th to 400-th frame, which have fast camera motion. Fast camera motions usually happen during offensive-defensive changes, and they cause significant motion blur that reduces the chance of detecting edges of the court.

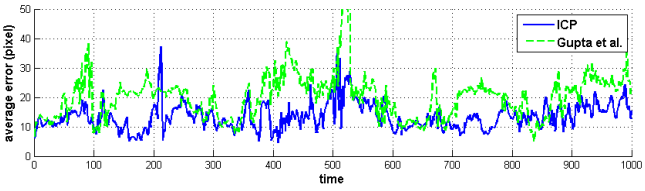
We compare our algorithm with [47] in their hockey dataset of 1000 frames, as shown in Figure 11(b). [47] uses a combination of SIFT matching and area minimization to estimate the homography. Their method has an average error of 20.21 pixels in the hockey video (green line). In comparison, the average error of ICP is only 13.35 pixels (blue line), which reduces the errors by 33%. Since the image resolution is 1920×980 pixels, the error is about 1.4% of the image size, which is similar to the basketball case. This demonstrates that the proposed algorithm is effective in different kinds of sports videos. The speed of ICP is also much faster (2 seconds per frame), while [47] runs in 60 seconds per frame.

4. [20] used tracking to reduce the number of training data, but they did not utilize temporal consistency in their learning algorithm.

5. http://www.cs.ubc.ca/~vailen/pami/

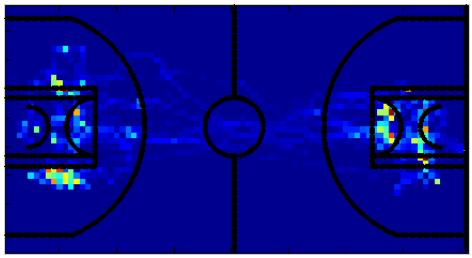


(a) basketball

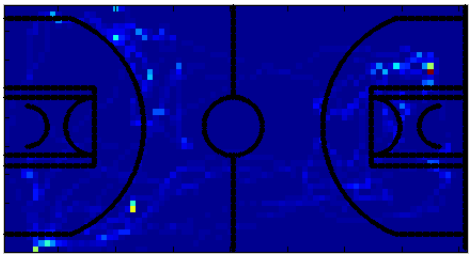


(b) hockey

Fig. 11. Homography estimation results in (a) basketball, and (b) hockey, compared to [47]. Errors are measured by the average distance between points transformed by the annotated and estimated homography.



(a) Bynum



(b) Vujacic

Fig. 12. The spatial histograms (heatmaps) of basketball players. Players offend in the left-hand side while defend in the right-hand side. In all images, lighter colors mean higher values in the histogram. (a) Bynum: a center whose job is to attack the basket from a short distance. (b) Vujacic: a 3-point shooter whose job is to shoot from a long distance.

6.5 Automatic collection of player statistics

Statistics are very important for analyzing the performance of sports players. However, these statistics are recorded by human annotators during the game, which is a very expensive and tedious process.

With the proposed tracking, identification, and homography estimation systems, it is possible to automatically generate player statistics from a single video. In Figure 12, we show the spatial histogram (heatmap) of a player's trajectory in the court coordinates. The heatmap is generated by tracking and identifying a specific player over 5000 frames, and project their foot locations from the image to the court coordinates. Specifically, we divide the court into grids of 1×1 meter. If the player stands on top of the grid, its histogram value will be increased by one. The heatmap represents the probability of a player being in each grid cell.

Since different players possess different motion patterns and playing styles, the heatmap might be used to identify players. We tried adding the heatmap as features to Equation 2, but we found that this had negligible effect on performance. One possible reason is that there are 12 players in a team, but only 3 distinctive playing styles and heatmaps. We anticipate that the heatmap would work better to identify players in sports such like soccer, where players have more diverse motion patterns and playing styles. Nevertheless, localizing players on the court is a useful output of the system, even if it does not help with player identification.

7 Discussion

We introduce a novel system that tackles the challenging problem of tracking and identification of players in broadcast sports videos taken from a single pan-tilt-zoom camera. The system possesses the ability to detect and track multiple players, estimates the homography between video frames and the court, and identifies the players. The identification problem is formulated as finding the maximum a posterior configuration in a Conditional Random Field (CRF). The CRF combines three weak visual cues, and exploits both temporal and mutual exclusion constraints. We also propose a Linear Programming Relaxation algorithm for predicting the best player identification in a video clip. For learning the identification system, we introduce the use of weakly supervised learning with the play-by-play texts. Experiments show that the identification system can achieve similar accuracies by using merely 200 labels in weakly supervised learning, while a strongly supervised approach needs a least 20000 labels.

Note that the identification system relies on the tracking system to construct the graphical model. When tracking results are unreliable, one may consider enabling the weak interactions in Equation 4 (set $\epsilon > 0$) in order to split unreliable tracks [2], or adding another weak interaction between the ends of two tracklets to encourage merging [10]. In both cases, max-product or sum-product BP can be used for the inference. Another possibility is to improve tracking algorithm itself by modelling the player's motion in the court coordinates instead of the image coordinates, as in [5]. This can be achieved by using the estimated homography to project players to the court coordinates.

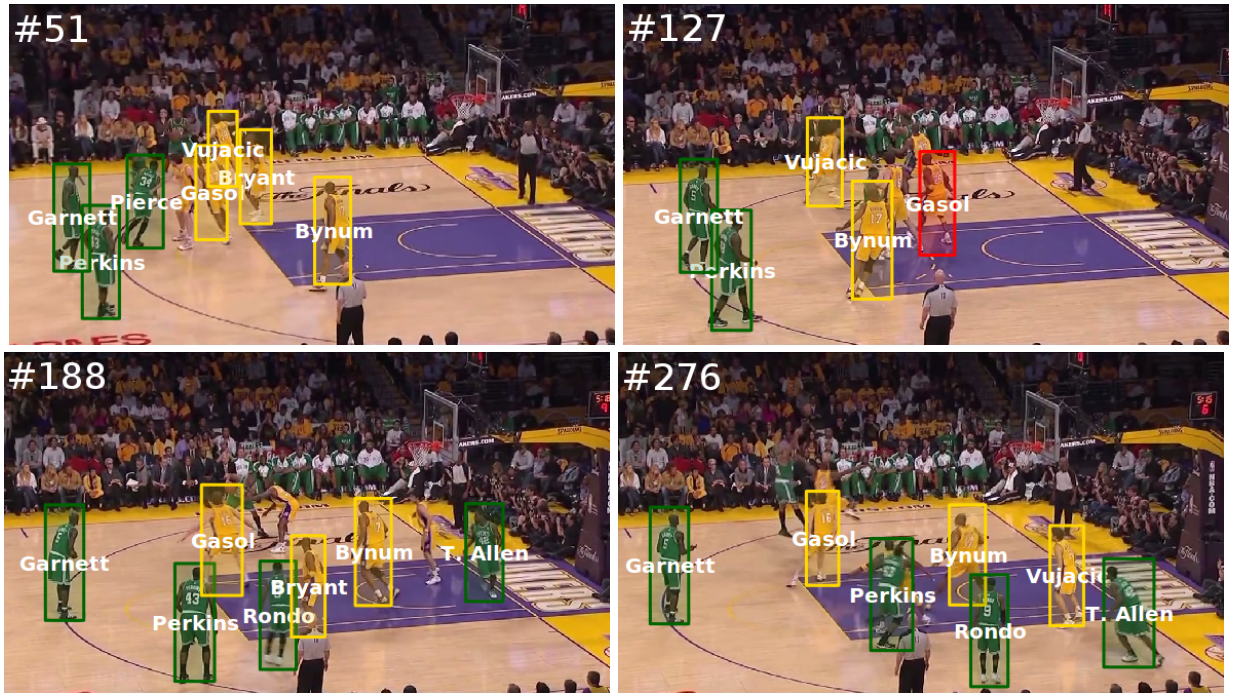


Fig. 13. Automatic tracking and identification results in a broadcast basketball video. Green boxes represent Celtics players, and yellow boxes represent Lakers players. Text within boxes are identification results (player's name), while red boxes highlight misclassifications.

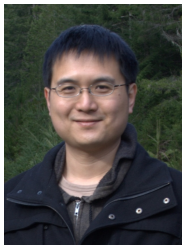
ACKNOWLEDGMENT

This work has been supported by grants from the the National Science and Engineering Research Council of Canada, GEOIDE Network of Centres of Excellence, and the Canadian Institute for Advanced Research. We also thank for the National Basketball Association Entertainment for kindly providing us the image dataset.

REFERENCES

[1] H. BenShitrit, J. Berclaz, F. Fleuret, and P. Fua, "Tracking Multiple People under Global Appearance Constraints," in *ICCV*, 2011.
[2] W.-L. Lu, J.-A. Ting, K. P. Murphy, and J. J. Little, "Identifying Players in Broadcast Sports Videos using Conditional Random Fields," in *CVPR*, 2011.
[3] A. Yilmaz and O. Javed, "Object Tracking: A Survey," *ACM Computing Surveys*, vol. 38, no. 4, p. No. 13, 2006.
[4] K. Okuma, A. Taleghani, N. de Freitas, J. J. Little, and D. G. Lowe, "A Boosted Particle Filter: Multitarget Detection and Tracking," in *ECCV*, 2004.
[5] Y. Cai, N. de Freitas, and J. J. Little, "Robust Visual Tracking for Multiple Targets," in *ECCV*, 2006.
[6] D. Comaniciu, V. Ramesh, and P. Meer, "Kernel-Based Object Tracking," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 25, no. 5, pp. 564–575, 2003.
[7] M.-C. Hu, M.-H. Chang, J.-L. Wu, and L. Chi, "Robust Camera Calibration and Player Tracking in Broadcast Basketball Video," *IEEE Transactions on Multimedia*, vol. 13, no. 2, pp. 266–279, 2011.
[8] B. Wu and R. Nevatia, "Detection and Tracking of Multiple, Partially Occluded Humans by Bayesian Combination of Edgelet based Part Detector," *International Journal of Computer Vision*, vol. 75, no. 2, pp. 247–266, 2007.
[9] B. Song, T.-Y. Jeng, E. Staudt, and A. K. Roy-Chowdhury, "A Stochastic Graph Evolution Framework for Robust Multi-Target Tracking," in *ECCV*, 2010.
[10] B. Yang, C. Huang, and R. Nevatia, "Learning Afnities and Dependencies for Multi-Target Tracking using a CRF Model," in *CVPR*, 2011.
[11] H. Jiang, S. Fels, and J. J. Little, "Optimizing Multiple Object Tracking and Best Biew Video Synthesis," *IEEE Transactions on Multimedia*, vol. 10, no. 6, pp. 997–1012, 2008.
[12] J. Liu, X. Tong, W. Li, T. Wang, Y. Zhang, and H. Wang, "Automatic player detection, labeling and tracking in broadcast soccer video," *Pattern Recognition Letters*, vol. 30, pp. 103–113, 2009.
[13] L. Ballan, M. Bertini, A. D. Bimbo, and W. Nunziati, "Soccer players identification based on visual local features," in *CIVR*, 2007.
[14] M. Bertini, A. D. Bimbo, and W. Nunziati, "Player Identification in Soccer Videos," in *MIR*, 2005.
[15] M. Saric, H. Dujmic, V. Papic, and N. Rozic, "Player Number Localization and Recognition in Soccer Video using HSV Color Space and Internal Contours," in *ICSIP*, 2008.
[16] Q. Ye, Q. Huang, S. Jiang, Y. Liu, and W. Gao, "Jersey number detection in sports video for athlete identification," in *SPIE*, 2005.
[17] X. Zhu and A. B. Goldberg, *Introduction to Semi-Supervised Learning*. Morgan & Claypool, 2009.
[18] C. Vondrick, D. Ramanan, and D. Patterson, "Efficiently Scaling Up Video Annotation with Crowdsourced Marketplaces," in *ECCV*, 2010.
[19] K. Barnard, P. Duygulu, D. Forsyth, N. de Freitas, D. M. Blei, and M. I. Jordan, "Matching Words and Pictures," *Journal of Machine Learning Research*, vol. 3, pp. 1107–1135, 2003.
[20] T. Cour, B. Sapp, C. Jordan, and B. Taskar, "Learning from Ambiguously Labeled Images," in *CVPR*, 2009.
[21] T. Cour, B. Sapp, A. Nagle, and B. Taskar, "Talking Pictures: Temporal Grouping and Dialog-Supervised Person Recognition," in *CVPR*, 2010.
[22] M. Everingham, J. Sivic, and A. Zisserman, ""Hello! My name is... Buffy" - Automatic Naming of Characters in TV Video," in *BMVC*, 2006.
[23] J. Sivic, M. Everingham, and A. Zisserman, ""Who are you" - Learning person specific classifiers from video," in *CVPR*, 2009.
[24] O. Duchenne, I. Laptev, J. Sivic, F. Bach, and J. Ponce, "Automatic Annotation of Human Actions in Video," in *ICCV*, 2009.
[25] I. Laptev, M. Marszalek, C. Schmid, and B. Rozenfeld, "Learning realistic human actions from movies," in *CVPR*, 2008.
[26] M. Marszalek, I. Laptev, and C. Schmid, "Actions in Context," in *CVPR*, 2009.

[27] A. Gupta, P. Srinivasan, J. Shi, and L. S. Davis, "Understanding Videos, Constructing Plots: Learning a Visually Grounded Storyline Model from Annotated Videos," in *CVPR*, 2009.
[28] G. Zhu, C. Xu, Q. Huang, Y. Rui, S. Jiang, W. Gao, and H. Yao, "Event Tactic Analysis Based on Broadcast Sports Video," *IEEE Transactions on Multimedia*, vol. 11, no. 1, pp. 49–66, 2009.
[29] P. Felzenszwalb, D. McAllester, and D. Ramanan, "A Discriminatively Trained, Multiscale, Deformable Part Model," in *CVPR*, 2008.
[30] L. Rabiner, "A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition," *Proceedings of the IEEE*, vol. 77, no. 2, pp. 257–286, 1989.
[31] Tesseract-OCR, "http://code.google.com/p/tesseract-ocr/."
[32] C. M. Bishop, *Pattern Recognition and Machine Learning*. Springer, 2006.
[33] K. Okuma, "Active exploration of training data for improved object detection," Ph.D. dissertation, University of British Columbia, 2012.
[34] R. Kalman, "A new approach to linear filtering and prediction problems," *Journal of Basic Engineering*, vol. 82, no. 1, pp. 35–45, 1960.
[35] J. Matas, O. Chum, M. Urban, and T. Pajdla, "Robust Wide Baseline Stereo from Maximally Stable Extremal Regions," in *BMVC*, 2002.
[36] D. G. Lowe, "Distinctive Image Features from Scale-Invariant Keypoints," *International Journal of Computer Vision*, vol. 60, no. 2, pp. 91–110, 2004.
[37] P.-E. Forssen and D. G. Lowe, "Shape Descriptors for Maximally Stable Extremal Regions," in *ICCV*, 2007.
[38] J. Nocedal and S. J. Wright, *Numerical Optimization*. Springer, 1999.
[39] B. Frey and D. Mackay, "A revolution: Belief propagation in graphs with cycles," in *NIPS*, 1998.
[40] A. Y. Ng, "Feature selection, L1 vs. L2 regularization, and rotational invariance," in *NIPS*, 2004.
[41] M. Schmidt, E. van den Berg, M. P. Friedlander, and K. P. Murphy, "Optimizing Costly Functions with Simple Constraints: A Limited-Memory Projected Quasi-Newton Algorithm," in *AISTATS*, 2009.
[42] A. Dempster, N. Laird, and D. Rubin, "Maximum Likelihood from Incomplete Data via the EM Algorithm," *Journal of the Royal Statistical Society. Series B*, vol. 39, no. 1, pp. 1–38, 1977.
[43] R. Y. Tsai, "A versatile camera calibration technique for high-accuracy 3D machine vision metrology using off-the-shelf TV cameras and lenses," *IEEE Journal of Robotics and Automation*, vol. 3, no. 4, pp. 323–344, 1987.
[44] Z. Zhang, "A Flexible New Technique for Camera Calibration," *IEEE Transactions on Pattern Analysis and Machine Inlligence*, vol. 22, no. 11, pp. 1330–1334, 2000.
[45] K. Okuma, J. J. Little, and D. G. Lowe, "Automatic Rectification of Long Image Sequences," in *ACCV*, 2004.
[46] R. Hess and A. Fern, "Improved Video Registration using Non-Distinctive Local Image Features," in *CVPR*, 2007.
[47] A. Gupta, J. J. Little, and R. J. Woodham, "Using Line and Ellipse Features for Rectification of Broadcast Hockey Video," in *CRV*, 2011.
[48] J. Canny, "A Computational Approach to Edge Detection," *IEEE Transactions on Pattern Analysis and Machine Inlligence*, vol. 8, no. 6, pp. 679–698, 1986.
[49] Z. Zhang, "Iterative Point Matching for Registration of Free-form Curves and Surfaces," *International Journal of Computer Vision*, vol. 13, no. 2, pp. 119–152, 1994.
[50] S. M. Tari, "Automatic initialization for broadcast sports videos rectification," Master's thesis, University of British Columbia, 2012.
[51] R. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*, 2nd ed. Cambridge University Press, 2003.



Wei-Lwun Lu is an Algorithm Developer in Ikomed Technologies. He received a Ph.D. in Computer Science from the University of British Columbia in 2012, a M.Sc. in Computer Science from the University of British Columbia in 2007, and a B.S. in Computer Science from National Tsing Hua University, Taiwan, in 2002. His primary research interests are computer vision and machine learning, especially in the problems of video analysis, geospatial data processing, and medical imaging.

Jo-Anne Ting is a Research Engineer at Bosch Research. She received a Ph.D. in Computer Science from the University of Southern California in 2009 and a BASc in Computer Engineering from the University of Waterloo in 2003. She was a Research Fellow in the School of Informatics at the University of Edinburgh and an NSERC Postdoctoral Fellow at the University of British Columbia.

James J. Little is a Professor in Computer Science at The University of British Columbia. He received his Ph.D. from University of British Columbia in 1985. He has been a research analyst at the Harvard Laboratory for Computer Graphics and Spatial Analysis, a research associate at Simon Fraser University, and a research scientist at the MIT Artificial Intelligence Laboratory. His research interests include early vision, understanding image sequences, surface representation, and visually-guided mobile robotics, where his research includeds mapping, navigation, and object recognition.

Kevin Patrick Murphy was born in Ireland, grew up in England, went to graduate school in the USA (MEng from U. Penn, PhD from UC Berkeley, Postdoc at MIT), and then became a professor at the Computer Science and Statistics Departments at the University of British Columbia in Vancouver, Canada in 2004. After getting tenure, Kevin went to Google in Mountain View, California for his sabbatical. In 2011, he converted to a full-time research scientist at Google. Kevin has published over 50 papers in refereed conferences and journals related to machine learning and graphical models. He has recently published an 1100-page textbook called "Machine Learning: a Probabilistic Perspective" (MIT Press, 2012).