Person Count Localization in Videos from Noisy Foreground and Detections

Sheng Chen, Alan Fern and Sinisa Todorovic
Oregon State University
{chenshen, afern, sinisa}@eecs.oregonstate.edu

Abstract

This paper formulates and presents a solution to a new problem called person count localization. Given a video of a crowded scene, our goal is to output for each frame a set of: 1) Detections optimally covering both isolated individuals and cluttered groups of people; and 2) Counts of people inside these detections. This problem is a middle-ground between frame-level person counting, which does not localize counts, and person detection aimed at perfectly localizing people with count-one detections. Our problem formulation is important for a wide range of domains, where people appear frequently under severe occlusion within a crowd. As these crowds are often visually distinct from the rest of the scene, they can be viewed as "visual phrases" whose spatially tight localization and count assignment could facilitate higher-level video understanding. For count localization, we specify a novel framework of iterative error-driven revisions of a flow graph derived from noisy input of people detections and foreground segmentation. Each iteration creates and solves an integer program for count localization based on iterative revisions of the flow graph. The graph revisions are based on detected violations of basic integrity constraints. They in turn trigger learned modifications to the graph aimed at reducing noise in input features. For evaluation, we introduce a new metric that measures both count precision and localization of our approach on American football and pedestrian videos.

1. Introduction

Motivation. In this paper, we consider the problem of detecting people in videos of crowded scenes, where people frequently appear under severe occlusion by other people in the crowd. This is an important line of research, since detecting people in video frames has become the standard initial step of many approaches to activity recognition [1, 16, 10, 2, 7, 4, 23], and multi-object tracking by detection [19, 25, 22, 12, 3, 24, 11]. They typically use as input human appearance, pose, and orientation, and thus critically depend on robust person detections. In many domains, however, such as videos of American football (Fig.1) or public spaces crowded with pedestrians, detecting every individual person is highly unreliable, and remains an open problem.

This motivates us to study alternative formulations that do not require perfect person localization, especially under severe clutter and occlusion, and still prove useful for higher-level video understanding. One related problem that has successfully addressed videos of crowded scenes is frame-level counting of people [5, 20, 17, 6, 14]. This frame-level count information, however, is a very coarse description of the video, with limited utility for high-level tasks. In particular, these problem formulations and evaluations do not address location of individuals or sub-groups.

Rather than counting people per frame, we would like to retain as much localization capability of detecting individuals as possible, but gracefully transition to counting people within areas in the frame occupied by crowds. As these crowded groups are often isolated and visually distinct from the rest of the scene, they can be viewed as "visual phrases" whose spatially tight localization and count assignment could provide useful cues for higher-level processing. For example, as shown in Fig.1, localized counts provide rich information about the activity unfolding during a football play by identifying many isolated and small groups of players and the primary larger player groups. Similarly, localized counts can provide space-time density statistics of crowds in an area of interest and also serve as a basis for more refined individual tracking when desired.

New Problem. In this paper, we introduce a new problem, *person count localization* from noisy foreground and person detections. Our formulation strikes a middle-ground between person detection and frame-level counting. Given a video, our goal is to output for each frame a set of:

1. Detections optimally covering both isolated individuals and crowds of people in the video; and
2. Counts assigned to each detection indicating the number of people inside.

Overview of Approach and Contributions. Our approach first extracts noisy foreground by running a person detector and foreground segmentation, which will sometimes be redundant. A flow graph is then built and trans-

Figure 1: Our count localization results for an image sequence from American football. Challenges include severe occlusion, clutter, and similar appearance of players.

formed into a integer linear program (ILP). The construction of this ILP is our first contribution as it must deal with the redundancy and false positives in input.

Our second contribution is to improve the initial solution via the new framework of iterative error-driven graph revision (EGR). The key idea is that the ILP is derived from a flow graph whose structure is based on hard-to-tune parameters and noisy input. As a consequence, for any fixed graph construction approach, there will be cases where the ILP solution can be observed to have visually obvious errors that violate basic integrity constraints (e.g. a disappearing person in the middle of a scene). To address this issue, EGR iteratively constructs a sequence of such graphs, and the corresponding integer programs, such that each new flow graph is a refinement of the previous graph, and aimed at correcting violated integrity constraints of the previous solution. We use a simple learning strategy to select among various refinements available at each step, such as running a tracker to produce new detections in an area or adding edges to the graph. The EGR process stops when no further integrity constraints are found or refinements are selected and in our experiments produce significantly improved results.

Our third contribution is to introduce a new metric for person count localization that accounts for both errors in localization and counting. We provide experiments in two challenging domains: American football and pedestrian crowds. The results demonstrate the benefits of our approach compared to prior work and a number of baselines. Also, our evaluation suggests that EGR is a promising framework for improving other vision tasks based on fixed compilations to optimization problems.

2. Prior Work

While we are the first to define the problem of person count localization and associated evaluation metrics, prior work has studied related problems.

In multi-object tracking, counting problems are sometimes solved as part of the overall approach to deal with groups of people. [19] greedily assigns and propagates counts, which can lead to inconsistent counts. Instead, [11] formulate counting as a flow problem but uses a simple linear cost function based on detection size, which is not appropriate when there is heavy occlusion or the number of counts vary significantly. Similar to [11], we also formulate our count localization problem in a flow framework but we have a more complex objective function and constraints and place an additional focus on being robust to segmentation/detection noise. The most similar work to ours was on the problem of biological cell tracking, where maintaining cell counts in foreground blobs is the main step followed by heuristic id maintenance [21]. They solved the problem by formulating an integer program based on a flow graph over foreground blobs. Their approach, however, is insufficient for our problems, as our experiments show. The key issue is that contrary to our primary motivating application of team sports, their cell tracking application allowed for high quality non-intersecting foreground blobs with few false positives and negatives, which significantly simplifies the problem. In our application of American football, foreground extraction is often quite noisy, producing high rates of both false positives and negatives.

The crowd counting problem generally has the goal of accurately counting the number of people in a frame. Prior work [5, 20, 17, 6, 14, 13] typically follows a two-step pipeline where foreground segments are first extracted and then counts are independently estimated for each segment based on local features. Research has focused mainly on feature design and training reliable estimators, but has mostly ignored the consistency and interactions between segments. While some of these methods are able to generate counts for more localized segments, the evaluations have all focused on the accuracy of total counts, which ignores localization performance.

3. Approach

We now describe our approach to the *person count localization problem*, where the input is a video and the output is a set of detections, each labeled by a count of people. The quality measure of the output is based on both the accuracy of the counts assigned to detections and the localization of the detections. Intuitively, we desire count-one detections to be associated with individual people when possible, but for cluttered crowds we are satisfied with a crowd-level detection and accurate count. In our experiments, we introduce a new evaluation metric to capture these concerns.

3.1. Overview

Our iterative solution approach, *error-driven graph revision (EGR)*, is depicted in Fig.2. In the first iteration we extract foreground objects/blobs from the video and build a corresponding initial flow graph representation G_0 that represents the temporal-spatial relationships among the foreground objects. An integer linear program (ILP) is then formulated based on G_0 that both selects a subset of detections and assigns counts to them, giving a solution denoted by C_0. The ILP is designed with the goal of maintaining accurate counts that also maintain temporal-spatial consistency.

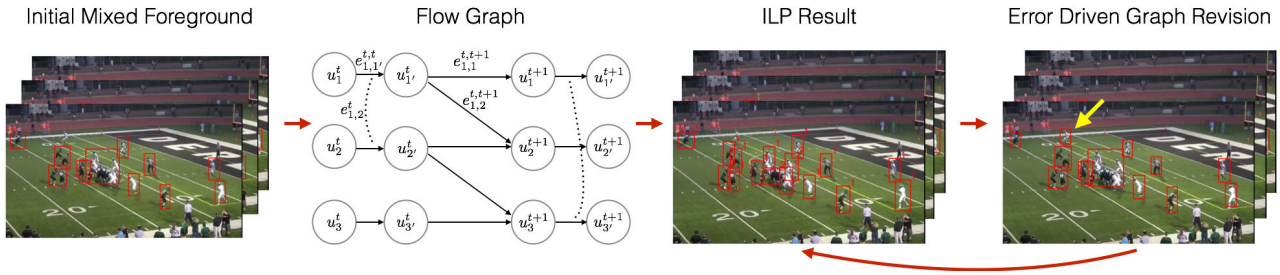

Figure 2: System Overview. First extract noisy foreground by running an object detector and foreground segmentation. A flow graph is then built and transformed into an integer program. In the following iterations, the approach detects places where integrity/domain constraints are violated by the current solution and applies one or more graph-revision operators to obtain a new graph and updated solution.

At iteration i of EGR, we first look at the ILP solution C_{i-1} from the previous iteration in order to identify violations of common-sense integrity and domain constraints (for example a person cannot appear or disappear in the middle of the frame). Such violations are inevitable in our experience for any fixed way of constructing graphs from the input. Associated with each type of constraint violation are potential graph-revisions operations that may address the violation, e.g. adding edges, adding nodes, etc. A trained classifier is then used to select appropriate graph revisions to G_{i-1} that yields G_i, resulting in a new ILP and solution C_i. The iteration ends when no constraint violations are detected or a maximum number of iterations.

Note that, we do not assume the initial extracted foreground objects to be perfect. In fact, the iterative process aims at dealing with this noisy input. As we will see later, the ILP can help address the problem of false positive foreground objects by not selecting them or assigning them counts of zero. However, the ILP does not have a natural way to deal with false negatives, which do not even appear in the corresponding flow graph. The key idea behind the EGR approach is that the ILP solutions in such cases will often violate common-sense integrity constraints that can be easily checked. Further, for a detected violation, there are natural ways to revise the graph that will potentially correct the violation, for example, by using a tracking mechanism to acquire detections in a certain space-time region of the video that were missed by the initial processing.

We note that an alternative to EGR would be to construct a single graph G^* and ILP that accounts for all possible missing detections and edges. This, however, is impractical for at least two reasons. First, the enormous number of such possibilities would stress even state-of-the-art IP solvers. Second, the number of false-positives represented in the graph would grow dramatically, resulting in less reliable solutions due to the increase in ambiguity. Rather, EGR can be viewed as an approach that aims to incrementally construct a graph G containing only the "necessary" parts of G^* for a particular problem instance.

3.2. Foreground Detection and Graph Building

Given a video, we first need to extract foreground detections that will serve as candidate detections to be labeled by counts. In contrast to previous work that either runs an object detector or performs foreground segmentation to obtain the foreground detections, we apply **both** a person detector and foreground segmentation. As shown in Fig.3, these two methods have their own strengths. A person detector usually works well for single isolated people, but has problems when there is occlusion or inside a crowd, while foreground segmentation usually works well when there is a clutter of people but performs poorly for smaller object due to noisy background modeling and registration. We combine the two methods by using both the person detector's results and the relatively larger connected components from the foreground segmentation. In this way, we can get an initial set of foreground detections with a reasonable recall. Note that there will often be significant overlap between the foreground and detections, which is a complication that the optimization process must account for.

We denote all the foreground detections obtained from the two methods by $\{d_i^t\}$ where t is the frame index. We then build a flow graph $G = (V, E \cup E')$ as shown in Fig.2. Each foreground detection is represented by two vertices u_i^t and $u_{i'}^t$. There are two types of edges in G. The solid edge set E contains edges $e_{i,i'}^{t,t}$ and $e_{i,j}^{t,t+1}$ where $e_{i,i'}^{t,t}$ links u_i^t to $u_{i'}^t$. $e_{i,j}^{t,t+1}$ link $u_{i'}^t$ to a subset of u_j^{t+1} in the next frame. The linkage is determined by the following rules: we first link foreground detections that form reliable tracklets as suggested by [15]. For other foreground detections, $e_{i,j}^{t,t+1}$ is added if d_j^{t+1} is in the neighborhood of d_i^t according to a threshold. Note that, a particular threshold may not work for all cases because of different viewpoints and perspectives. Our approach will adaptively change this threshold more locally in later EGR stage if needed. The other set of dashed edges E' are hyper edges and link a subset of $e_{i,i'}^{t,t}$ in the same frame. Since our foreground detections can overlap, we add a hyper edge $e_{i,j}^{t}$ between $e_{i,i'}^{t,t}$ and $e_{j,j'}^{t,t}$ if for

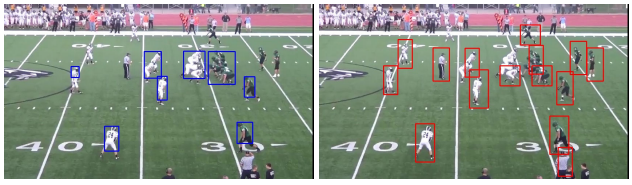

Figure 3: Example results of object detector and foreground segmentation. Left: foreground segmentation, Right: DPM detector. The segmentation misses several isolated small players while the DPM misses players in the crowd.

d_i^t and d_j^t, one is covered by the other or their intersection of union score is larger than a threshold.

3.3. Integer Programming Formulation

We now wish to convert the flow graph to an optimization problem that when solved will assign counts to the detections represented in the graph in a way that maximizes the estimated count accuracy while satisfying basic flow constraints. Given the above graph G, consider assigning each edge in E a corresponding variable x indicating the amount of flow (number of people) going through that edge, for example, $x_{i,i'}^{t,t}$ is a variable indicating the flow across edge $e_{i,i'}^{t,t}$. Note that the flow assigned to $x_{i,i'}^{t,t}$ is interpreted as the count of people assigned to detection d_i^t. Given these variables we would like to find flow values (equivalently count values) that result in consistent flows and also maximize some measure of count accuracy for each detection.

In order to measure count accuracy we use a function $f_{i,i'}^{t,t}(x_{i,i'}^{t,t})$ that assigns an accuracy score to the count assigned to d_i^t. In traditional network flow formulations, these functions are linear in $x_{i,i'}^{t,t}$ and in that case yields a polynomial time algorithm. However, for our problem it is unlikely that any linear function will approximate f well, since the accuracy of a count assignment is going to non-trivially depend on the visual evidence associated with detection d_i^t.

In this work, we define f values based on the confidence of learned random forest classifiers. In particular, given an upper bound N on the maximum count that can be assigned to a detection, we train a random forest based on labeled training data to predict the discrete count $\{0, \ldots, N\}$ for a detection d_i^t given visual features of that detection. The value of $f_{i,i'}^{t,t}(x_{i,i'}^{t,t})$ is then taken to be the confidence that d_i^t should be assigned count $x_{i,i'}^{t,t}$. The prediction of the random forest is based on the following features: detection type (whether the detection is from a person detector or foreground), location, size, number of foreground pixels, and the spatial distribution of foreground pixels. We note that more sophisticated regression algorithms that provide confidences could also be used to define f.

Given this definition of f we would like to find the flow assignment $\mathbf{x}$ that maximizes the total estimated count accuracy $\sum_{i,t} f_{i,i'}^{t,t}(x_{i,i'}^{t,t})$ subject to standard flow conservation constraints that make sure the counts are consistent across frames. Further, we also want to enforce constraints corresponding to the hyper edges in G, which state that we only want to assign non-zero counts to one detection in a pair of overlapping detections. Unfortunately, in contrast to standard network flow formulations where f is linear, when f is relatively arbitrary as in our case, the problem of optimizing the count accuracy objective is NP-complete. This means that we are unlikely to find an efficient exact algorithm. However, below we formulate this problem in terms of an integer linear program (ILP), which allows us to apply state-of-the-art ILP solvers to our problem.

To formulate our problem as an ILP we introduce indicators for each flow variable to linearize the objective. We denote $x_{i,i',n}^{t,t}$ to be the indicator of flow variable $x_{i,i'}^{t,t}$ taking value n and similarly for $x_{i,j}^{t,t+1}$. We also define $c_{i,i',n}^{t}$ to be the accuracy score of f for assigning detection d_i^t a count of n. The ILP can now be defined as follows:

$$
\begin{aligned}
\max_{\mathbf{x}} \quad & \sum_{i,t} \sum_{n} c_{i,i',n}^{t} x_{i,i',n}^{t,t} \\
\text{s.t.} \quad & \text{for all i, t} \\
& \sum_{n} x_{i,i',n}^{t,t} \leq 1, \sum_{n} x_{i,j,n}^{t,t+1} \leq 1 \text{ for } e_{i,j}^{t,t+1} \in E, \quad \text{(a)} \\
& \sum_{n} n x_{i,i',n}^{t,t} = \sum_{j:e_{i,j}^{t,t+1} \in E} \sum_{n} n x_{i,j,n}^{t,t+1}, \quad \text{(b)} \\
& \sum_{n} n x_{i,i',n}^{t,t} = \sum_{j:e_{j,i}^{t-1,t} \in E} \sum_{n} n x_{j,i,n}^{t-1,t}, \quad \text{(c)} \\
& \sum_{n} x_{i,i',n}^{t,t} + \sum_{n} x_{j,j',n}^{t} \leq 1 \text{ for } e_{i,j}^{t} \in E' \quad \text{(d)} \\
& 0 \leq \mathbf{x}, \mathbf{x} \in \mathbb{I} \quad \text{(e)}
\end{aligned}
\tag{1}
$$

The set of constraints (a) make sure all indicators for one edge sum up to less or equal to 1. The flow conservation constraints correspond to (b) and (c) and the hyper edge constraints correspond to (d).

With the hyper edge constraints, not all input detections will be assigned a count in the ILP solution. When there is a large group with a large detection from segmentation and a few smaller, overlapping detections from the person detector, the large detection will usually be chosen as it better facilitates the flow constraints. However, when foreground extraction is noisy, sometimes these large detections can contain significant areas that do not contain people and also people that can be localized by people detectors. In such cases, we could get improved localization by also using the overlapping smaller detections. To account for this, after solving the ILP, we perform an additional optimization at the detection level for any detection d that contains smaller detections. In particular, given such a detection d with a

count value of c from the ILP solution, we wish to best assign counts to the smaller detections in order to provide better localization within d. We use a greedy optimization for this and greedily assign counts to the small detections in order to maximize their f scores with the constraint that the total of the counts does not exceed c. After doing this, if the total count c' assigned to the smaller detections is less than c we assign d a count of $c - c'$ indicating that the remaining people are somewhere in d but not precisely localized.

3.4. Error-driven Graph Revision

The initial set of detections we get are noisy and there can be both false positives and missing detections. The ILP attempts to address the problem of false positives by allowing for counts of zero to be assigned to any detection. However, the ILP has no way of dealing with missing detections. In addition, as we mentioned above, the ILP relies on the graph G which is built based on certain thresholds, which are hard to define so as to work well in all situations. It is thus, desirable to be able to adjust the thresholds locally if the need is detected. In order to deal with these problems, we introduce the iterative EGR framework.

To apply EGR one must specify integrity constraints that hold for (nearly) all solutions. The constraints can come from common sense or domain-specific knowledge. In our case, we use the simple constraint that people cannot appear/disappear at non entry/exit locations. In our current domains, this single constraint was sufficient to allow EGR to significantly improve performance. Note that such domain constraints are often hard to directly impose in the ILP while retaining non-trivial solutions due to missing detections. However, they are easy to check given an ILP solution. In our case, we simply look for foreground detections that have non-zero counts assigned and have no successor in the next frame or predecessor in the previous frame in the graph. We denote these detections by $\{d_{e_i}\}$.

Next, we need to update these places. As we mentioned, there are two error sources, one is missing detections and the other is inappropriate thresholds used in graph construction. We propose three operators to correct these errors.

Add a node (Fig.4 top). This operator applies when we have a small gap of missing detection. If we decide to apply this operator at d_{e_i}, we will create a new detection that is identical to d_{e_i} in the next (previous) frame, and then modify the location according to a constant velocity model.

Add a tracker (Fig.4 middle). When we are missing foreground detections for multiple frames, we fire an object tracker at d_{e_i} to track the target forward (backward). We stop tracking when the tracker is not confident or the tracking result overlaps with existing foreground detections. We then add all the tracking results to the graph. The idea is that the tracker behaves as a localized detector for d_{e_i} that can overcome mistakes made by the more general detectors.

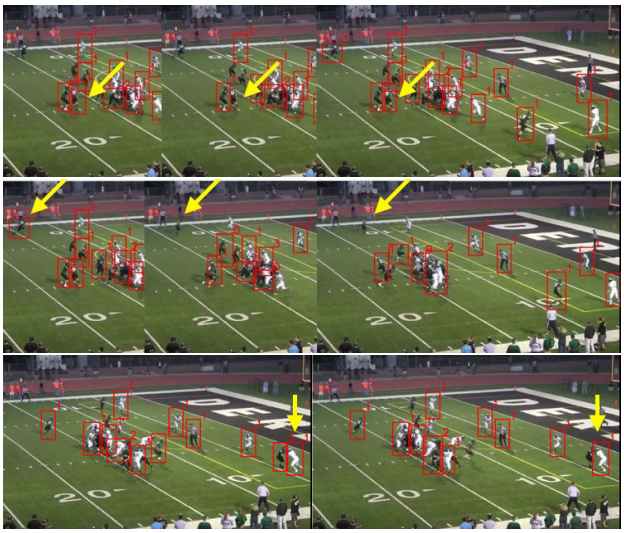

Figure 4: Illustration of three operators. Top: Add a node. Miss a detection in one frame while there are corresponding detections in neighboring frames. Middle: Add a tracker. A target continue missing for several frames. Bottom: Add an edge. This should be a merge, but we did not connect initially because of inappropriate threshold.

Add an edge (Fig.4 bottom). When there is foreground detections around d_{e_i} in the next (previous) frame, we might just lower the threshold of the graph construction and add an edge between d_{e_i} and some existing foreground detections.

It is not straightforward to decide which operator to apply, especially for adding a node and adding an edge. So we train a random forest classifier to mimic the choices made by an experienced human in various situations. The features for the classifier include distance to the existing closest detection in the next (previous) 1 frame and 5 frames, size of uncovered foreground pixels in the neighborhood in the next (previous) 1 frame and 5 frames, and sum of optical flow magnitude within the detection. Training examples were generated by creating ILP solutions on a training set, finding integrity constraint violations and then have the human label them by the most appropriate operator.

After making these local updates to the graph we create a new ILP and rerun the solver initialized with the previous solution. We iterate over solving the ILP and updating the graph until there is no error detected in the solution or a certain number of iterations is reached.

4. Experimental Results

Datasets. We evaluate on two datasets from the domain of American football and a pedestrian domain. The football dataset contains 10 videos from a football game where each video depicts a complete play ranging from 200 to 400 frames with resolution 852×480. The challenges here in-

clude large view-point variations, fast camera motions and complex player interactions. Ground truth bounding boxes are labeled for all 22 players and 1 defensive referee. The pedestrian dataset is taken from [5] and depicts pedestrians walking in two directions along a sometimes crowded walkway. The camera is stationary and the video contains 2000 frames with a resolution of 238×158. The ground truth location for each pedestrian is also provided for evaluation. The challenge here is that the density of people is high and there are seldom isolated people. Compared to the football domain, however, the foreground segments provided are much less noisy.

Implementation Details. For football, the foreground segmentation is done by automatically registering each frame to a panorama of the football field and then doing background subtraction to obtain foreground blobs. We also used training videos from the same game to train a DPM detector [9] to recognize players. The count prediction model required to produce scores for our ILP is obtained by training a random forest classifier on 4 videos with different view points and then testing on the remaining videos. The precision for the initial input foreground detections is 83.36%. 1.2 players out of 23 are not covered by any initial foreground detection in each frame on average. For the pedestrian dataset [5], we use the same foreground segmentation from [5] and train a Haar detector to detect individual people. We follow the same training strategy as [5] and use frames from 600 to 1399 to train the random forest classifier. The single object tracker used in our EGR framework for both datasets is from [26]. We use the Gurobi ILP solver and perform a maximum of 5 EGR iterations.

Baselines. For the football dataset, we compare with several variants of our approach: 1) EGR, our full count localization approach, 2) RF, we use the trained random forest to assign count to every input detection independently without enforcing flow constraints, 3) EGR_i, our approach run for i iterations, in particular, EGR_1 is the results of our ILP with the initial input, 4) EGR_{n^-,e^-,t^-}, our approach without applying one of the operators (adding nodes, edges, trackers) respectively, 5) EGR^*, our approach applied to the ground truth foreground, i.e. each input detection is a connected component in the ground truth foreground, 6) [21], we implement the method described in [21] and made it work for the football domain. 7) [21]*, [21] applied to the ground truth foreground. For the pedestrian dataset [5], we compare against prior state-of-the-art results.

4.1. Evaluation Metrics

There are no existing metrics designed to measure the performance of count localization. Prior work such as [21] did not measure the count performance explicitly but rather the overall event-recognition system. Crowd counting work such as [5, 18] focus on global count accuracy without regard for localization. Thus we propose a new metric called *count localization accuracy (CLA)* that is aimed to evaluate both count and localization accuracy.

Suppose for one frame, we have n ground truth people and produce a solution with m detections each with a count c_i. To calculate the metric, we first greedily match ground truths to detection results. If the intersection over union score (IoU) between a ground truth and a detection is over a threshold, we call this a match candidate, and each ground truth is matched to only one detection with the highest IoU score. A detection with count c cannot be matched to more than c ground truths. After the matching is done, for each ground truth, we calculate the IoU score s_i between the ground truth and its corresponding detection (0 if there is no match). The metric is then calculated as: $CLA = \sum_{0 \le i}^{n} s_i / \left(\sum_{0 \le i}^{m} c_i + n' \right)$, where n' is the number of unmatched ground truths. In the ideal case, where our results are the same n detections as ground truth and each with count 1, the metric gives a score of one. This metric evaluates both counting precision and localization. In one extreme case where we have all the correct detections but wrong counts, the denominator gives a penalty. In another extreme case where we have a single large detection that covers all ground truths with the correct count, the numerator which evaluates the IoU gives a penalty.

In addition to CLA we also report some other common metrics. *Localization Accuracy (LA)* keeps the same numerator as CLA and uses ground truth counts as the denominator and thus only accounts for localization accuracy. *Missing count (MC)* calculates the percentage of the ground truth that are miss counted. *Count error (CE)* calculate the average absolute count error. For the football dataset, we compute CE based on each output detection and denote it as CE_d. For the pedestrian dataset, CE is computed based on the entire frame to be able to compare with previous results and is denoted as CE_f.

4.2. Results

Quantitative Results: Tab.1 shows results for the football dataset. First, we observe that our approach EGR outperform [21] since [21] has no way to deal with missing detections in the noisy input. In fact, [21] performs similar to our approach in the first pass, i.e. without any graph revision and working with initial input. Further, when applying to the ground truth foreground, [21] has the same performance as ours. It shows that under ideal input, even if [21] has a more complex objective (transition score), we are able to achieve the same results. As a reference point we also give results for an oracle frame-level approach *Frame* that has a single detection over all people with the correct count. We see EGR achieves a CLA much closer to the oracle EGR* than to *Frame*. When applying the random forest

Method	CLA	LA	MC	CE_d
$Frame$	0.0112	0.0112	0	0
EGR^*	0.1718	0.1976	0	0.15
EGR	0.1551	0.1830	0.04	0.18
RF	0.1101	0.1562	0.22	0.49
EGR_1	0.1166	0.1506	0.24	0.34
EGR_2	0.1387	0.1811	0.13	0.38
EGR_{n-}	0.1273	0.1668	0.06	0.31
EGR_{t-}	0.1186	0.1684	0.20	0.42
EGR_{e-}	0.1471	0.1753	0.05	0.34
[21]	0.1205	0.1562	0.23	0.25
[21]*	0.1718	0.1976	0	0.15

Table 1: Results for American Football dataset. Maximum number of targets is 23 per frame. To provide a better understanding of our CLA metric. *Frame* is an oracle frame-level approach.

classifier RF independently to detection, the count error increases. Comparing different iterations of EGR, we can see that we have a big performance boost from the first pass to second pass. This is mainly because with the tracker, we recover many missing detections. Among the three operators we have, the tracker plays the most important role under a limited number of iterations. It is interesting to note that for the football dataset over 70% of the detections output by our approach EGR have an assigned count of 1 and over 90% have a count of 4 or less. This shows that the detections output by the approach are generally quite localized to individuals or small groups. Detailed results are in the supplementary materials.

Tab.2 shows results for the pedestrian dataset. In terms of frame-level counting error for left-traveling and right-traveling people, which was the focus of prior work, our frame-level performance is a bit worse than two of the prior systems [5, 18]. In this dataset, the foreground segmentation provided by [5] is quite accurate compared to the football dataset, which means that our graph revision framework plays a much less important role here. Instead, the performance of the local random forest classifier becomes the dominating factor. Since our random forest classifier is much simpler than the regressors developed for previous work [5, 18], which was their primary focus but not ours, this result is not surprising. However, our approach is able to provide localization information in addition to frame-level counts. Since we cannot compute CLA scores for these frame-level approaches we report in the table the CLA for the oracle frame-level approach that provides 2 detections each covering all foreground segments in one direction and are assigned the correct count. We see that the CLA score of our approach is orders of magnitude larger. Our localization performance is also depicted in Fig.5, which illustrates the heat maps of counts for each pixel for the pedestrian dataset. We can draw some somewhat obvious

Method	Left CE_f	Right CE_f	CLA
EGR	1.05	3.83	0.2794
[5]	1.291	1.621	0.0853*
[8]	1.4458	11.1492	0.0853*
[18]	0.6040	0.6883	0.0853*

Table 2: Results for Pedestrian dataset. * We calculated CLA for [5, 8, 18] assuming they output 2 detections, one for each direction, and each direction has the correct counts.

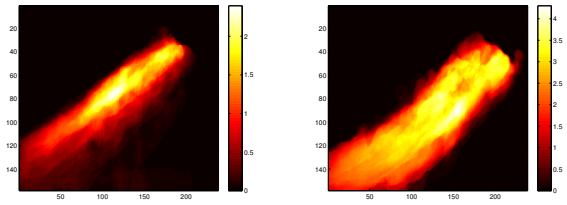

Figure 5: Heat map of counts for the pedestrian dataset. Left: left direction, Right: right direction. See Fig.7 for image of scene.

but interesting conclusions from these maps. For example, people in large groups tend to walk in the middle of the walkway. Also, most people walk along their right hand side of the walkway.

Runtime. The run time of our approach varies from one to 20 minutes for different videos on a desktop with a 4-core 3.4GHZ CPU. The majority of time is spent on the ILP solver. This suggests that the best way to speedup our current system would be to investigate approximate and fast solution techniques for the ILP problems we generate.

Qualitative Evaluation: Fig.6 illustrates count localization results on an example American Football video. [21] has similar results as EGR_1 and misses several players due to the noisy input. Some of the under-counting we see here is also due to missing players in previous frames. Our approach is able to identify these errors in the input and correct them through EGR. Similarly, Fig.7 shows our results on some frames of the pedestrian dataset. With the help of the person detector, we are able to get more localized counts. More results can be found in supplementary materials.

5. Conclusion

We formulated the new problem of person count localization that is useful for crowded domains with severe occlusion and interference among people such as team sports domains. We presented an approach to this problem called iterative error-driven graph revision, which attempts to overcome noisy input detections by detecting integrity constraint violations and then adjusting the optimization problem appropriately. This idea was shown to be useful in our experiments and more generally may be useful in other applications where global optimization problems are

Figure 6: An example image sequence from American football dataset. First row: inital input of foreground detections, second row: ([21]), third row :(EGR_1), forth row(EGR). [21] and EGR_1 miss a few players because of the initial input. EGR is able to recover these miss detections by iteratively revise the graph.

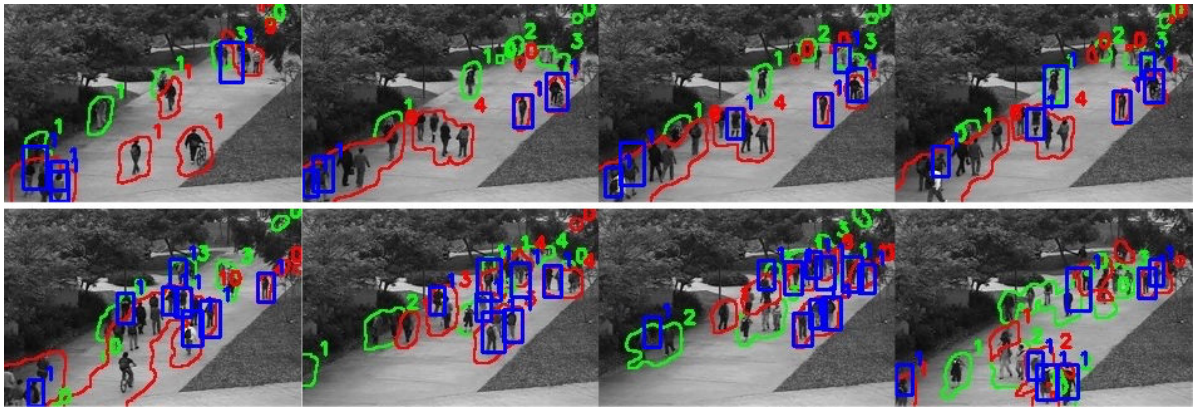

Figure 7: Pedestrian dataset. Green and Red boundaries outline the foreground blobs moving left and right. Green bounding boxes show smaller detection from greedy assignment.

formulated from vision data. We introduced a new metric called count localization accuracy for evaluating the localization and count quality of solutions and evaluated our approach on datasets derived from American football and moving pedestrians. The results show that our approach is significantly better than competitors when the input is noisy and that for the much less noisy pedestrian dataset our approach was competitive in terms of frame-level counts while also providing localization information.

Acknowledgment

This work was supported in part by grants NFS IIS 1219258 and NSF RI 1302700.

References

[1] J. Aggarwal and M. Ryoo. Human activity analysis: A review. *ACM Comput. Surv*, 2011.

[2] M. Amer and S. Todorovic. Sum-product networks for modeling activities with stochastic structure. In *CVPR*, 2012.

[3] M. Andriluka, S. Roth, and B. Schiele. People-tracking-by-detection and people-detection-by-tracking. In *CVPR*, 2008.

[4] B. Antic and B. Ommer. Learning latent constituents for recognition of group activities in video. In *ECCV*, 2012.

[5] A. B., C. Z.-S. John, and L. N. Vasconcelos. Privacy preserving crowd monitoring: Counting people without people models or tracking. In *CVPR*, 2008.

[6] A. B. Chan and N. Vasconcelos. Counting people with low-level features and bayesian regression. In *TIP*, 2012.

[7] W. Choi and S. Savarese. Understanding collective activities of people from videos. In *PAMI*, 2014.

[8] Y. Cong, H. Gong, S.-C. Zhu, and Y. Tang. Flow mosaicking: Realtime pedestrian counting without scene-specific learning. In *CVPR*, 2009.

[9] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan. Object detection with discriminatively trained part based models. *PAMI*, 32(9):1627–1645, 2010.

[10] W. Ge, T. R. Collins, and R. B. Ruback. Vision-based analysis of small groups in pedestrian crowds. *IEEE TPAMI*, 2012.

[11] J. Henriques and J. Caseiro R., Batista. Globally optimal solution to multi-object tracking with merged measurements. In *ICCV*, 2011.

[12] Y. Huang and I. Essa. Tracking multiple objects through occlusions. In *CVPR*, 2005.

[13] S. D. Khan, G. Vizzari, S. Bandini, and S. Basalamah. Detecting dominant motion flows and people counting in high density crowds. In *WSCG*, 2014.

[14] D. Kong, D. Gray, and H. Tao. A viewpoint invariant approach for crowd counting. In *ICPR*, 2012.

[15] C.-H. Kuo, C. Huang, and R. Nevatia. Multi-target tracking by on-line learned discriminative appearance models. In *CVPR*, 2010.

[16] T. Lan, L. Sigal, and G. Mori. Social roles in hierarchical models for human activity recognition. In *CVPR*, 2012.

[17] V. Lempitsky and A. Zisserman. Learning to count objects in images. In *NIPS*, 2010.

[18] Z. Ma and A. B. Chan. Crossing the line: Crowd counting by integer programming with local features. In *CVPR*, 2013.

[19] P. Nillius, J. Sullivan, and S. Carlsson. Multi-target tracking-linking identities using bayesian network inference. In *CVPR*, 2006.

[20] D. Ryan, S. Denman, C. Fookes, and S. Sridharan. Crowd counting using multiple local features. In *DICTA*, 2009.

[21] M. Schiegg, P. Hanslovsky, B. X. Kausler, L. Hufnagel, and F. A. Hamprecht. Conservation tracking. In *ICCV*, 2013.

[22] G. Shu, A. Dehghan, O. Oreifej, E. Hand, and M. Shah. Part-based multiple-person tracking with partial occlusion handling. In *CVPR*, 2012.

[23] L. Sun, H. Ai, and S. Lao. Activity group localization by modeling the relations among participants. In *ECCV*, 2014.

[24] J. Xing, H. Ai, and S. Lao. Multi-object tracking through occlusions by local tracklets filtering and global tracklets association with detection responses. In *CVPR*, 2009.

[25] B. Yang and R. Nevatia. Online learned discriminative part-based appearance models for multi-human tracking. In *ECCV*, 2012.

[26] K. Zhang, L. Zhang, and M.-H. Yang. Real-time compressive tracking. In *ECCV*, 2012.