This CVPR workshop paper is the Open Access version, provided by the Computer Vision Foundation.
Except for this watermark, it is identical to the version available on IEEE Xplore.

Real-time, Embedded Scene Invariant Crowd Counting Using Scale-Normalized Histogram of Moving Gradients (HoMG)

Parthipan Siva[‡], Mohammad Javad Shafiee[†], Michael Jamieson[‡], Alexander Wong[†]
[‡] Aimetis Corporation, Waterloo, ON, Canada
[†] VIP Research Group, University of Waterloo, Waterloo, ON, Canada
[‡]{parthipan.siva, mike.jamieson}@aimetis.com
[†]{mjshafiee, a28wong}@uwaterloo.ca

Abstract

Automated crowd counting has garnered significant interest for video surveillance. This paper proposes a novel scene invariant crowd counting algorithm designed for high accuracy yet low computational complexity in order to facilitate widespread use in real-time embedded video analytics systems. A novel low-complexity, scale-normalized feature called Histogram of Moving Gradients (HoMG) is introduced for highly effective spatiotemporal representation of crowds within a video. Real-time crowd region detection is achieved via boosted cascade of weak classifiers based on HoMG features. Based on the detected crowd regions, linear support vector regression (SVR) of crowd-region HoMG features is introduced for real-time crowd counting. Experimental results using a multi-scene crowd dataset show that the proposed algorithm outperforms state-of-the-art crowd counting algorithms while embedded on modern surveillance cameras. Thus demonstrating the efficacy of the proposed method for accurate, real-time, embedded crowd analysis.

1. Introduction

Video analytics – computer vision algorithms to process and understand video – for the surveillance industry has garnered significant interest for some time now. Traditionally, most video analytics applications for surveillance purposes have required the use of a centralized computer server, which has several key disadvantages. First, all video from the cameras must be sent to the centralized server, which puts a heavy load on the network. Second, the server must decode the compressed video streams coming from the cameras, which adds additional computational strains on the server. Due to these limitations, scaling video analytics systems from a few cameras to hundreds or even thousands of cameras needed for large scale video surveillance of transportation hubs and cities have become a major challenge. To address this limitation, many surveillance camera manufacturers such as Axis, Samsung, and Hikvision provide open platforms to embed video analytics algorithms directly on the camera. Many of the simpler video analytics algorithms such as motion detection, people tracking and counting in low traffic areas, have already been embedded on such video cameras. However, due to the limited free computational power available on these cameras, it is very difficult to embed more demanding algorithms.

In particular, for large-scale monitoring of busy public spaces, such as transportation hubs and city squares, algorithms for determining the presence and distribution of crowds are highly desired as they enable numerous applications ranging from analyzing crowd congestion patterns and customer attention behaviour, to detecting long queues, unsafe crowding or even mass panic. In particular, the crowd counting problem is of significant interest to the surveillance community and involves estimating the number of individuals within crowds.

Many methods have been proposed for the crowd counting problem [16, 17, 20, 22, 29, 4, 6, 13, 28], with a recent survey on state-of-the-art methods found in [25]. However, these methods require high-dimensional data formed from many computationally intensive features, making such methods unsuitable for embedding on cameras for large scale systems. Furthermore, most existing methods require expensive, manually-annotated training data for each scene, and even methods that can perform per-scene training over a period of time ([29, 17]) significantly complicate large-scale deployment given the need to adopt such methods to different scene settings and views. Ryan *et al*. [23, 24, 22] have proposed scene invariant approaches that do not require per-scene annotation or training; however, such approaches require the use of several computationally intensive features, which makes them unsuitable for real-time embedded crowd counting.

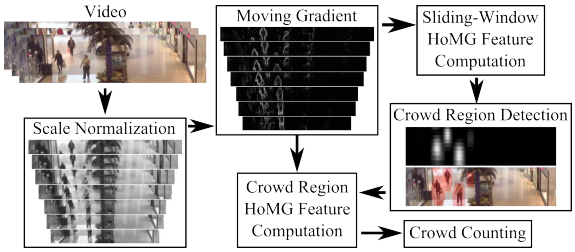


Figure 1: *Overview of proposed method. First, scale-normalized moving gradients are computed. Second, sliding-window Histogram of Moving Gradient (HoMG) features are computed and used to detect crowd regions. Finally, crowd-region HoMG features are computed based on the detected crowd regions and used to obtain the crowd count.*

In this paper, we propose a novel method for realtime, embedded scene invariant crowd counting. The proposed method centers around a novel, scale-normalized feature called Histogram of Moving Gradients (HoMG) that is designed to represent crowds with high accuracy yet low computational complexity. We show that our method significantly improves on the state-of-the-art for the seven scenes tested in [22] and introduce an expanded set of annotated videos for testing. We also demonstrate, with timing results, that our algorithm can be embedded on an Axis Q1615 camera with an ARTPEC-5 processor.

2. Methodology

An overview of the proposed crowd counting method is shown in Fig. 1. First, scale-normalized moving gradients are obtained to facilitate the computation of HoMG features (Section 2.1). Based on the obtained moving gradients, sliding-window HoMG features are computed and classified to obtain crowd regions (Section 2.2). Finally, using both the crowd region detection results and the scale normalized moving gradients, crowd-region HoMG features are computed and used by a linear regressor to obtain the crowd count (Section 2.3).

2.1. Histogram of Moving Gradients (HoMG)

We introduce the concept of histogram of moving gradients (HoMG) a single, powerful, yet low-complexity, scale-normalized feature descriptor used for both crowd region detection and counting. We will show that sliding-window HoMG features can be used for crowd region detection in an effective and efficient manner. Furthermore, we show that the cumulative scale-normalized moving gradient magnitude in crowd regions is linearly related to the number of people in the frame, thus leading to crowd-region HoMG features that can be easily used to obtain the crowd count.

Figure 2: *Scale normalization: each video frame is broken into overlapping strips of height $W_h(r)$ (the height of person at a given image row r) and re-scaled such that the person width in each strip goes to w_p pixels.*

2.1.1 Scale Normalization

It is often helpful, when seeking to detect or recognize objects, to compute features at a scale that is relative to the object's size in the image. For instance, in SIFT [18], descriptors are computed at a fixed window size based on the scale of the detected key points. Since our interests lie in crowd detection, we instead compute HoMG feature descriptors at a fixed scale relative to person size. Therefore, in this work, the goal is to re-scale the expected person size windows at all locations in the image such that the width of the person is w_p pixels wide.

The scale normalization process here is driven by the notion that while the general shape of people is similar, there can be significant clothing variations. As such, it is important to fix the scale such that it is small enough that clothing details are removed, yet large enough that the general shape of individuals are preserved. Motivated by this, we leverage the camera calibration model, with the assumption that the person height and width per image row is approximately constant[1], and decompose the image into overlapping strips (Fig. 2) of height $W_h(r)$, where $W_h(r)$ is the height of a person at row r obtained from the calibration model.

Based on the calibration model, a person within each strip will have a width of $W_w(r)$ pixels. Therefore, each strip is re-scaled by s such that the average person width becomes w_p pixels (i.e., $s = \frac{w_p}{W_w(r)}$). The resulting scale-normalized frame $\mathcal{S}$ is illustrated in Fig. 2. Furthermore, based on the calibration model, any locations in the scene where the person size is less than w_s pixels in width (in the original video resolution) will not be analyzed.

2.1.2 Moving Gradients

A key observation when incorporating temporal information for crowd analysis is that a person, even if they are waiting at a location, is never perfectly stationary over a period of time. Such slight motions can cause problems for standard background subtraction techniques [26], as parts

[1] while this assumption may sometimes be violated due to lens distortion and camera rotations, such an assumption makes the sliding-window approach computationally tractable.

of the person will become background while other parts are treated as foreground, thus leading to unreliable crowd segmentation. To mitigate such issues, we identify moving gradients instead, allowing us to capture the outlines of individuals who are standing at a single location but exhibit slight motions.

For a given scale-normalized frame $\mathcal{S}$, we wish to identify its moving gradients, which essentially characterize the edges of moving foreground objects. There are a number of approaches to computing moving gradients such as background modeling followed by edge detection, background modeling on the edge image, frame differencing, etc. For the purposes of this paper, we chose to compute moving gradients based on frame differencing in the Sobel [14] gradient domain as it is computationally efficient.

Formally, given an input video, we represent a scale-normalized grayscale strip at time t as $S^t \in \mathcal{S}$, where $t = 0$ refers to the current strip and $t = -a$ refers to the strip a seconds ago. Furthermore, we define the Sobel [14] gradient of strip S^t as $\vec{E}^t = (E_x^t, E_y^t)$, where E_x is the horizontal gradient component and E_y is the vertical gradient component.

The moving gradient, E, of a current-frame strip S^0 is defined as the truncated median gradient between S^0 and a set $\mathcal{A}$, consisting of the past frame (a_0) and l past keyframes ($a_1, \ldots, a_l$) (Fig. 3):

$$E = \min(\max(0, \bar{E} - T_1), T_2) \quad (1)$$

$$\bar{E} = \underset{a \in \mathcal{A}}{\text{median}}(D^a) \quad (2)$$

$$D^a = \begin{cases} \|\vec{E}^0 - \vec{E}^a\| & \text{if } \|\vec{E}^0\| > \|\vec{E}^a\| \\ 0 & \text{else} \end{cases} \quad (3)$$

where T_1 and T_2 are thresholds on gradient magnitude to minimize the effects of very strong or very weak gradient magnitudes, and $\mathcal{A} = \{-a_0, -a_1, \ldots, -a_l\}$ is the temporal scale of the moving gradient. Eq. 3 effectively detects edges on the current strip not present in the strip a seconds ago.

The moving gradient magnitude E is given by Eq. 1, while the orientation O is obtained as

$$O = \left\lfloor N \frac{atan2\left(E_y^0, E_x^0\right) + \pi}{2\pi} \right\rfloor \quad (4)$$

where $\vec{E}^0$ denotes the Sobel gradient of the current frame, and N represents the number of discretized orientation bins.

To allow for immediate reporting of crowd information when the algorithm is first started, we allow $\mathcal{A}$ to be small (i.e. $\mathcal{A} = \{-a_0\}$) and then grow to the full size $\mathcal{A} = \{-a_0, -a_1, \ldots, -a_l\}$. As such, at least a_0 seconds of video is needed before reporting crowd statistics.

Frame differencing over long periods of time requires a significant amount of memory to store the historical frames $\mathcal{A}$. For embedded systems we have limited memory availability; as such we use frame differencing with

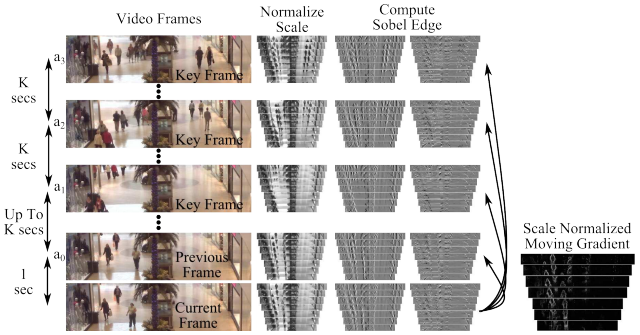


Figure 3: *The moving gradient is obtained as the truncated median gradient between the current frame and a set consisting of the past frame and l past keyframes. The closest previous keyframe (a_1) can be up to K seconds from the current frame.*

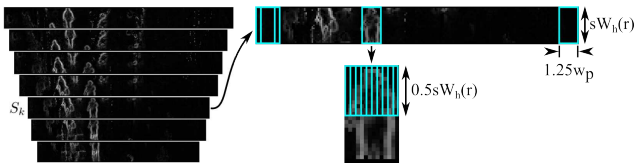


Figure 4: *Illustration of a sliding-window within a strip and the $1 \times 1.25w_p$ celled HoMG block used within a sliding-window.*

keyframes, where keyframes are obtained at a much lower frames per second (FPS) than the video being processed. Set $\mathcal{A}$ consists of the past frame (a_0) and past l keyframes ($a_1, \ldots, a_l$), rather than a large set of past frames (Fig. 3). Only $l + 1$ frames need to be kept in memory, which significantly reduces memory overhead.

2.1.3 Histogram Construction

Based on the moving gradient magnitude E and the moving gradient orientation O, the histogram of moving gradients (HoMG) within a window of interest W is defined as

$$h(\theta) = \sum_{p \in W | O(p) = \theta} E(p) \quad (5)$$

2.2. Crowd Region Detection

For crowd region detection, we wish to detect the region occupied by crowds on each scale-normalized strip S_k in $\mathcal{S}$. This is accomplished using a sliding-window approach (Fig. 4), with the window spanning the height of the strip and a width of $1.25w_p$ pixels, where w_p is the width of the person as defined in Section 2.1.1.

2.2.1 Sliding-Window HoMG

Given a sliding-window of interest, a single $1 \times 1.25w_p$ HoMG block is used as defined in Fig. 4. The HoMG representation of the block is denoted by h^g_θ, where θ is the discretized moving gradient orientations and g is the cells in the HoMG block. Furthermore, we compute the total moving gradient (MG) magnitude within each of the cells as $s^g = \sum_\theta h^g_\theta$. Finally, the total moving gradient (MG) magnitudes within the entire window of interest t is also computed. The HoMG block, MG magnitudes within the cells, and the MG magnitude within the entire window are concatenated to form the crowd detection feature f.

$$f = \{h^g_\theta, s^g, t\} \tag{6}$$

In this work, we use 8 moving gradient orientations and $1 \times 1.25w_p = 1 \times 10$ cells in the HoMG block; as a result, the crowd region detection feature vector f is of dimensions 1×91.

While all strips are scale-normalized to obtain constant person width, the person height (i.e., height of the strip) may still vary within an image due to differences in viewing angle at different image locations. To counteract this variance, we normalize our feature vector f (6) by the height of the strip.

2.2.2 Classification

Given the computed feature vector, the next step is to classify whether the window of interest is crowd or non-crowd. While there are many effective classification algorithms, we find that a boosted cascade of weak classifiers [27], due to its low computational complexity at run time, best satisfies our primary goal of achieving a real-time algorithm on relatively slow hardware. Specifically, the AdaBoost [12] algorithm is used to obtain 100 boosted weak classifiers, where each weak classifier is a decision tree [8] with three decision nodes.

For training, the positive samples centred around each annotated head location were used, with the negative samples obtained from unannotated areas. Since the negative space is much larger than the positive space, we employ the negative mining technique of [11].

2.2.3 Crowd Region Detection

The boosted weak classifier is used to classify each sliding-window on each strip as crowd or non-crowd with a confidence score. Each sliding-window W_s in the strip S_k has a corresponding window W in the original-resolution video frame. The scores are accumulated at the original video frame resolution using the correspondence window W. The accumulated score image C is thresholded to obtain the crowd region detection image $\hat{C}$ (Fig. 5).

Figure 5: *The accumulated crowd classification score image C is thresholded to identify crowd regions $\hat{C}$ (shown in red).*

2.3. Crowd Counting

Given the crowd region detection results, we now employ a linear regression approach to crowd counting using HoMG. We assume that the cumulative scale-normalized moving gradient magnitudes in the crowd regions are linearly related to the number of people in the video frame. While this is not strictly true due to occlusions and camera angles, we will show that this approach works very well in comparison to state-of-the-art.

2.3.1 Crowd-Region HoMG

Based on the aforementioned linear relationship assumption, we introduce a cumulative HoMG for the crowd counting feature R, where only one histogram bin is used (i.e., moving gradient orientation is ignored):

$$R = \sum_{S_k \in \mathcal{S}} \frac{1}{h_k} \sum_{p \in S_k} \delta(p) E(p) \tag{7}$$

where $S_k \in \mathcal{S}$ are all the scale-normalized strips in the frame, h_k is the height of the k^{th} strip after scale normalization, $p \in S_k$ are all the pixels in stripe S_k, $E(p)$ is the moving edge magnitude at pixel p, and $\delta(p) = \{0, 1\}$ is the crowd detection result at pixel p. This results in a single one-dimensional feature which is proportional to the number of people in the frame.

2.3.2 Regression

We use linear support vector regression (SVR) [5] as our regression method for crowd counting. Linear SVR was chosen for two reasons: i) its robust nature when fitting a line to the data, and ii) low computation complexity at run time, which is critical for real-time embedded crowd counting. Furthermore, our assumption is that there is a linear relationship between moving gradient magnitude and the number of people in the video frame; as such it stands to reason that when the moving gradient magnitude approaches zero, the estimated number of people in the frame should approach zero. Therefore, we assume the bias term in our linear SVR model is zero.

Figure 6: *Scenes from the proposed new dataset: comprising of public datasets (scenes 4-13) and 3 new scenes (scenes 1-3).*

3. Evaluation Setup

To quantitatively evaluate the proposed crowd counting method, which we will refer to as HoMG, we introduce a new multi-scene dataset, which has almost double the number of scenes as previously-used datasets. Competing methods used for evaluation as well as evaluation metrics used are all described below.

3.1. Dataset

We introduce a new dataset for scene invariant crowd counting which brings together existing crowd detection datasets [19, 22], tracking datasets [3] as well as a number of new videos, all with a consistent annotation format. Furthermore, for each scene, we provide coarse camera calibration information as discussed in Section 3.4. The proposed dataset has 13 different scenes (Fig. 6): 1-3 are new videos, 4 is the Mall dataset [19], 5 is the City Center dataset [3], 6 is chosen from the i-LIDS dataset [15], and 7-13 are from the PETS and QUT dataset previously used for crowd counting in [22].

All evaluations are based on a leave-one-out framework (same as [22]) where one scene is used for testing, while remaining scenes are used for training. To balance the data and be consistent with the previous works ([22]) only the first 50 annotated frames from each scene are used for training.

3.2. Competing Methods

The proposed HoMG method is compared quantitatively with two state-of-the-art frameworks (note that the comparison methods were not implemented on the embedded architecture and are only used for accuracy comparisons):

Single Camera Crowd Counting (S3C) Several types of features including segments, edges, GLCM, and LBP are extracted from video sequence, and a kernel ridge regression (KRR) [19] is utilized to estimate the crowd count of the scene. This method originally was used in the situation when training and test videos are from the same scene. However, this method is compared here in a scene invariant scenario.

Scene Invariant Multi Camera Crowd Counting (SIM3C) This method [22] is the state-of-the-art approach in scene invariant crowd counting that doesn't require any adaptation to the scene being tested. SIM3C obtains several types of features including size, shape, edge, and key-points extracted using SURF [2], and feeds these features into a non-linear Gaussian process regression framework to estimate the crowd count.

3.3. Evaluation Metric

Crowd counting performance is evaluated using three different quantitative metrics: Mean Absolute Error (MAE) [19], Mean Square Error (MSE) [19] and Mean Deviation Error (MDE) [6].

$$MAE = \frac{1}{N}\sum_{i=1}^{N}|y_n - \hat{y}_n| \quad (8) \qquad MSE = \frac{1}{N}\sum_{i=1}^{N}(y_n - \hat{y}_n)^2 \quad (9)$$

$$MDE = \frac{1}{N}\sum_{i=1}^{N}\frac{|y_n - \hat{y}_n|}{y_n} \quad (10)$$

where N represents the number of testing frames, and y_n and $\hat{y}_n$ denote the actual count and the estimated count of frame n, respectively.

3.4. Algorithm Setup

As with other crowd counting methods [22, 19], a camera calibration model is needed for our HoMG method. In [19], a very simple perspective model is used which does not account for the non-linear behaviour of perspective models. In [22], a full calibration model is used; however, this is often unavailable in industry surveillance applications. Therefore, we instead employ a model currently employed in industry that acts as a compromise between [19] and [22]. Here, we use the interactive camera calibration model used commercially in the Aimetis Symphony [1] surveillance software package, where an approximate calibration model is obtained by having the users select calibration parameters that matches the model's estimated person sizes to the person sizes in the video.

Using this calibration model, we set the scaled person width (w_p), from Section 2.1.1 to $w_p = 8$ pixels. We select 8 pixels as a value small enough to detect crowds far from camera but at the same time is large enough to learn features to describe individuals and crowds.

Similar to [22, 19], which requires parameters for background subtraction algorithms used to obtain moving blobs, the HoMG algorithm requires the temporal scale $\mathcal{A} = \{1, K, 2K, \ldots, lK\}$ (Section 2.1.2) to be defined for computing HoMG features. A temporal scale of 120 seconds was used to determine moving gradients, where keyframes

are set 30 seconds apart (i.e., $l = 4$ and $K = 30$). The scale of 120 seconds was chosen as it is long enough to ensure people who exhibit slight motions be identified, while still being computationally tractable. As a result of our keyframe based frame differencing only 5 frames must be kept in memory to do frame differencing over 120 seconds, greatly reducing the memory requirements.

For crowd region detection (Section 2.2.2), we use the AdaBoost classifier implemented in MATLAB [21], with a 3-node decision tree as the weak classifier. All other learning parameters were left to the default values. For crowd counting (Section 2.3.2), we use the LIBLINEAR [10] implementation of linear SVR with default learning parameters. The resulting boosted classifier and SVR weights are ported to the embedded environment, thus illustrating the efficacy of the proposed method for real-time, embedded crowd counting.

3.5. Algorithm Implementation

There are many camera manufacturers with open platforms for developing embedded applications; however, Axis cameras and their 3rd party development platform is among the most popular and mature platforms currently available in the surveillance industry. Their latest cameras are available with Axis' ARTPEC-5 chip-set (MIPS 1004Kc V2.12 CPU model) running a striped down version of Linux. Their older generation cameras utilize the Axis' ARTPEC-4 chip-set (MIPS 34Kc V5.0 CPU model). To support the widest range of available platforms, we implement our algorithms on both ARTPEC-4 and ARTPEC-5 chip-set cameras. While our implementation will work on any Axis camera that supports embedded development and has either the ARTPEC-5 or ARTPEC-4 chip-set, we test our algorithms on the Q1614 and Q1615 cameras, with the Q1614 camera having an ARTPEC-4 chip-set and the Q1615 camera having an ARTPEC-5 chip-set.

The proposed HoMG approach was implemented using C++ and compiled with the Axis development SDK. One of the most computational intensive part of our algorithm is the $atan2$ computation in Eq. 4. To avoid the associated computational complexity, we perform the $atan2$ computation using a lookup table.

4. Crowd Counting Results

Performance analysis of HoMG for crowd counting is carried out on the 13 scene dataset (Section 4.1). Furthermore, to allow for direction comparison with state-of-the-art [22], HoMG is also tested on a 7 scene subset of the dataset, which contains only the scenes used by [22].

4.1. Full Dataset

The crowd counting results are shown in Table 1 and Fig. 7. The proposed HoMG method achieved relatively

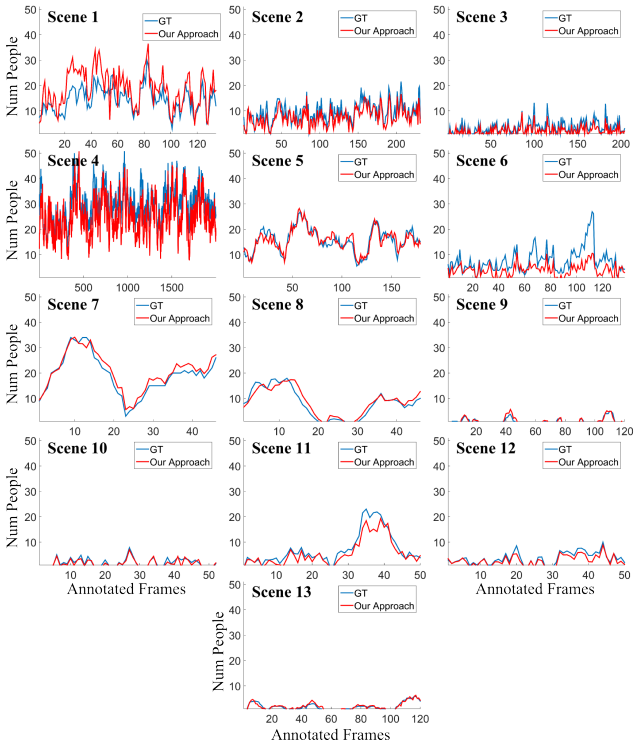


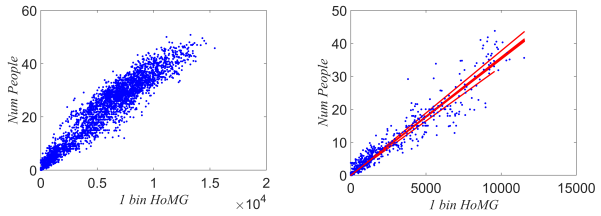
Figure 7: *The ground truth (GT) and predicted counts for all annotated frames, from the leave-one-out testing.*

Scenes		1	2	3	4	5	6	7	8	9	10	11	12	13	Avg.
MAE	HoMG	**4.92**	**2.09**	**1.33**	**5.34**	**1.53**	**3.86**	**1.82**	**1.41**	**0.31**	**0.49**	**1.75**	**0.85**	**0.33**	**2.00**
	S3C	15.0	17.9	7.9	23.7	15.5	4.0	9.1	7.8	1.5	2.5	2.8	1.7	1.6	7.9
MSE	HoMG	**35.1**	**6.9**	**3.0**	**38.8**	**3.7**	**24.4**	**1.7**	**3.5**	**0.3**	**0.5**	**6.2**	**1.3**	**0.2**	**9.9**
	S3C	62	339	83	608	234	25	99	95	4	10	15	5	4	122
MDE	HoMG	**0.35**	**0.23**	**0.37**	**0.18**	**0.11**	**0.49**	**0.12**	**0.25**	**0.35**	**0.28**	**0.26**	**0.24**	**0.20**	0.27
	S3C	1.0	2.1	3.6	0.8	0.9	0.9	0.5	3.3	47.8	203	0.5	0.7	1.7	20.5

Table 1: *Crowd counting results on the full dataset*

low average error across all scenes with respect to both MSE and MAE. In comparison to **S3C**, which is a baseline comparison, HoMG achieves almost four times lower MAE rate and over twelve times lower MSE rate. However, as previously stated in Section 3.2, **S3C** was not originally designed as a scene invariant approach; nevertheless, we treat it as such because it does have a very coarse perspective normalization.

To illustrate that the crowd-region HoMG feature is linearly related to the people count, we plot the crowd-region HoMG feature vs number of people in the frame in Fig. 8. The 13 regression lines (Fig. 8b) – obtained from leave-one-out testing – are almost identical, indicating this is a highly-effective and consistent scene invariant feature. The average slope of the 13 lines is 3.5E-3 with a standard deviation of 9.9E-5. However, even with this small deviation in the fit-



(a) *13 Scenes All Data* (b) *13 Scenes Training Data*

Figure 8: *(a) crowd-region HoMG feature vs number of people in the scene. (b) plots only the data used for training (first 50 frames) as well as all linear SVR lines obtained during leave-one-out testing on the sceans.*

Scenes		7	8	9	10	11	12	13	Avg.
MAE	HoMG	1.741	**1.534**	**0.303**	**0.493**	1.749	**0.849**	**0.330**	**0.926**
	SIM3C	**1.321**	3.365	0.405	1.574	**0.886**	1.448	0.487	1.355
	S3C	6.84	4.74	1.51	15.55	4.26	2.03	1.65	5.23
MSE	HoMG	**3.893**	**3.956**	**0.294**	**0.469**	4.031	**1.158**	**0.182**	**1.997**
	SIM3C	4.250	17.514	0.495	3.506	**1.524**	3.625	0.441	4.479
	S3C	76.77	38.154	5.52	3.91	50.06	6.61	3.79	26.40
MDE	HoMG	0.123	0.284	0.363	0.263	0.186	0.204	**0.191**	0.231
	SIM3C	**0.103**	**0.096**	**0.250**	**0.140**	**0.122**	**0.182**	0.222	**0.159**
	S3C	0.72	1.99	31.82	44.23	0.45	0.69	1.82	11.67

Table 2: *Crowd counting results on 7 Scenes subset used in [22].*

ted line there are two outliers, which we take a closer look in Fig. 9. From Fig. 9 we can see that for Scene 1 there is a fairly consistent over-estimation of the people count, while there is an under-estimation of the people count in Scene 4.

On closer examination of Scene 1, we find that the store front glass displays are acting as a mirror (Fig. 9), reflecting people as they walk by. The reflections are classified as crowd regions by HoMG and are thus included in our regression estimation, resulting in an over-estimation of the people count.

On closer examination of Scene 4 we find that there are missed detections due to people sitting down and people standing where only their heads are visible (Fig. 9). Seated people are very stationary for long periods of time, and as such our moving gradient based approach to crowd segmentation will tend to miss them. Our HoMG block for crowd segmentation covers a person's torso (Fig. 4) and as such when only an individual's head is visible, HoMG will not be as effective.

4.2. Comparison to State-Of-The-Art (7 Sceans)

We compare HoMG to the state-of-the-art results reported by Ryan et al. [22] (**SIM3C**) on the same 7 Scenes used by Ryan et al. For a fair comparison, we train both the proposed crowd segmentation and regression methods only on the 7 Scenes used in **SIM3C** [22]. The results of the crowd counting are shown in Table 2.

HoMG, on average, achieved 1.5 times lower MAE and 2.25 times lower MSE than **SIM3C**. This result is achieved even though we use a single feature (HoMG) in comparison to **SIM3C** which use multiple features such as moving blob shape features, edge features, and key point descriptors. However, HoMG exhibited a 1.4 times higher MDE than **SIM3C**, due to the variation of data from the line fit seen in Fig. 8a, especially when the number of people in the frame is low.

5. Runtime Analysis

Our HoMG algorithm was able to run on the Axis Q1615 (ARTPEC-5 chip-set) with a run-time of about 500 ms for a 640 by 480 video frame. On the older Axis Q1614 (ARTPEC-4 chip-set) it took about 600 ms to process a 640 by 480 video frame. This results in a processing speed of only 1 to 2 FPS. However, since crowd behaviour is slow to evolve over time, processing one frame per second is more than sufficient for the purpose of real-time crowd counting. In fact, many of the existing work on crowd counting [19] are designed and tested for processing video at 1 FPS. This level of run-time performance on these two videos illustrate the efficacy of the proposed algorithm for real-time, embedded crowd counting.

Using the same C++ implementation, we run our algorithm on an Intel Core i7-2720QM processor at 2.2GHz. It takes about 20 ms to process a 640 by 480 frame which results in a processing speed of 50 FPS. As can be seen, the embedded environments on modern surveillance cameras are 25 to 30 times slower than modern computers. Even on these low computation systems our algorithm is still capable of real-time crowd counting.

6. Conclusions

In this paper, a novel, low-complexity scale-normalized histogram of moving gradients (HoMG) feature is introduced for robust and real-time embedded scene invariant crowd counting. Experimental results using an existing multi-scene dataset demonstrate that the proposed crowd counting method using HoMG can outperform state-of-the-art approaches while being able to process 1 FPS stream embedded directly on modern surveillance cameras. Furthermore, we also introduce an expanded dataset with 13 scenes featuring a greater variety of camera angles to demonstrate the performance of the proposed method for crowd counting. Based on the existing and expanded datasets, we show that the proposed method using HoMG facilitates robust, real-time, embedded crowd analysis, which is important for widespread industrial adoption.

References

[1] Aimetis. Symphony. http://www.aimetis.com. 5

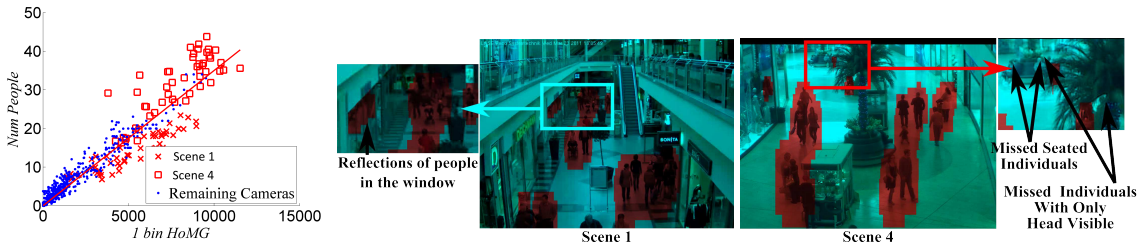


Figure 9: *Scene 1 and 4 differs slightly from the linear SVR fit, exhibiting consistent bias below or above the linear SVR fit, respectively. This causes an over-estimation of crowd size in Scene 1 and an under-estimation in Scene 4. The over-estimation is due to edge contributions of people's reflections in the store windows and the under-estimation is due to missed detection of seated individuals.*

[2] H. Bay, A. Ess, T. Tuytelaars, and L. Van Gool. Speeded-up robust features (surf). *Computer vision and image understanding*, 2008. 5

[3] B. Benfold and I. Reid. Guiding visual surveillance by tracking human attention. In *BMVC*, 2009. 5

[4] A. Chan and N. Vasconcelos. Counting people with low-level features and bayesian regression. *Image Processing, IEEE Transactions on*, 21(4):2160–2177, 2012. 1

[5] V. Cherkassky and Y. Ma. Practical selection of svm parameters and noise estimation for svm regression. *Neural networks*, 2004. 4

[6] D. Conte, P. Foggia, G. Percannella, and M. Vento. A method based on the indirect approach for counting people in crowded scenes. In *IEEE International Conference on Advanced Video and Signal Based Surveillance*, 2010. 1, 5

[7] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *CVPR*, 2005.

[8] T. G. Dietterich. Ensemble methods in machine learning. In *Multiple classifier systems*. Springer, 2000. 4

[9] M. Everingham, L. Gool, C. K. Williams, J. Winn, and A. Zisserman. The pascal visual object classes (voc) challenge. *IJCV*, 88(2):303–338, June 2010.

[10] R.-E. Fan, K.-W. Chang, C.-J. Hsieh, X.-R. Wang, and C.-J. Lin. Liblinear: A library for large linear classification. *J. Mach. Learn. Res.*, 9:1871–1874, June 2008. 6

[11] P. Felzenszwalb, R. Girshick, D. McAllester, and D. Ramanan. Object detection with discriminatively trained part-based models. *TPAMI*, 32(9):1627–1645, Sept 2010. 4

[12] Y. Freund and R. E. Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of computer and system sciences*, 1997. 4

[13] S. Ghidoni, G. Cielniak, and E. Menegatti. Texture-based crowd detection and localisation. In *Intelligent Autonomous Systems*. 2013. 1

[14] R. C. Gonzalez and R. E. Woods. *Digital Image Processing (3rd Edition)*. Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 2006. 3

[15] HOSDB. Imagery library for intelligent detection systems (i-lids). In *IEEE Conf. on Crime and Security*, 2006. 5

[16] K. Kang and X. Wang. Fully convolutional neural networks for crowd segmentation. *CoRR*, abs/1411.4464, 2014. 1

[17] T.-Y. Lin, Y.-Y. Lin, M.-F. Weng, Y.-C. Wang, Y.-F. Hsu, and H.-Y. Liao. Cross camera people counting with perspective estimation and occlusion handling. In *Information Forensics and Security (WIFS)*, 2011. 1

[18] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *IJCV*, 2004. 2

[19] C. Loy, K. Chen, S. Gong, and T. Xiang. Crowd counting and profiling: Methodology and evaluation. In *Modeling, Simulation and Visual Analysis of Crowds*. Springer New York, 2013. 5, 7

[20] M. Manfredi, R. Vezzani, S. Calderara, and R. Cucchiara. Detection of static groups and crowds gathered in open spaces by texture classification. *Pattern Recognition Letters*, 44, 2014. 1

[21] Mathworks. Matlab. http://www.mathworks.com/products/matlab/. 6

[22] D. Ryan, S. Denman, C. Fookes, and S. Sridharan. Scene invariant multi camera crowd counting. *Pattern Recognition Letters*, 44, 2014. 1, 2, 5, 6, 7

[23] D. Ryan, S. Denman, S. Sridharan, and C. Fookes. Scene invariant crowd counting. In *Digital Image Computing Techniques and Applications (DICTA), 2011 International Conference on*, pages 237–242, Dec 2011. 1

[24] D. Ryan, S. Denman, S. Sridharan, and C. Fookes. Scene invariant crowd counting and crowd occupancy analysis. In C. Shan, F. Porikli, T. Xiang, and S. Gong, editors, *Video Analytics for Business Intelligence*, volume 409 of *Studies in Computational Intelligence*, pages 161–198. Springer Berlin Heidelberg, 2012. 1

[25] D. Ryan, S. Denman, S. Sridharan, and C. Fookes. An evaluation of crowd counting methods, features and regression models. *Computer Vision and Image Understanding*, 130:1 – 17, 2015. 1

[26] C. Stauffer and W. Grimson. Adaptive background mixture models for real-time tracking. In *CVPR*, 1999. 2

[27] P. Viola and M. Jones. Rapid object detection using a boosted cascade of simple features. In *CVPR*, 2001. 4

[28] S. Yang, H. Bao, B. Wang, and H. Lou. Crowd density estimation based on ELM learning algorithm. *JSW*, 8(11):2839–2846, 2013. 1

[29] C. Zhang, H. Li, X. Wang, and X. Yang. Cross-scene crowd counting via deep convolutional neural networks. In *CVPR*, 2015. 1