Efficient Point Process Inference for Large-scale Object Detection

Trung T. Pham, Seyed Hamid Rezatofighi, Ian Reid and Tat-Jun Chin
School of Computer Science
The University of Adelaide
{trung.pham,hamid.rezatofighi,ian.reid,tjchin}@adelaide.edu.au

Abstract

We tackle the problem of large-scale object detection in images, where the number of objects can be arbitrarily large, and can exhibit significant overlap/occlusion. A successful approach to modelling the large-scale nature of this problem has been via point process density functions which jointly encode object qualities and spatial interactions. But the corresponding optimisation problem is typically difficult or intractable, and many of the best current methods rely on Monte Carlo Markov Chain (MCMC) simulation, which converges slowly in a large solution space.

We propose an efficient point process inference for large-scale object detection using discrete energy minimization. In particular, we approximate the solution space by a finite set of object proposals and cast the point process density function to a corresponding energy function of binary variables whose values indicate which object proposals are accepted. We resort to the local submodular approximation (LSA) based trust-region optimisation to find the optimal solution. Furthermore we analyse the error of LSA approximation, and show how to adjust the point process energy to dramatically speed up the convergence without harming the optimality. We demonstrate the superior efficiency and accuracy of our method using a variety of large-scale object detection applications such as crowd human detection, birds, cells counting/localization.

1. Introduction

Object detection in images and video is one of the fundamental problems in computer vision. While there has been remarkable progress in detecting small and moderate numbers of potentially complex deformable objects in 2D images [12, 11], in this work we focus on large-scale object detection, considering images that may capture hundreds or even thousands of objects; Figure 1 shows typical examples. Such scenarios are encountered in many useful real-world applications ranging from estimating crowds in video surveillance [14, 4, 25] to counting cells in microscope images [20], bird populations (e.g., flamingos) [6] and tree crowns [23] in remotely sensed images. Beside being large-scale, these scenarios often include significant overlap/occlusion of objects, which significantly complicates the process of localizing individual objects.

Traditional object detection methods usually take a two-stage approach. First, a large number of object hypotheses are generated by running a scanning window detector at different scales and locations. This procedure unavoidably returns many imprecise overlapping object hypotheses. To prune out redundant object detections, a greedy non-maximum suppression (NMS) step is often applied, which basically selects the maximum responses at each location. However such a heuristic elimination suffers from several limitations such as inability to detect nearby or overlapping objects [17].

On the other hand, *global* methods [23, 8, 19, 28] jointly optimize object detection and selection. Almost all of these methods fall under the stochastic geometry framework [2], which utilises point processes to model *global object configurations*. Point processes allow convenient modelling of the spatial pattern of the object configuration, as well as the interaction between objects, with the optimal object configuration inferred by optimizing the point process probability density. Typically the inference uses Reversible Jump Markov Chain Monte Carlo (RJMCMC) simulation with simulated annealing, whose convergence is slow, especially for large-scale problems.

In this paper we propose a novel *global* method for large-scale object detection, which utilizes both the elegant point process formulation and efficient discrete energy minimization. As in [19, 28], we define an energy function of object configurations pertaining to the point process, whose minimizer will give rise to an optimal set of objects. The energy simultaneously encodes object detection scores (*e.g.* confidences) and spatial object interactions (*e.g.* overlapping). Unlike [19, 28] where the optimization is solved over a continuous object state space using RJMCMC simulation, we instead approximate the state space by a finite set of all possible objects—for example proposing objects at all pixel

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

Figure 1. Best viewed electronically. Typical examples of large-scale object detection. (a) and (b) respectively show an image of stem cells and the detection result using our proposed method, where 4144 stem cells are detected. (c) displays an image of a crowd participating in a marathon. (d) Our method is able to detect 492 runners.

locations with all possible (discrete) sizes, orientations. We will show that such a discretization not only does not greatly affect the detection accuracy, but also permits efficient discrete optimization.

We begin by constructing an energy function of binary variables whose values indicate which of all possible object proposals are selected. Since the resulting energy is non-submodular in general, we resort to the local submodular approximations (LSA) based trust region method [15]. Though [15] provides for efficient optimization, the approximation error of LSA for a naive implementation of the point process energy could be arbitrarily large (and hence cause the trust-region optimisation to converge slowly or to become stuck at a low quality solution). As the approximation error is caused by sub-modularizing the pairwise object energies (which are used to encode spatial object constraints), we propose to reduce the hurdles of the pairwise sub-modularization by conditionally decreasing the pairwise energies to their minima such that inter-object constraints are still guaranteed. Technically the new energy function will admit the same global optimum as the original one. We empirically validate the superior efficiency and accuracy of our framework using a variety of large-scale object detection problems including bird detection from remotely sensed images, cell detection/counting from microscope images and crowd human detection from surveillance cameras.

2. Related Work

In computer vision, object detection is a process of identifying individual object instances in images or video sequences. Objects of interest can be anything ranging from semantic categories [10] such as humans, cars, animals, furniture to geometric ones such as building outlines [22], road networks [24, 18]. To this end, one would need to have an object model, which is used to measure the likelihood of an object instance (with location and shape parameters) appearing in the input image. Depending on object complexities and scene contexts, the object likelihood can be computed using simply pixel intensities [13, 6] (*e.g.* contrast) or sophisticated deformable part models (DPM) [11] learned from annotated training data. Also depending on the specific applications the number of objects in a single image could vary from a few [11] to thousands (*e.g.* bird colony [6]).

In this work we are interested in large-scale object detection, where hundreds to thousands of objects present in each image, and objects heavily overlap (see Fig. 1). Dollar *et al.* [9] have shown that the object detection performance degrades disproportionately to the degree of object occlusion. Though the work [9] has been validated on human detection, the conclusion on the negative effects of occlusion generalizes well to other objects.

Large-scale object detection has been encountered in various vision applications such as bird counting from remotely sensed images (e.g., flamingo [6]), or line-network [18] and building [22] extraction from digital elevation models. In common, all these works rely on stochastic point processes [2] to model spatial distribution of objects, which could account for object overlaps, angles between line segments or alignments between rectangles. Though demonstrated promising results, the simulation of the point process framework is often computationally expensive and converges unstably [28]. In fact, these works use simulated annealing coupled with RJMCMC sampling techniques to infer the optimal object configuration that best explains the input data. Basically, RJMCMC performs a sequence of births, deaths or updates on the state space until convergence which is usually very slow in practice. More advanced samplers such as multiple birth and death [7, 6] or jump diffusion dynamics [19] have been proposed to speed up the convergence. Unsurprisingly, GPU based parallelization has also been exploited to reduce the computation time [27, 28]. Although the efficiency of these advanced samplers is improved over the standard RJMCMC, the computational expense for large scale problems is still problematic, as we will show experimentally.

The object detection framework introduced in this paper also utilizes the point process formulation, in which the spatial distribution of objects and their detection scores are globally modelled. However rather than using slow MCMC based inference, we show that the point process inference can be solved efficiently using advanced discrete energy minimization (*e.g.* [15]) without performance degradation.

Many object detection systems still rely on the heuristic non-maximum suppression (NMS) algorithm to eliminate false positive detections. Basically NMS works by sequentially extracting local maxima which hopefully correspond to the underlying objects. Clearly, NMS fails to tackle overlapping objects, which happens frequently in crowded scenes. In contrast, our method overcomes the limitations of NMS by encoding spatial object constraints (e.g., overlap) into the point process density function, which is then solved globally.

Actually there has been work proposed to tackle the limitations of NMS previously, e.g., [3, 5]. Similar to ours, these methods optimise a global energy (objective) function which jointly models detection scores and inter-object relations. However what makes our work different from [3, 5] is the scale of the problem. While [3, 5] detect only dozens of objects from the images, we extract thousands of object instances. This often leads to energy functions with millions of variables, which can not be solved efficiently and effectively by using greedy optimization methods as done in [3, 5].

Large-scale object detection is clearly also closely related to object counting problems such as [21, 16, 1] in which the objective is to determine the number of objects present. Obviously our detection results can be used for counting purposes. However, unlike [21, 16, 1] where only the quantity of objects is estimated, our method returns also object locations which can be used for analysing object spatial distributions, or object tracking.

3. Spatial Point Process for Object Detection

A spatial point process (SPP) is a random collection of points describing phenomena occurring at random locations. A spatial point process can be formulated as a finite-set-valued random variable $\mathbf{u} = \{u_1, u_2, \ldots, u_n\}$ and what distinguishes a SPP from a random vector is that the number of constituent variables is random and the variables themselves are random, distinct and unordered. A statistical function describing a point process $p(\mathbf{u})$ is a combinatorial probability density function which consists of a discrete cardinality distribution, and a family of probability densities of the constituent variables, respectively. We refer readers to [2] for details of spatial point processes.

Clearly the point process formulation is a principled approach for problems with unknown cardinality and states, and is therefore well-suited to modelling the large-scale object detection problem in which we seek to simultaneously estimate the number of objects and their state parameters (*i.e.* locations and bounding box sizes). More formally, let $u_i \in \mathbf{U} \subset \mathbb{R}^d$ be i^{th} object state where $\mathbf{U}$ denotes a state space describing the object's location and shape (*e.g.* the class of bounding boxes with different widths and heights). Then, the problem becomes to estimate the optimal subset (object configuration) $\mathbf{u}^* = \{u_1, u_2, \ldots, u_m\} \subset \mathbf{U}$ (from a finite set of feasible object configurations generated by $\mathbf{U}$), that best describes the image data $\mathcal{D}$. For most applications, an optimal object configuration $\mathbf{u}^*$ should satisfy two main properties. First, each object u_i must be *attractive*—reflecting the true object in the image. Second, the objects should follow a favoured spatial pattern (e.g., minimal overlaps). The latter constraint implicitly forbids duplicated objects in the solution.

The optimal subset $\mathbf{u}^*$ can be attained by solving the following combinatorial MAP problem:

$$\mathbf{u}^* = \underset{\mathbf{u} \subset \mathbf{U}}{\operatorname{argmax}}\, p(\mathbf{u}|\mathcal{D}), \tag{1}$$

where $p(\mathbf{u}|\mathcal{D})$ is the posterior distribution of the object configuration, given the image data $\mathcal{D}$. For notational simplicity, we drop the input image data $\mathcal{D}$ and simply denote $p(\mathbf{u}|\mathcal{D})$ as $p(\mathbf{u})$.

To model populations of objects, Markov point processes [2, 26, 27, 13] are widely used. Markov point processes model pairwise interactions between objects giving rise to global spatial patterns and effectively control the number of objects (cardinality). A commonly used [27, 13] Markov point process density function for modelling populations is the Gibbs distribution:

$$p(\mathbf{u}) \propto \prod_{u_i \in \mathbf{u}} \psi(u_i) \prod_{(u_i \sim u_j) \in \mathbf{u}} \phi(u_i, u_j), \tag{2}$$

where $\psi(u_i)$ is the density function representing data term, and $\phi(u_i, u_j)$ is the interaction function between neighbouring objects $u_i \sim u_j$. The corresponding Gibbs energy is

$$E(\mathbf{u}) = \sum_{u_i \in \mathbf{u}} D(u_i) + \sum_{\substack{u_i \sim u_j \\ u_i, u_j \in \mathbf{u}}} V(u_i, u_j) \tag{3}$$

$$\text{such that} \quad p(\mathbf{u}) \propto e^{-(E(\mathbf{u})+\lambda)},$$

where λ is a positive constant used to ensure $E(\mathbf{u}) + \lambda \geq 0$. The functions D and V are the unary and pairwise interaction energies respectively. Typically, $D(u_i)$ computes the confidence of u_i being a true object. Without loss of generality we assume that the values of function $D(.)$ have been normalized to $[-d, +d]$, in which smaller values indicate better object hypotheses. $V(u_i, u_j)$ measures spatial pattern cost, *i.e.*

$$V(u_i, u_j) = \begin{cases} g(\mathcal{R}(u_i, u_j)) & \text{if } \mathcal{R}(u_i, u_j) < T_o \\ K & \text{if } \mathcal{R}(u_i, u_j) \geq T_o, \end{cases} \tag{4}$$

where $\mathcal{R}(u_i, u_j) \in [0, 1]$ evaluates spatial consistency between u_i and u_j, $T_o \in [0, 1]$ is a tolerance threshold, and g is a non-negative monotonically increasing function (we used $g(x) = x$), and K is a constant number. Note that K needs to be a very large number to forbid spatial object inconsistencies. In Sec. 4, we provide a theoretically justifiable way of selecting K, so as to speed up the optimization without introducing extra errors. Intuitively, u_i and u_j are disallowed from appearing together if they are spatially inconsistent with respect to the threshold T_o, otherwise they could be both selected by paying a cost $g(\mathcal{R}(u_i, u_j))$. A typical example of $\mathcal{R}$ is the degree of overlapping between u_i and u_j, and solutions with strong object overlaps (*e.g.* $T_o \geq 0.5$) should not be accepted.

Consequently the optimal object configuration can be equivalently calculated by

$$\mathbf{u}^* = \underset{\mathbf{u} \subset \mathbf{U}}{\operatorname{argmin}} E(\mathbf{u}) \tag{5}$$

Since there is no analytical solution for solving (5), previous methods [7, 6, 27, 28] resorted to simulated annealing coupled with sampling techniques (e.g., MCMC). However such simulations are slow in practice. In the following we show that the continuous object state space $\mathbf{U}$ can be discretized without losing much information, and subsequently the point process inference can be solved efficiently using global discrete optimization techniques.

4. Efficient Point Process Inference

4.1. State Space Discretization

Recall that each object u_i is described by a set of state parameters including its location and shape, *i.e.* $u_i \in \mathbf{U} \subset \mathbb{L} \times \mathbb{S}$. $\mathbb{L}$ and $\mathbb{S}$ are location and shapes space respectively, for example $\mathbb{L} = \mathbb{R}^2$, $\mathbb{S} = \mathbb{R}^s$, where s is the object shape dimension (*e.g.* two for boxes, three for ellipses). As optimizing the object configuration $\mathbf{u}$ over the continuous space $\mathbf{U}$ is difficult, we instead perform a fine discretization of $\mathbf{U}$ ($\mathbb{L}$ and $\mathbb{S}$) so that it permits the usage of efficient discrete optimization (see Sec. 4.2).

Specifically, we approximate the location space $\mathbb{L}$ by the discrete image space (*i.e.* pixel locations), *i.e.* $\mathbb{L} \approx \{1, 2, \ldots, W\} \times \{1, 2, \ldots, H\}$; W and H are width and height of the image. The object shape can also be discretized similarly—for instance considering the bounding box shape, $\mathbb{S} \approx \{w_{min}, w_{min} + 1, \ldots, w_{max}\} \times \{h_{min}, h_{min} + 1, \ldots, h_{max}\}$, in which $w_{min}, h_{min}, w_{max}, h_{max}$ are the minimum and maximum width and heigh of objects. (See Sec. 5 for specific discretizations for different types of objects). We will show in Sec. 5 that such a fine discretization does not really affect the detection performance.

Consequentially, the state space $\mathbf{U}$ can be sufficiently approximated by a finite set of all possible object hypotheses $\hat{\mathbf{U}} = \{u_1, u_2, \ldots, u_N\}$ (by combining all possible locations and shapes). Note that N can be huge – in the example above $N = W \times H \times (w_{max} - w_{min}) \times (h_{max} - h_{min})$. The object detection becomes

$$\mathbf{u}^* = \underset{\mathbf{u} \subset \hat{\mathbf{U}}}{\operatorname{argmin}} \quad E(\mathbf{u}). \tag{6}$$

4.2. Binary Energy Minimization

Given a set of all possible object proposals $\hat{\mathbf{U}}$ (with a fixed ordering), denote $X = [x_1, x_2, \ldots, x_N]$ as a vector of binary variables in which $x_i = 1$ denotes that proposal u_i participates in the object configuration $\mathbf{u}$. We define the following binary energy function

$$E(X) = \sum_{i=1}^{N} D(u_i) x_i + \sum_{x_i \sim x_j} V(u_i, u_j) x_i x_j. \tag{7}$$

x_i and x_j are linked (indicated with the notation $x_i \sim x_j$) if the corresponding u_i and u_j are linked, $u_i \sim u_j$ (for example, if they are neighbors). It is clear that minimizing the energy $E(X)$ (7) corresponds to minimizing the energy $E(\mathbf{u})$ over $\hat{\mathbf{U}}$. As a result, the optimal object configuration can be obtained by solving:

$$X^* = \arg\min_X E(X). \tag{8}$$

Unfortunately, since the energy (7) is non-submodular [1], minimizing (7) is NP-hard. In our large-scale object detection problem, the dimension of X can be extremely large (up to millions of variables depending on the image sizes), ruling out standard quadratic programming solvers. Thus we resort to the local approximation based trust-region method [15] due to its proven effectiveness and efficiency. However care must be taken when using [15] for large problems as its accumulated approximation errors could lead to unsatisfactory results as well as slow convergences, as we will show shortly.

4.2.1 LSA Trust-region Optimization

Trust-region methods are a class of optimization algorithms that iteratively optimize an approximate energy function constructed near the current best solution within a "trust" region. The approximate functions should be chosen such that they are "close" to the true energy function and can be optimized efficiently. The convergence will be reached when the improvement is too subtle.

Actually what makes the energy (7) hard to optimize is the nonsubmodular quadratic terms (*i.e.* $V(u_i, u_j)x_i x_j$). Inspired by [15], we approximate them by submodular

[1] $\theta_{ij}(1,1) + \theta_{ij}(0,0) \geq \theta_{ij}(0,1) + \theta_{ij}(1,0)$, where $\theta_{ij}(x_i, x_j) = V(u_i, u_j)x_i x_j$.

functions (*e.g.* linear functions). In particular, letting $X^t = [x_1^t, x_2^t, \ldots, x_N^t]$ be the current solution, we construct the following energy

$$E^t(X) = \sum_{i=1}^{N} D(u_i)x_i + \sum_{x_i \sim x_j} A(x_i, x_j | x_i^t, x_j^t), \quad (9)$$

where

$$A(x_i, x_j | x_i^t, x_j^t) = \frac{1}{2}V(u_i, u_j)x_j^t x_i + \frac{1}{2}V(u_i, u_j)x_i^t x_j. \quad (10)$$

It can be seen that the pairwise terms have been decomposed into linear terms, and the energy $E^t(X)$ can be rewritten as:

$$E^t(X) = \sum_{i=1}^{N}[D(u_i) + \sum_{x_j \sim x_i} \frac{1}{2}V(u_i, u_j)x_j^t]x_i. \quad (11)$$

Given X^t, the next solution X^{t+1} can be obtained by minimizing the following energy function

$$L^t(X) = E^t(X) + \lambda_t ||X - X^t||. \quad (12)$$

$||.||$ is Hamming distance. The parameter λ_t controls the trust region size, which is updated at each iteration based on the quality of the current solution. Notice that the local approximate energy function (12) is linear and contains no constraints, thus minimizing $L^t(X)$ can be done efficiently using simple min operators.

Nevertheless the performance (accuracy and efficiency) of the trust region method highly depends on how accurate $E^t(X)$ approximates $E(X)$ at each iteration. If the approximation is poor, one needs to tighten the trust region. Consequently the algorithm might either get stuck at bad local optimum or take an enormous number of iterations before convergence. For our problem we note that the approximation error is proportional to the value of function V (see Eq. (10)), which can be arbitrarily large (see Eq. (4)).

Recall that the function $V(u_i, u_j)$ measures the spatial inconsistency between two objects u_i and u_j. If u_i and u_j are inconsistent (with respective to some threshold), V pays a very large penalty K to prevent u_i and u_j from appearing together in the solution. Thus, the approximation error mainly depends the value of K.

One naive way to soften the hurdles of the above approximation errors is to carefully choose a small K which still guarantees inter-object constraints. However manually picking a proper K is tedious and time-consuming. Alternatively, one could use training data to learn the value of K, however for large-scale object detection problems, ground truth data is not always available.

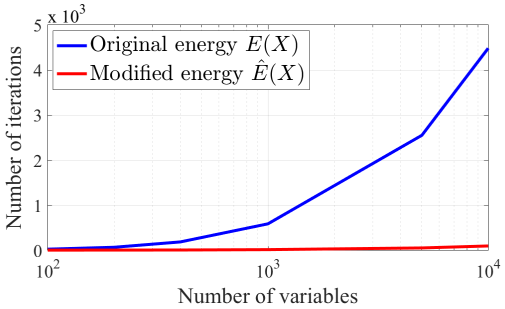


Figure 2. Comparing the numbers of iterations before convergences when optimizing the two energy functions $E(X)$ and $\hat{E}(X)$, respectively.

4.2.2 Adaptive Pairwise Energies

Here we propose to a simple way for automatically computing the smallest inconsistent penalty for u_i and u_j, but still guaranteeing valid solutions. In particular, we adaptively adjust $V(u_i, u_j)$ based on the qualities of u_i and u_j, and will show that such modification does not change the global optimality. Specifically we define

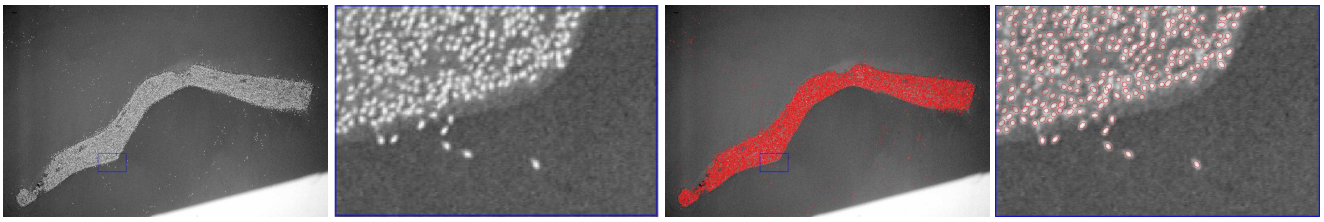
$$\hat{V}(u_i, u_j) = \begin{cases} g(\mathcal{R}(u_i, u_j)) & \text{if } \mathcal{R}(u_i, u_j) < T_o \\ \alpha & \text{if } \mathcal{R}(u_i, u_j) \geq T_o, \end{cases} \quad (13)$$

where $\alpha = \max(|D(u_i)|, |D(u_j)|) + \epsilon$; ϵ is a small positive number; We set $\epsilon = 0.001$. The corresponding modified energy function is

$$\hat{E}(X) = \sum_{i=1}^{N} D(u_i)x_i + \sum_{x_i \sim x_j} \hat{V}(u_i, u_j)x_i x_j. \quad (14)$$

Proposition 1. *If X^* is the globally minimal solution of the energy function (14), X^* is also the global minimizer of the function (7), and vice versa.*

The proof is given in the supplementary material. Basically, the proposition 1 reveals that the two energy functions $E(X)$ (7) and $\hat{E}(X)$ (14) admit the same globally optimal solution. To demonstrate the advantage of optimizing $\hat{E}(X)$ over $E(X)$, we synthetically generate energy functions of different sizes ranging from 100 to 10000 variables. For each problem size, we run the LSA trust-region optimisation on the energies $\hat{E}(X)$ and $E(X)$ respectively, and record the numbers of iterations before convergences. Fig. 2 shows the difference, where as expected, optimising energy $\hat{E}(X)$ requires much less number of iterations than that of $E(X)$. Also we observe that $E(\hat{X}^*)$, on average, are slightly lower than $E(X^*)$, where $\hat{X}^*$ and X^* are the solutions of minimizing $\hat{E}(\cdot)$ and $E(\cdot)$ respectively.



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

Figure 3. Flamingo detection and counting from remotely sensed images. (a) and (b) display an input image and a cropped region, respectively. Our algorithm is able to detect 10687 flamingoes in 13 seconds (the ground truth is 10800). (c) and (d) show qualitative results, where ellipses indicate detected objects.

Data	Methods	Detection	True Pos.	False Pos.	False Neg.	Precision	Recall	Time (s)
Bird colony GT = 10800 $r_{min} = 1, r_{max} = 4$	OURS	10678	-	-	-	-	-	**13.6**
	MBD [6]	9891	-	-	-	-	-	897.3
	PMC [28]	11280	-	-	-	-	-	187.32
	MBD*	10154	-	-	-	-	-	813.3
	PMC*	10903	-	-	-	-	-	266.6
Bird colony small GT = 148 $r_{min} = 1 r_{max} = 4$	OURS	153	146	7	2	95.4	**98.6**	**0.29**
	MBD	129	126	3	22	97.6	85.1	37.3
	PMC	153	139	14	9	90.55	93.99	10.74
	MBD*	148	143	5	5	96.6	96.6	10.1
	PMC*	137	133	4	15	97.1	89.9	9.6
Stomata GT = 676 $r_{min} = 2, r_{max} = 5$	OURS	750	627	123	49	83.6	**92.7**	**1.4**
	PMC	707	613	94	63	**86.73**	90.65	64.21
	PMC*	716	560	156	116	78.2	82.4	168.0
Cells GT = 500 $r_{min} = 6, r_{max} = 10$	OURS	479	479	0	21	**100**	95.8	**2.57**
	MBD	440	436	4	64	99.0	87.2	433.6
	PMC	483	463	20	37	95.92	92.54	60.44
	MBD*	447	447	0	53	**100**	89.4	104.5
	PMC*	482	480	2	20	99.6	**96**	7.5
Yellow Cabs GT = 100 $r_{min} = 2, r_{max} = 6$	OURS	130	87	43	12	84.38	81.82	**0.8**
	MBD*	-	-	-	-	-	-	-
	PMC*	86	81	5	19	**94.2**	81.0	165.0

Table 1. Quantitative results on large-scale object detection. Note that MBC* and PMC* indicate the results obtained from [28]. For each image, the best results are **boldfaced**. It can be seen that our method is an order of magnitude faster than others. Also in most cases our method is more accurate, except the Yellow Cabs image. This is because the unary term we used (15) which relies on simple intensity contrast does not robustly detect yellow objects. In contrast we believe that the unary model used in [28] is much stronger, but this is not detailed in [28]. GT = Ground truth number of objects; r_{min}, r_{max} are the minimum and maximum radii of objects respectively.

5. Experimental Results

5.1. 2D Parametric Object Detection

We first compare the performance of our method against the state-of-the-art point process inference method using parallel Monte Carlo [28], denoted as PMC. While our method is implemented using MATLAB and CPU, PMC used C++ and GPU parallel implementation. We also include multiple births and deaths (MBD) method [6] for comparison though their low performance relative to PMC has been reported in [28]. We used the benchmarking datasets [28] for experiments. These datasets are equipped with ground truth information so that the accuracy can be measured. The objects of interest in these images are birds, cells, stomata and yellow cabs, which can be modelled using ellipses. The unary energy is defined as:

$$D(u) = \begin{cases} 1 - \frac{d_u}{d_0} & \text{if } d_u < d_0 \\ \exp(-\frac{d_u - d_0}{d_0}) - 1 & \text{if } d_u \geq d_0, \end{cases} \quad (15)$$

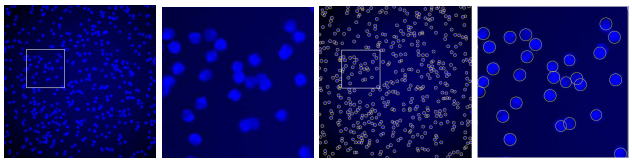
where d_u is the contrast between object u and background (computed as the Bhattacharyya distance between the inside and outside rings of the object [28]), d_0 is a tuning parame-

Methods	Lempitsky (L1-reg.) [21]				Lempitsky (Tikhonov-reg.) [21]				OURS
	N=2	N=4	N=8	N=16	N=2	N=4	N=8	N=16	
Mean Absolute Error	7.96	7.02	6.76	4.81	5.27	4.99	4.92	**4.23**	4.7
Standard Deviation	7.28	6.66	6.54	4.22	4.69	4.54	4.45	3.64	**3.5**
Average Time (s)	0.86	0.86	0.86	0.86	0.86	0.86	0.86	0.86	**0.31**

Table 2. Performance comparison results for counting bacterial cells in fluorescence-light microscopy images. The method [21] comes with two different regularizations, *i.e*. L1 and Tikhonov. N is the number of training images. All the methods are tested on 100 cell images. The best results are **boldfaced** (smaller is better). It is clear that our method not only performs better but also is more stable, with less computational cost. Note that the main computation cost of Lempitsky and Zisserman's method is for feature extraction.

(a) Our method (b) NMS

Figure 5. Crowd human detection qualitative performance comparison. (a) and (b) display the detection results returned by our method and NMS respectively. At about 67% recall, our precision is 82.84% while the precision of NMS is only 61.08%. Only heads are shown.



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

Figure 4. Bacterial cell detection and counting from fluorescence-light microscopy images. (a) and (b) display an input image with 500 cells and a cropped part, respectively. (c) and (d) show qualitative results, where ellipses indicate detected cells. Our method correctly detects 479 cells in less than 3 seconds.

ter. The non-overlapping constraint is imposed on the object configuration, *i.e*.

$$\mathcal{R}(u_i, u_j) = \frac{\mathcal{A}(u_i \cap u_j)}{\min(\mathcal{A}(u_i), \mathcal{A}(u_j))}. \quad (16)$$

$\mathcal{A}(u_i)$ returns the area of u_i. The angles of ellipses are selected from a range $[0, \pi/8, 2\pi/8, \ldots, \pi]$, and the ranges of radii for different objects are given in Tab 1.

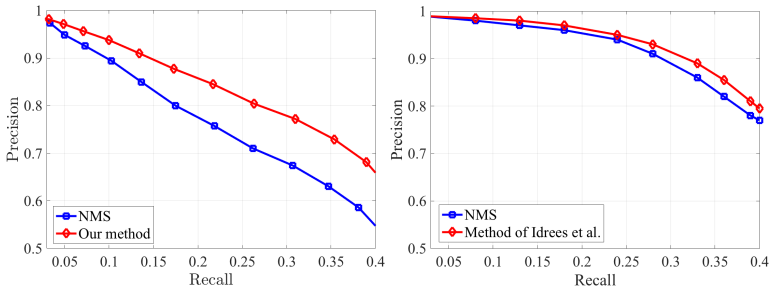
The comparison results are given Tab. 1. It is clear that our method is significantly faster than the competitors while our accuracies are comparable, if not superior. Fig. 3 shows an example of detecting flamingoes using our method.

5.2. Object Counting

As our method can be used for object counting, we apply our method for counting bacterial cells in fluorescence-light microscopy images [20]. Cells are modelled using ellipses. We compare against the learning based method [21]. Note that the method in [21] only returns an estimated number of cells in each image, whereas our method additionally gives cell locations, which are useful for cell tracking. Moreover our method does not require any training. The results reported in Tab. 2 show that our method is not only more accurate but also more stable than [21]. The method [21] only performs better when using more training data. Furthermore it can be seen that our method is very efficient, which takes about 0.3 seconds per image. Fig. 4 shows a sample of qualitative cell detection results using our method.

5.3. Crowd Human Detection

Crowd human detection and counting is another interesting problem which has many real-world applications such as event management (*i.e*. protests, marathons), video surveillance and anomaly detection. Here we aim to test the performance of our algorithm on detecting humans in crowd scenes. We used the UCF-HDDC dataset recently published in [17] for evaluations. In this application, human are rep-



(a) (b)

Figure 6. The graphs report quantitative comparison results for the large-scale human detection application using UCF-HDDC dataset [17]. (a) compares our method against NMS. (b) shows the improvement of the global occlusion reasoning [17] over NMS (the results are taken from [17].) Note that the set of human proposals in our experiment is different from [17], which leads to the difference in the overall accuracies.

resented as bounding boxes. Unlike previous applications, which only consider the object overlap, the pairwise potential functions here jointly penalize both strong object overlap and scale inconsistency between nearby objects. In particular $\mathcal{R}(u_i, u_j)$ is defined as below:

$$\mathcal{R}(u_i, u_j) = \frac{\mathcal{A}(u_i \cap u_i)}{\min(\mathcal{A}(u_i), \mathcal{A}(u_j))} + \left(1 - \min(\frac{s_{u_i}}{s_{u_j}}, \frac{s_{u_j}}{s_{u_i}}\right) \exp(\frac{-d(u_i, u_j)}{\sigma_p}), \tag{17}$$

where s_{u_i} is the scale of object u_i, $d(u_i, u_j)$ computes the Euclidean distance between the centres of u_i and u_j, σ_p is the deviation threshold (we set $\sigma_p = 200$). As full bodies are hardly visible in crowd images, we adapt the DPM human detector [11] to detect combination-of-parts (CoP), namely upper bodies and head-shoulders, as done in [17]. Also similar to [17] we re-score the detections using confidence and scale priors. We refer readers to [17] for details. For each object proposal, the unary function is defined as $D(u_i) = -S(u_j)$, where $S(u_i)$ is the detection score.

We compare our global point process object selection algorithm against the standard local non-maximal suppression (NMS). Both methods take the same set of object proposals (head bounding boxes only) as input. Fig. 5 shows a sample of qualitative comparison results between the two methods. For this image, we select a detection threshold for each method such that both methods have approximately the same recall. Our corresponding precision is 82.84% while that of NMS is only 61.08%. Notice that NMS returns many false positives, and also its detection scales are not globally consistent. In contrast, the scales of our detections change gradually.

Quantitative comparison results over 100 test images are reported in Fig. 6(a). As expected, our result is clearly superior to that of NMS, providing a boost in performance of around 10% for recall values greater than 0.25. Ideally, we would also compare our method directly against that of [17], but this is not possible because of the different factors in particular closed implementations of CoP and scale estimation that contribute to their overall result. Instead we show in Fig. 6(b), their result against NMS, taken directly from [17]. The salient points to note here are that: (i) their improvement over the NMS baseline is considerably lower than ours, at around only 3-4%; (ii) our method is orthogonal to the value added by the better proposals that are the key to their good performance. We therefore believe that with access to [17]'s object proposals, our method would provide a further boost in performance.

6. Conclusion

We have proposed a general framework for large-scale object detection. We formulate the object detection problem using a point process probabilistic model whose density function includes object confidences and spatial object patterns. These two terms can be arbitrarily defined depending on the specific applications. As the point process inference is difficult and expensive, we developed a highly efficient point process inference based on a fine discretization of the object state space and discrete energy minimization. We showed that our algorithm is just as accurate, but significantly faster than a state-of-the-art point process inference that uses a GPU implementation. We also demonstrated the superior performance of our algorithm over the standard non-maximal suppression (widely used for object detection) using a crowd human detection application. As our framework is general, it could be extended to detect objects in 3D or higher dimensional spaces, which we consider in our future work.

Acknowledgements

This research was supported by the Australian Research Council through the Centre of Excellence for Robotic Vision (CE140100016), Laureate Fellowship (FL130100102) to IDR and Discovery Project DP160103490.

References

[1] C. Arteta, V. Lempitsky, J. Noble, and A. Zisserman. Interactive object counting. In *Computer Vision ECCV 2014*, volume 8691 of *Lecture Notes in Computer Science*, pages 504–518. 2014. 3

[2] A. J. Baddeley and M. N. M. V. Lieshout. Stochastic geometry models in high-level vision. *Journal of Applied Statistics*, 20(5-6):231–256, 1993. 1, 2, 3

[3] O. Barinova, V. Lempitsky, and P. Kohli. On detection of multiple object instances using hough transforms. In *CVPR*, 2010. 3

[4] A. Chan, Z.-S. Liang, and N. Vasconcelos. Privacy preserving crowd monitoring: Counting people without people models or tracking. In *CVPR*, 2008. 1

[5] C. Desai, D. Ramanan, and C. C. Fowlkes. Discriminative models for multi-class object layout. *International Journal of Computer Vision*, 95(1):1–12, 2011. 3

[6] S. Descamps, X. Descombes, A. Bechet, and J. Zerubia. Automatic flamingo detection using a multiple birth and death process. In *ICASSP*, 2008. 1, 2, 4, 6

[7] X. Descombes, R. Minlos, and E. Zhizhina. Object extraction using a stochastic birth-and-death dynamics in continuum. *J. Math. Imaging Vis.*, 33(3):347–359, 2009. 2, 4

[8] X. Descombes and J. Zerubia. Marked point process in image analysis. *Signal Processing Magazine, IEEE*, 19(5):77–84, 2002. 1

[9] P. Dollar, C. Wojek, B. Schiele, and P. Perona. Pedestrian detection: An evaluation of the state of the art. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 34(4):743–761, April 2012. 2

[10] M. Everingham, S. Eslami, L. Van Gool, C. Williams, J. Winn, and A. Zisserman. The pascal visual object classes challenge: A retrospective. *International Journal of Computer Vision*, 111(1):98–136, 2015. 2

[11] P. Felzenszwalb, R. Girshick, D. McAllester, and D. Ramanan. Object detection with discriminatively trained part-based models. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 32(9):1627–1645, Sept 2010. 1, 2, 8

[12] P. Felzenszwalb, D. McAllester, and D. Ramanan. A discriminatively trained, multiscale, deformable part model. In *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pages 1–8, June 2008. 1

[13] A. Gamal-Eldin, X. Descombes, and J. Zerubia. Multiple birth and cut algorithm for point process optimization. In *SITIS*, 2010. 2, 3

[14] W. Ge and R. Collins. Marked point processes for crowd counting. In *CVPR*, 2009. 1

[15] L. Gorelick, Y. Boykov, O. Veksler, I. B. Ayed, and A. Delong. Submodularization for binary pairwise energies. In *Proceedings of the 2014 IEEE Conference on Computer Vision and Pattern Recognition*, CVPR '14, pages 1154–1161, Washington, DC, USA, 2014. 2, 3, 4

[16] H. Idrees, I. Saleemi, C. Seibert, and M. Shah. Multi-source multi-scale counting in extremely dense crowd images. In *Proceedings of the 2013 IEEE Conference on Computer Vision and Pattern Recognition*, CVPR '13, pages 2547–2554, Washington, DC, USA, 2013. IEEE Computer Society. 3

[17] H. Idrees, K. Soomro, and M. Shah. Detecting humans in dense crowds using locally-consistent scale prior and global occlusion reasoning. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 37(10):1986–1998, Oct 2015. 1, 7, 8

[18] C. Lacoste, X. Descombes, and J. Zerubia. Point processes for unsupervised line network extraction in remote sensing. *IEEE TPAMI*, 27(10):1568–1579, 2005. 2

[19] F. Lafarge, G. Gimel'farb, and X. Descombes. Geometric feature extraction by a multimarked point process. *IEEE TPAMI*, 32(9):1597–1609, 2010. 1, 2

[20] A. Lehmussola, P. Ruusuvuori, J. Selinummi, H. Huttunen, and O. Yli-Harja. Computational framework for simulating fluorescence microscope images with cell populations. *Medical Imaging, IEEE Transactions on*, 26(7):1010–1016, July 2007. 1, 7

[21] V. Lempitsky and A. Zisserman. Learning To Count Objects in Images. In *NIPS 23*, 2010. 3, 7

[22] M. Ortner, X. Descombes, and J. Zerubia. Building outline extraction from digital elevation models using marked point processes. *IJCV*, 72(2):107–132, 2007. 2

[23] G. Perrin, X. Descombes, and J. Zerubia. Adaptive simulated annealing for energy minimization problem in a marked point process application. In *EMMCVPR*, 2005. 1

[24] R. Stoica, X. Descombes, and J. Zerubia. A gibbs point process for road extraction from remotely sensed images. *IJCV*, 57(2):121–136, 2004. 2

[25] A. Utasi and C. Benedek. A 3-d marked point process model for multi-view people detection. In *CVPR*, 2011. 1

[26] N. Van Lieshout. *Markov Point Processes and Their Application*. Imperial College Press, 2000. 3

[27] Y. Verdié and F. Lafarge. Efficient monte carlo sampler for detecting parametric objects in large scenes. In *ECCV*, 2012. 2, 3, 4

[28] Y. Verdi and F. Lafarge. Detecting parametric objects in large scenes by monte carlo sampling. *International Journal of Computer Vision*, 106(1):57–75, 2014. 1, 2, 4, 6

[29] Y.-x. Yuan. A review of trust region algorithms for optimization. In *ICIAM 99 (Edinburgh)*, pages 271–282. Oxford Univ. Press, Oxford, 2000.