Unbiased population size estimation on still gigapixel images

Abstract

Population sizing is essential in sociology and in various other real life applications. Gigapixel cameras can provide high resolution images of an entire population in many cases. However, exhaustive manual counting is tedious, slow and difficult to verify, whereas current computer vision methods are biased and known to fail for large populations. A design unbiased method based on geometric sampling has recently been proposed. It typically requires only between 50 and 100 manual counts to achieve relative standard errors of $5\% - 10\%$ irrespective of population size. However, the large perspective effect introduced by gigapixel images may boost the relative standard error to $30\% - 40\%$. Here we show that projecting the sampling grid from a map onto the gigapixel image using the camera projection, neutralizes the variance due to perspective effects and restores the relative standard errors back into the $5\% - 10\%$ range. The method is tested on 6 simulated images. A detailed step-by-step illustration is provided with a real image of a 30000 people crowd.

Index Terms

Crowd size, population size, design unbiased method, geometric sampling, demonstration, political rally, gigapixel image.

I. Introduction

A population is a finite set of separate items or "particles" of interest. Determining the total number of particles in the population i.e. the population size, N, is important in many real world problems. Human crowd sizing is relevant to sociology (e.g. demonstrations, political rallies, concerts, marathons, etc), whereas herd, flock, or swarm sizes, are useful in ecology.

The traditional density method [1], [2] is still widely used for crowd sizing. However it has high and unpredictable errors and is often not verifiable. Frequently, size estimates differ widely among convention organizers, media, and police.

High resolution photography has made it possible to image the entire population or at least an important sample of it [3], so that the particles are unambiguously distinguishable for counting. However, for population sizes above a few thousands, manual counting in still images is too laborious, slow, observer dependent, and difficult to verify.

A number of computer vision methods have been proposed in the particular case of human crowd sizing on still images (see for instance [4]–[8]). Unfortunately they are unable to handle crowd sizes above a few thousands and there is still a ten fold improvement to be made before reaching human-based performance [9]. The application of these methods to ecology needs spefic remodelling for the particular target population (see for instance [10]). It is well known that estimators have two sources of error, namely variance and bias. Automatic computer vision

methods are biased since their error is due to systematic errors (e.g. false detections, false negatives) which do not have zero expected value and are unpredictable.

Recently, [11] proposed an unbiased population size estimation method (hereafter *CountEm* method, with reference to the free software available at http://countem.unican.es) that can be applied to any kind of particle. It was adapted from well known principles of geometric sampling for stereology, which are widely used in many disciplines (see [12]–[14] and references therein). The main tool is systematic sampling with a uniform random (UR) grid of quadrats. The standard choice is a square grid of square quadrats defined by two parameters, namely the separation, T, between quadrat centers and the quadrat side length t, $(0 < t \leq T < \infty)$ (see Fig. 1). The total number N of particles in a given image is estimated by the total number Q of units captured by the quadrats, times the sampling period T^2/t^2. The estimator $\widehat{N}$ is unbiased, which means that for any pair (t, T) the mean of the error $\widehat{N} - N$ over all potential UR superimpositions of the grid is equal to zero, as long as the sampling units are distinguishable for counting. Thus, unbiasedness is a mathematical property which renders empirical validation unnecessary in the absence of observation artifacts. Error variance predictors from a single sample, however, are generally not unbiased approximations, and require testing.

The *CountEm* estimator was tested in [11] for two annotated crowd images, namely the spectators image which can be downloaded at http://countem.unican.es and Fig. 7b of [6]. The number of annotated humans in these images was 1220 and 4633 respectively. Monte Carlo resampling showed that manually counting about $50(100)$ individuals sampled in about 20 non-empty quadrats, can yield relative standard errors of about $8\%(5\%)$. With such small sample sizes, *CountEm* is an attractive alternative to automatic methods, which are hitherto biased to unknown degrees. The resampling experiments were mainly intended to test the performance of a new error variance prediction formula (*Cavalieri* estimator, $\mathrm{var}_{\mathrm{Cav}}\left(\widehat{N}\right)$, Eq. 3 of [11]) which is also tested for gigapixel images in the present paper.

Here the *CountEm* method is implemented and tested on a 51350×21078 pixels gigapixel image of a demonstration at *Puerta del Sol*, Madrid taken in January 2015 by Adriano Morán (http://93metros.com). The full resolution picture can be seen in [15] whereas a low resolution version is shown in Fig. 1(c). Estimating the size of the crowd pictured in this image is challenging since it is about an order of magnitude higher than the crowd sizes considered in computer vision studies [4]–[7]. Satellite or aircraft images do usually have insufficient resolution to allow unambiguous counting. Gigapixel pictures yield high resolutions, but they are often ground based, producing inhomogeneous population patterns due to perspective effects. The *CountEm* method is unbiased irrespective of population size and pattern, but the sampling variance depends on the distribution of quadrat counts. The huge perspective artifacts shown for instance in Fig. 1(c) may significantly increase the variance. Therefore we propose to generate a UR standard grid of quadrats on the map of the area where the picture has been taken (see Figs. 2(a) and 3(a)) and transform it using the camera projection (Fig. 3(c)). The grid transformation preserves unbiasedness. Hereafter the new implementation is called *CountEm2*. Automatic resampling on a simulated point cloud (Fig. 3(c)) resembling the real image Fig. 1(c) revealed that *CountEm2* could reduce the relative standard error from about 30% (if the method in [11] is directly applied) down to $5\% - 10\%$.

The paper is organized as follows: The *CountEm* and *CountEm2* methods are outlined in section II. The projection

Fig. 1. (a): A portion of the grid of quadrats proposed in [11] for systematic sampling. The sampling period is T^2/t^2. (b): Magnified version of the quadrat marked in (c). Only the two marked heads should be counted, considering heads as counting units and applying the forbidden line counting rule [11]. (c): Low resolution version of a gigapixel picture by Adriano Morán (*93metros*) of a demonstration at Puerta del Sol in Madrid. The high resolution gigapixel image can be better appreciated in [15]. A grid of quadrats of the type shown in (a), with $T = 4000$ and $t = 200$ pixels has been superimposed with a tilt of 60°.

formulas and the camera parameter estimation are explained in section II-C. In section II-E we describe how to simulate the point clouds shown in Fig. 3. The results of the Monte Carlo resampling analysis are presented in section III-A. A robustness test is performed in section III-B, in order to assess the impact of possible projection inaccuracies. Population sizing with the real gigapixel picture is illustrated in section IV. Finally, the conclusions are given in section VI.

II. Materials and Methods

A. *Outline of the* CountEm *Method:*

The main steps of the standard *CountEm* method [11] are:

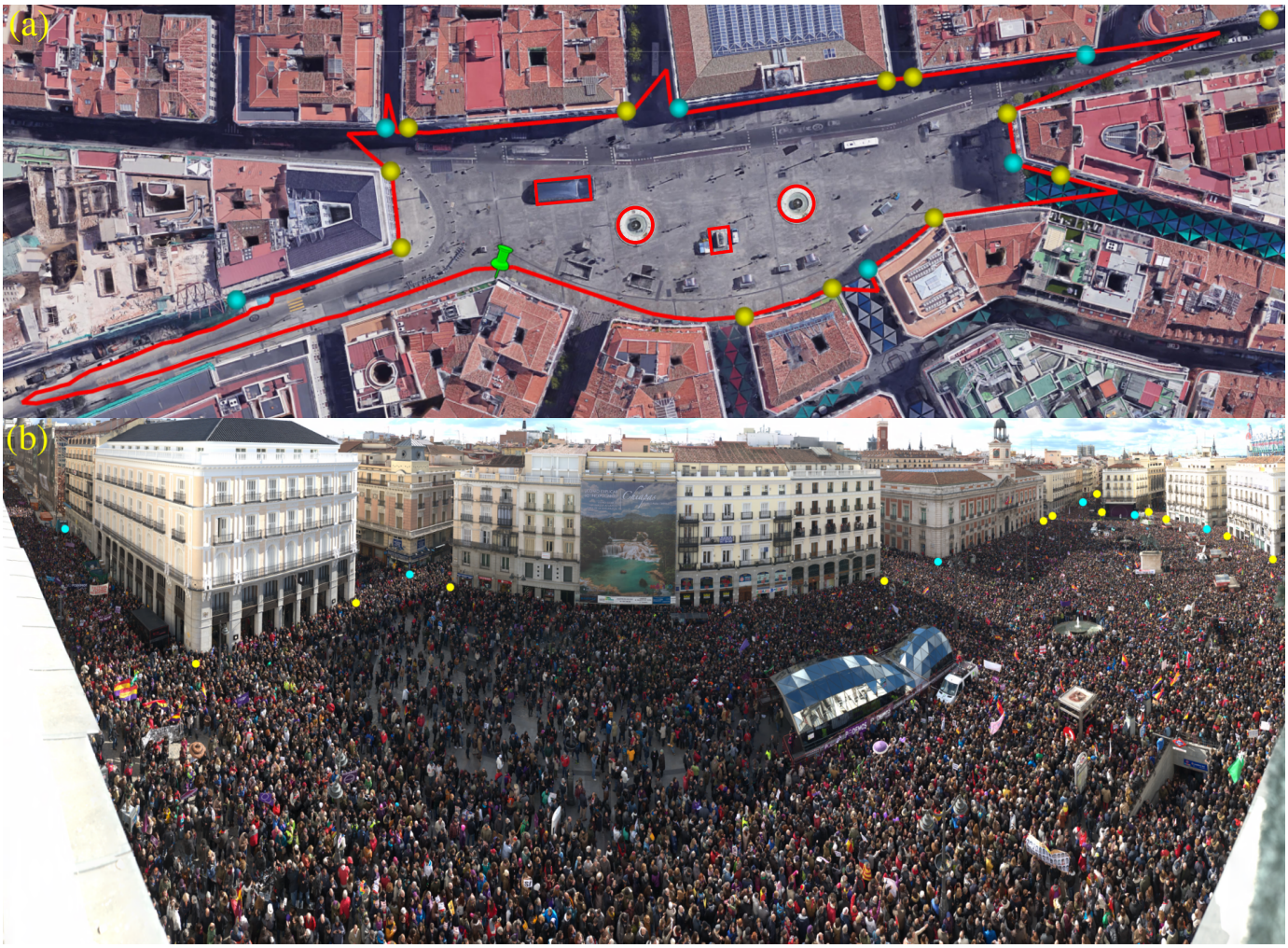

Fig. 2. (a): Aerial view of the Puerta del Sol, Madrid (PNOA, Instituto Geogrfico Nacional). Reference points for parameter estimation are marked in yellow and cyan. (b): Same image as in Fig. 1(c) with the reference points of (a).

1) Define a sampling grid by choosing parameters t and T (see Fig. 1(a)). Two practical criteria were given in [11] to design an efficient grid. First, aim at a sample size Q of about 100 particles and second, aim at counting no more than 4 or 5 particles per quadrat. In images with weak perspective effects, this tipically yields a coefficient of error below 10% irrespective of population size N. Optionally, the grid might be tilted at will a given, fixed angle in order to avoid undesirable alignments of the population pattern and the grid rows. An alignment does not affect unbiasedness but it might induce higher error variance. For instance a whole row of spectators can be included or excluded of the sample depending on the particular grid position (see Fig. 11 of [16]).
2) Superimpose the grid uniformly at random on the crowd image, e.g. Fig. 1(c).
3) Manually count the total number, Q, of particles captured by the quadrats, and multiply by the sampling period T^2/t^2:

$$\widehat{N} = \frac{T^2}{t^2} \cdot Q. \tag{1}$$

Note that the forbidden line rule should be used to avoid edge effects leading to biased counting [11], [13], [17], [18]. A particle is counted in a quadrat only if it intersects the quadrat but it does not hit the extended forbidden line of the quadrat (in red in Fig. 1(b)).

4) Estimate the sampling variance using the Cavalieri estimator proposed in Eq. 3 of [11].

B. Outline of the CountEm2 *Method:*

The basic steps of the proposed *CountEm2* population size estimation method are:

1) Identify the area occupied by the target population in a map e.g. Fig. 2(a). Note that we outline the reference area for simulation purposes and it is not necessary otherwise.
2) Apply standard *CountEm* steps 1) and 2) on the map, e.g. Fig. 3(a). We recommend to transform longitude and latitude coordinates into meters choosing an arbitrary neighboring origin, and to obtain the corresponding sea level elevations if available.
3) Project the grid onto the gigapixel picture using the camera projection parameters as explained in section II-C, e.g. Fig. 3(c) or Fig. 7.
4) Apply standard *CountEm* steps 3) and 4) to the gigapixel image with the superimposed projected grid.

C. Cylindrical projection: Cylindrical coordinates

Cylindrical projection allows a large field of view and preserves vertical lines. Therefore, it is often the preferred option for gigapixel images such as Fig. 1(c). Our aim is to project a grid of quadrats from real world $3D$ coordinates of a map onto the gigapixel image using the relevant cylindrical camera projection parameters. Hence we need to know how to project an arbitrary point in real space, $X \in \mathbb{R}^3$ with known coordinates $X = (x_1, x_2, x_3)$ in arbitrary distance units, into a point G in the gigapixel image, with coordinates $G = (g_1, g_2)$. We chose the x_1 and x_2 axes defining a horizontal ground plane with a vertical x_3-axis and the origin at the location of the observer $O \in \mathbb{R}^3$. The g_1, g_2 axes in the image follow the image analysis convention, with the g_1 axis increasing from left to right, the g_2 axis increasing downwards and the origin at the upper left corner of the image.

If the camera projection parameters are unknown, then they can be estimated using simple pinhole camera equations and applying the following steps.

First, consider a right circular cylinder of radius $r > 0$ with symmetry axis along the x_3-axis. The point X, with cylindrical coordinates $X = (R, \phi, x_3)$, is projected by a ray emanating from O into a point $C = (r, \phi, z)$ on the cylinder surface. It is easy to show that:

$$\begin{aligned} R &= \sqrt{x_1^2 + x_2^2} \\ \phi &= \operatorname{atan2}(-x_2, -x_1) \\ z &= -\frac{r}{R} x_3. \end{aligned} \tag{2}$$

Note that there are two possible intersections of the ray with the cylinder. In the latter two equations, the intersection corresponding to the minus signs is chosen in order to obtain the desired image orientation.

Unwrapping the cylinder, the dimensionality is reduced from 3 to 2, and $C \in \mathbb{R}^3$ is transformed into $P \in \mathbb{R}^2$ with coordinates $P = (p_1, p_2)$, namely:

$$\begin{aligned} p_1 &= r\phi \\ p_2 &= -\frac{r}{R}x_3. \end{aligned} \tag{3}$$

Since the origin is arbitrary, we apply a shift $\Delta = (\Delta_1, \Delta_2)$ to P in order to match the photo coordinates. Thus, each point X is mapped into a point $G = P + \Delta = (g_1, g_2)$ in the image, where:

$$\begin{aligned} g_1 &= r\phi + \Delta_1 \\ g_2 &= -\frac{r}{R}x_3 + \Delta_2, \end{aligned} \tag{4}$$

with $\phi \in (-\pi, \pi]$. The parameter vector $\theta = (\Delta_1, \Delta_2, r)$ has to be estimated in order to perform the desired projection of the quadrat grid as in Fig. 3(c).

In future applications it should be technically possible to read the projection parameters directly from the camera. The camera information was unfortunately not available for the gigapixel image Fig. 1(c). Therefore the required parameters had to be estimated as we describe below.

D. Cylindrical projection: Parameter estimation

Several reference points are needed in order to estimate the parameter vector, θ. Each reference point $G = (g_1, g_2)$ in the gigapixel image, has to be identified and matched with its corresponding real space coordinates $X = (x_1, x_2, x_3)$.

We used the 18 points represented with yellow and cyan dots in Fig. 2. The longitude and latitude coordinates of a point X can be transformed into $3D$ coordinates (x_1, x_2, x_3) in meters choosing an arbitrary origin and using for instance the Python Geocoding Toolbox and Google Earth Pro elevation data or similar tools. Their coordinates in degrees, sea level elevations in meters and corresponding (g_1, g_2) photo coordinates in pixels are listed in Table I. The point number 0 corresponds to the observer location O which was determined with Google Earth Pro and is marked with a green pushpin icon in Fig. 2(a).

An overdetermined system can be obtained from the foregoing equations, namely:

$$\begin{aligned} \operatorname{atan2}(-x_2, -x_1)\, r + \Delta_1 &= g_1 \\ \frac{-x_3}{\sqrt{x_1^2 + x_2^2}} r + \Delta_2 &= g_2, \end{aligned} \tag{5}$$

which can be written in matrix notation as $\mathbf{A}\theta = \mathbf{b}$. $\mathbf{A}$ is the coefficient matrix and b the independent term vector. An approximate solution to the overdetermined system can be found using ordinary least squares. Solving the

TABLE I

REFERENCE POINTS USED IN FIG. 2 FOR PARAMETER ESTIMATION

#	Long. (°)	Lat. (°)	Elevation (m)	g_1 (pix.)	g_2 (pix.)
0	-3.702702	40.417105	670.96	-	-
1	-3.702299	40.417011	649.88	7720	9820
2	-3.70226	40.416784	649.48	14130	7390
3	-3.702243	40.416656	649.47	16280	6250
4	-3.702331	40.416656	649.36	17900	6760
5	-3.703183	40.416614	648.41	35270	6510
6	-3.703388	40.416607	648.32	37420	5740
7	-3.704202	40.416533	647.09	41640	4110
8	-3.704306	40.416523	646.90	42000	3910
9	-3.704986	40.416462	646.52	43220	3360
10	-3.705702	40.416366	647.19	43780	3030
11	-3.704675	40.416634	646.22	43980	3760
12	-3.704697	40.416781	645.75	45290	3870
13	-3.704874	40.416821	645.31	45850	3750
14	-3.704383	40.416941	646.43	46550	4040
15	-3.704121	40.417091	647.28	48180	4440
16	-3.70398	40.417147	647.50	48990	4720
17	-3.703635	40.417227	648.36	50820	5650
18	-3.701658	40.417159	651.70	2490	4430

system with the reference points of Table I and choosing the origin at O we obtain:

$$\begin{aligned} r &= 1428.3 \text{ pixels} \\ \Delta_1 &= 2587.9 \text{ pixels} \\ \Delta_2 &= 162.3 \text{ pixels}. \end{aligned} \tag{6}$$

Note that for convenience the distance units for x_1, x_2, x_3 are meters whereas for $g_1, g_2, r, \Delta_1, \Delta_2$ we chose "pixels" (= pixel width lengths). The solution is approximate since the correspondence between (g_1, g_2) and the real space coordinates (x_1, x_2, x_3) is not exact. Hereafter, the projection using Eq. 4 with the parameters of Eq. 6 is called *basic* projection. In section III-B we show that small inaccuracies in parameter estimation do not significantly affect the results. For instance, the relative standard error increases from 7% to 9% in the considered case.

E. Crowd coordinate simulation

Assessing the precision of the estimators, requires either an annotated gigapixel image, or a realistic simulation of the particle positions. Since the former was not available, we simulated N coordinates (g_1, g_2) of a crowd, resembling the population pattern of Fig. 1(c). As the real crowd size is unknown, we chose to simulate $N = 20000$ points which seems large enough for our purpose.

First, we outlined a window in the map area where the N points had to be simulated (red line in Fig. 2(a)). The window coordinates were marked manually in Google Earth Pro following approximately the viewshed from the

observer location. Note that the viewshed is not exact since there are ledges of the building that do not appear in Google Earth Pro. Furthermore, we outlined some interior boundaries or "holes" in the window, where no people can be found, namely two fountains, a metro station and a statue.

The point pattern shown in Fig. 3(a) was simulated using the R-spatstat package [19] as shown in Appendix A. Distinct points were not allowed to come closer than 0.5 m apart. The x_3 coordinate of each of the N simulated points was determined using Google Elevation API. Each point in the simulated cloud was then projected using Eq. 4, thus obtaining the point cloud shown in Fig. 3(c). The grid was superimposed using the approximations described below (section III-B). More realistic crowd simulations are not necessary since our naive simulation adequately resembles the inhomogeneous population pattern in Fig. 1(c).

III. Results

A. Empirical assessment of the precision of the estimators through Monte Carlo resampling

Monte Carlo resampling on the simulated point cloud was used to assess the precision of $\widehat{N}$ and the performance of the $\mathrm{var}_{\mathrm{Cav}}\left(\widehat{N}\right)$ estimator. A standard grid of quadrats (i.e. *CountEm* method, see Fig. 3(b)) and a projected grid (i.e *CountEm2* method as in Fig. 3(c)) were applied with 60° tilt prior to sampling. The resampling procedure described in [11] was followed. Next we recall the necessary notation:

- $Y = \{y_1, y_2, \ldots, y_N\}$: finite set representing a population of size N in a bounded area.
- $y_i \in Y$: ith particle of the population.
- J_0: fundamental tile or box of side length T.
- $z \in J_0$: UR point in the fundamental tile.
- Λ_z: UR systematic grid of quadrats, generated by shifting the lower left corner of a quadrat from an arbitrary initial position in J_0 into the UR point z, thus dragging the whole quadrat grid together. The quadrats have side length t.
- $Q = Q\left(Y \cap \Lambda_z\right)$: sample size, namely the total number of particles captured by the quadrats.

For each pair (t, T) a total of $K^2 = 32^2 = 1024$ replicated superimpositions of the grid Λ_z onto Y were generated, corresponding to K^2 systematic replications $\{z_k, k = 1, 2, \ldots, K^2\}$ of the point z within J_0. These K^2 point locations were arranged in a random subgrid within J_0 which should be expected to be more efficient than independent random locations, as explained in [11]. Thus, $K^2 = 32^2 = 1024$ simulations are sufficient to ensure that the empirical values are very close to their respective true values.

For each k, the corresponding sample total,

$$Q_k = Q(Y \cap \Lambda_{z_k}), \tag{7}$$

was computed automatically using the R-spatstat package [19]. Hence, from Eq. 1 we obtain:

$$\widehat{N}_k = (T/t)^2 \cdot Q_k. \tag{8}$$

The empirical mean and variance of $\widehat{N}$ were computed respectively as follows,

$$\mathbb{E}_e\left(\widehat{N}\right) = K^{-2}\sum_{k=1}^{K^2}\widehat{N}_k, \tag{9}$$

$$\mathrm{Var}_e\left(\widehat{N}\right) = K^{-2}\sum_{k=1}^{K^2}\left\{\widehat{N}_k - \mathbb{E}_e\left(\widehat{N}\right)\right\}^2. \tag{10}$$

We also computed the corresponding K^2 replicates of $\left\{\mathrm{var}_{\mathrm{Cav}}\left(\widehat{N}_k\right)\right\}$ using Eq. 3 of [11]. The empirical square coefficient of error:

$$\mathrm{ce}^2_{\mathrm{Cav}}\left(\widehat{N}\right) = \frac{1}{N^2K^2}\sum_{k=1}^{K^2}\mathrm{var}_{\mathrm{Cav}}\left(\widehat{N_k}\right) \tag{11}$$

was compared with the corresponding empirical ("true") value,

$$\mathrm{CE}^2_{\mathrm{e}}\left(\widehat{N}\right) = \mathrm{Var}_e\left(\widehat{N}\right)/N^2. \tag{12}$$

The results and the considered t and T values are shown in Fig. 4. The grid units in the *CountEm* case are pixels. For *CountEm2*, sampling on the map as in Fig. 3(a) and on the picture Fig. 3(c) yield identical results assuming that the grid is transformed with the *exact* camera projection. For simplicity we chose to sample on the map and hence, in that case, t and T are in meters. In the bottom row of Fig. 4 the empirical error coefficient of the *CountEm2* method (blue line) is as expected in the $5\% - 10\%$ range, whereas the standard *CountEm* method (Fig. 4(a, b, c)) presents error coefficients in the $30\% - 40\%$ range. Thus, the projected grid is able to correct the huge variance increase due to perspective artifacts. The variance estimator (red line) exhibits a reasonable performance in all the considered cases. These statements are based on the assumption that there is no human measurement error, thus in practice the errors could be slightly higher.

B. Robustness Test: Projection

In the preceding section we assumed that the grid could be transformed using the *exact* camera projection. For simplicity, sampling was performed on the map Fig. 3(a) instead of on the picture Fig. 3(c). However, in practice some inaccuracies can arise in the calibration and/or grid projection processes. For instance, the reference points could be different, the elevation data could be inaccurate or unavailable, or the number of projected points per quadrat could be insufficient. Here we examine the impact of such inaccuracies by comparing a *simplified* grid projection with the *exact* projection, using the simulated point cloud.

The following approximations are used in the *simplified* projection which we applied to the grid:

- We assumed that, for all the quadrats of the grid, only a common, average elevation was known. For the grid projection, the elevation, x_3, was set at 648 m in Eq. 4.
- The following set of parameters was used in Eq. 4:

$$\begin{aligned} r &= 1428.6 \text{ pixels} \\ \Delta_1 &= 2584.6 \text{ pixels} \\ \Delta_2 &= 158.9 \text{ pixels}, \end{aligned} \tag{13}$$

which were estimated using only 6 reference points, numbered $3, 6, 9, 12, 15, 18$ in Table I. They are plotted in cyan color in Fig. 2(a).

- We chose to project 6 points per quadrat, namely the four vertices plus the two endpoints of the interrupted forbidden line segments. The quadrats were then approximated by connecting those 6 points with straight line segments. Note that only vertical straight lines are preserved in cylindrical projections.

On the other hand, the *basic* projection (defined in section II-D) was applied to the simulated point cloud using the elevations given by Google Elevation API.

Fig. 5 shows the results for $\widehat{N}$ and $\mathrm{ce}^2_{\mathrm{Cav}}\left(\widehat{N}\right)$ with $T = 20$ m and $t = 1.5$ m which are the parameters chosen in section IV. The results using the *exact* projection (i.e. sampling on the map) are plotted in panels (a) and (b), and the ones for the *simplified* grid projection combined with *basic* point cloud projection in (c) and (d). As it can be seen, the *simplified* projection does not introduce appreciable bias. The empirical variance increases slightly, since the relative coefficient of error is 7% for the *exact* projection and 9% for the *simplified* projection. The $\mathrm{ce}^2_{\mathrm{Cav}}\left(\widehat{N}\right)$ estimator performs only slightly worse.

C. Robustness Test: Population Size and Spatial Distribution

The crowd point pattern simulation of section II-E had population size $N = 20000$ and a homogeneous, hard core spatial distribution. Here we analyze how the projection correction does help to reduce the error in other simulated point patterns, varying N and the spatial distribution. We considered the six simulations shown in Fig. 6. They were simulated with the spatstat package [19] as described in Appendix A. The inhomogeneous distributions in Fig. 6(e)-(f) are expected to be more inhomogeneous than real crowds, since no hard core radius was considered.

The empirical values $\mathrm{CE}_{\mathrm{e}}\left(\widehat{N}\right)$ corresponding to the six simulated "gigapixel" point patterns (Fig. 6), are computed following the simulation procedure described in section III-A. The results with projection correction (*CountEm2*) are shown in Table II, whereas those obtained without projection correction (*CountEm*) are displayed in Table III. Parameters t and T were chosen in order to obtain a theoretical mean sample size of $Q = 200$. Quadrat side length was set to $t = 1.5$m and $t = 300$ pixels for *CountEm2* and *CountEm* respectively. The values of separation between quadrat centers, T, are shown in the Tables. A singular crowd size estimation $\widehat{N_1}$ with sample size Q_1 and error estimation $\mathrm{ce}_{\mathrm{Cav}}\left(\widehat{N_1}\right)$, obtained with a single superimposition of the grid of quadrats, is also shown. The mean values are given in the last row of each table. Note that as usual, $\mathrm{Mean}\left\{\mathrm{CE}\left(\widehat{N}\right)\right\} = \sqrt{\mathrm{Mean}\left\{\mathrm{Var}\left(\widehat{N}\right)\right\}}/N$.

The six *CountEm2* empirical coefficients of error, $\mathrm{CE}_{\mathrm{e}}\left(\widehat{N}\right)$, are below 11% showing the robustness of the method to variations in population size and spatial distribution. The corresponding *CountEm* values, without projection correction are above 27%.

IV. Population sizing with the gigapixel picture

The previous sections reveal that applying *CountEm2* with $T = 20$ m and $t = 1.5$ m using the *simplified* grid projection, should provide an unbiased estimator of the crowd size in Fig. 1(c) with empirical coefficient of error of around 9%. For simplicity, this was the chosen setup.

TABLE II

POPULATION SIZE ESTIMATION RESULTS WITH PROJECTION CORRECTION

Image	T	N	$\widehat{N_1}$	Q_1	$\mathrm{ce}_{\mathrm{Cav}}\left(\widehat{N_1}\right)$	$\mathrm{CE}_{\mathrm{e}}\left(\widehat{N}\right)$
(a)	150	20000	20800	208	0.039	0.062
(b)	150	20000	19500	195	0.041	0.045
(c)	$150/\sqrt{2}$	10000	9750	195	0.047	0.046
(d)	$150\sqrt{2}$	40000	38600	193	0.046	0.047
(e)	$150/\sqrt{2}$	10060	9750	195	0.079	0.109
(f)	$150/\sqrt{2}$	10020	9250	185	0.147	0.081
	Mean	18347	17942	195	0.062	0.063

TABLE III

POPULATION SIZE ESTIMATION RESULTS WITHOUT PROJECTION CORRECTION

Image	T	N	$\widehat{N_1}$	Q_1	$\mathrm{ce}_{\mathrm{Cav}}\left(\widehat{N_1}\right)$	$\mathrm{CE}_{\mathrm{e}}\left(\widehat{N}\right)$
(a)	3000	20000	15200	152	0.147	0.278
(b)	3000	20000	21000	210	0.305	0.419
(c)	$3000/\sqrt{2}$	10000	6450	129	0.101	0.395
(d)	$3000\sqrt{2}$	40000	36400	182	0.441	0.811
(e)	$3000/\sqrt{2}$	10060	9050	181	0.405	0.343
(f)	$3000/\sqrt{2}$	10020	4150	83	0.087	0.785
	Mean	18347	15375	156	0.431	0.784

In this context we consider a particle as the planar projection of a human head, or a clearly distinguishable fragment of it. This is a reasonable choice since most of the bodies are occluded but almost all heads are visible. Furthermore the forbidden line rule was used for unbiased manual counting in the sample [11]; i.e., a human head was counted in a quadrat only if it had points in common with the quadrat but it did not hit the extended, red forbidden line of the quadrat.

Using a 60° tilt, $Q = 166$ faces were counted in the 37 non-empty quadrats of Fig. 7, yielding $\widehat{N} \approx 29500$ and a predicted coefficient of error of 9%.

V. APPLICATION TO REAL CROWDS

Throughout the paper we have only estimated the number of particles clearly distinguishable in an image, ignoring those that have not been pictured. Therefore in order to estimate the size of a real crowd, all the people in the crowd should be distinguishable in the image. This is technically feasible but can be expensive in the case that the crowd covers several streets, or if some people are occluded by trees or other objects. The gigapixel image considered in Fig. 7 only covered a part of the demonstration, which continued along adjacent streets. The number of occluded people was neglected although a few people might be hidden by the metro station and the statue. Moreover only about 98% of the area of Fig. 3(a) appears on the cylindrical projection images Fig. 3(b),(c). Therefore we would need additional images of the unobserved area and the adjacent streets, in order to estimate the total size of the

demonstration. In quantitative microscopy the analogous problem is solved using systematically sampled images (Figs. 2, 4 in [20]).

A comparison with official estimates given by police, media and organizers is ill-posed because unbiased estimates obtained by proper sampling cannot be compared with "guesstimates". In addition *CountEm* directly estimates population size avoiding inaccurate area estimations. In fact defining the area corresponding to a particle cloud is a non trivial problem [21]. A rough estimation of the area observed in Fig. 7 yields $13600\ m^2$, which together with the result $\widehat{N} \approx 29500$ leads to a density of about 2.2 people$/m^2$. According to newspaper *EL PAIS* [22], the densities in this area were between 3 and 4 people$/m^2$ which would lead to a crowd size between 40800 and 54400 people.

VI. Summary and Conclusion

Geometric sampling on still images using a square grid of square quadrats provides an unbiased population size estimator irrespective of population size and pattern. Only $50 - 100$ manual counts are usually necessary to achieve relative standard errors of $5\% - 10\%$. This can be done within a few minutes and it allows fast and reliable quantification by any users. The only practical limitation is the basic requirement that all the particles in the population should be unambiguously identifiable for counting. For large populations $\left(N > 10^4\right)$ as the one considered here, the requirement can be met using gigapixel pictures. However the big perspective artifacts significantly increase the sampling variance. This can be circumvented by generating the sampling grid on a map and projecting it onto the gigapixel picture. We show that in our case the relative standard error can be reduced from about 30% to near 7%. Calibration errors and projection inaccuracies show a low impact on the precision of the estimator. For $T = 20$ m and $t = 1.5$ m the relative standard error increases only slightly, from 7% to 9% using the approximations considered in section III-B. The robustness against variations in population size and spatial distribution has been shown in section III-C. The coefficient of error remains below 11% for all the cases considered in Fig. 6, using sample size $Q \approx 200$. Finally we estimated the size of the large real crowd visible in Fig 1(c), as $\widehat{N} \approx 29500$, with an estimated relative standard error of only 9%.

Appendix A
Point pattern simulation

The point pattern shown in Fig. 3(a) and left columns of Fig. 6 (a)-(d) may be computed directly with the package spatstat [19] of R using the following commands:

```
mod <- list(cif="hardcore",par=list(beta=100,hc=0.5),w=window)
PP <- rmh(model=mod,start=list(n.start=N),control=list(p=1,nrep=5e6,nverb=5e3))
```

The `window` is the region in which the `N` points are simulated. Different number of points, and windows were considered in each case (see caption of Fig. 6). The number of points was fixed to N in each case. The intensity parameter, `beta`, determines the particle abundance. Parameter `hc` is the hard core radius. More details can be found in [23].

The R-spatstat function rpoispp [19] was used to simulate the inhomogeneous distributions in Fig. 6(e) and (f), with the following commands:

```
PPe <- rpoispp(function(x,y){ke/sqrt((x-x0)^2 + (y-y0)^2)},win=window)
PPf <- rpoispp(function(x,y) {kf/sqrt((x-x0)^2},win=window)}
```

with $k_e = 52$, $k_f = 3$ and (x_0, y_0) the coordinates of the centre of the window. The x and y axes are parallel to the long and short sides of the rectangle respectively. The functions determine the density of particles as a funcion of the position (x, y).

REFERENCES

[1] H. Jacobs, "To count a crowd," *Columbia Journalism Review*, vol. 6, no. 1, p. 37, 1967.

[2] R. Watson and P. Yip, "How many were there when it mattered?" *Significance*, vol. 8, no. 3, pp. 104–107, 2011.

[3] T. P. Lynch, R. Alderman, and A. J. Hobday, "A high-resolution panorama camera system for monitoring colony-wide seabird nesting behaviour," *Methods in Ecology and Evolution*, vol. 6, no. 5, pp. 491–499, 2015.

[4] V. Lempitsky and A. Zisserman, "Learning to count objects in images," in *Advances in Neural Information Processing Systems*, 2010.

[5] M. Rodriguez, I. Laptev, J. Sivic, and J. Y. Audibert, "Density-aware person detection and tracking in crowds," in *2011 International Conference on Computer Vision*, Nov 2011, pp. 2423–2430.

[6] H. Idrees, I. Saleemi, C. Seibert, and M. Shah, "Multi-source multi-scale counting in extremely dense crowd images," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2013, pp. 2547–2554.

[7] C. Zhang, H. Li, X. Wang, and X. Yang, "Cross-scene crowd counting via deep convolutional neural networks," in *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2015, pp. 833–841.

[8] F. Botta, H. S. Moat, and T. Preis, "Quantifying crowd size with mobile phone and twitter data," *Royal Society Open Science*, vol. 2, no. 5, p. 150162, 2015.

[9] S. Zhang, R. Benenson, M. Omran, J. Hosang, and B. Schiele, "How far are we from solving pedestrian detection?" in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 1259–1267.

[10] S. Descamps, A. Béchet, X. Descombes, A. Arnaud, and J. Zerubia, "An automatic counter for aerial images of aggregations of large birds," *Bird study*, vol. 58, no. 3, pp. 302–308, 2011.

[11] M. Cruz, D. Gómez, and L. M. Cruz-Orive, "Efficient and unbiased estimation of population size," *PLOS ONE*, vol. 10, no. 11, pp. 1–14, 11 2015. [Online]. Available: http://dx.doi.org/10.1371%2Fjournal.pone.0141868

[12] C. V. Howard and M. G. Reed, *Unbiased Stereology. Three-dimensional Measurement in Microscopy*, 2nd ed. Oxford: Bios/ Taylor & Francis, 2005.

[13] A. J. Baddeley and E. B. V. Jensen, *Stereology for Statisticians*. Chapman & Hall/ CRC, London, 2005.

[14] L. Cruz-Orive, "Stereology: A historical survey," *Image Analysis & Stereology*, vol. 36, no. 3, pp. 153–177, 2017. [Online]. Available: https://ias-iss.org/ojs/IAS/article/view/1767

[15] A. Morán. Gigapan podemos puerta del sol. [Online]. Available: http://lab.elespanol.com/estaticos/gigapan_sol

[16] H. J. G. Gundersen, E. B. V. Jensen, K. Kiêu, and J. Nielsen, "The efficiency of systematic sampling in stereology reconsidered," *J. Microsc.*, vol. 193, no. 3, pp. 199–211, 1999. [Online]. Available: http://dx.doi.org/10.1046/j.1365-2818.1999.00457.x

[17] H. J. G. Gundersen, "Notes on the estimation of the numerical density of arbitrary profiles: the edge effect," *J. Microsc.*, vol. 111, no. 2, pp. 219–223, 1977.

[18] A. J. Baddeley, "Spatial sampling and censoring," in *Stochastic Geometry: Likelihood and Computation*, O. E. Bandorff-Nielsen, W. S. Kendall, and M. N. M. van Lieshout, Eds. Chapman & Hall/ CRC, London, 1999, pp. 37–78.

[19] A. Baddeley, E. Rubak, and R. Turner, *Spatial Point Patterns: Methodology and Applications with R*. CRC Press, 2015.

[20] L.-M. Cruz-Orive and E. R. Weibel, "Sampling designs for stereology," *Journal of Microscopy*, vol. 122, no. 3, pp. 235–257, 1981.

[21] B. D. Ripley and J.-P. Rasson, "Finding the edge of a Poisson forest," *J. Appl. Probability*, vol. 14, no. 3, pp. 483–491, 1977.

[22] E. PAIS. Recuento de manifestantes en la 'marcha del cambio'. [Online]. Available: https://elpais.com/elpais/2015/01/31/media/1422734596_211375.html

[23] A. Baddeley and R. Turner, "Practical maximum pseudolikelihood for spatial point patterns," *Australian & New Zealand Journal of Statistics*, vol. 42, no. 3, pp. 283–322, 2000.

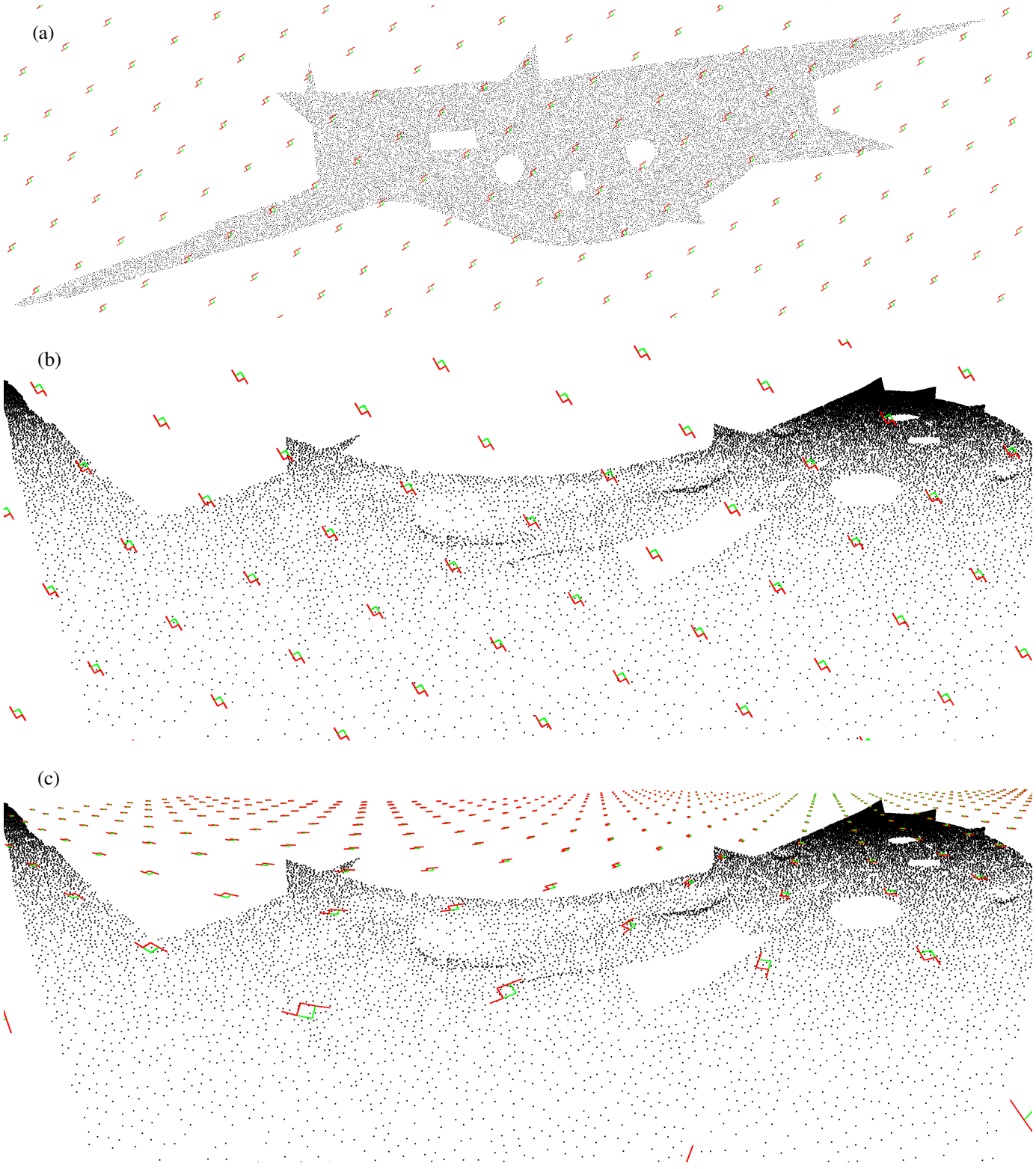

Fig. 3. (a): Simulation of crowd positions (1 dot = 1 person) on a map based on Fig. 2(a). A standard grid of systematic quadrats has been superimposed on it (Traditional *CountEm* method). (b): Unwrapped cylindrical projection of the crowd in (a) with a traditional *CountEm* grid. (c): Same crowd projection as in (b) with unwrapped cylindrical projection of the grid in (a) (novel *CountEm2* method proposed in this paper).

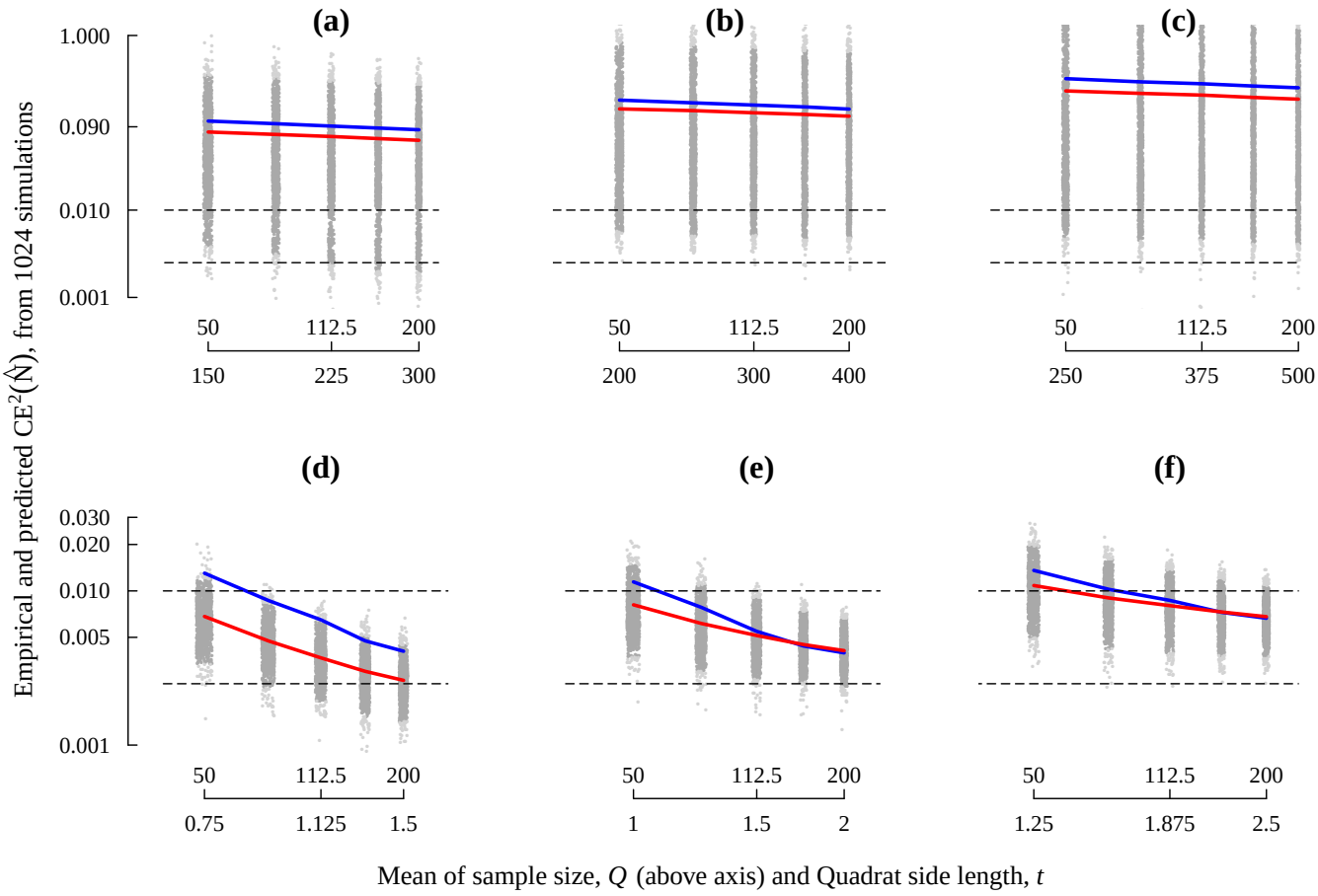

Fig. 4. (a, b, c) Monte Carlo results sampling with a standard grid of quadrats as in Fig. 3(b), for fundamental box side lengths $T = 3000, 4000, 5000$ pixels, respectively, and for different quadrat side lengths in each case. The equivalent mean sample sizes are also shown. The empirical square coefficient of error $\left(=\text{error variance divided by } N^2\right)$ is represented in blue, whereas the mean Cavalieri predictor is represented in red color. Grey dots represent all the replicated values of $\text{ce}^2_{\text{Cav}}\left(\widehat{N}\right)$. The dark gray dots lie between the 2.5% and 97.5% quantiles. The broken horizontal lines correspond to 5% and 10% coefficients of error, respectively. (d, e, f) Analogous data using a projected grid as in Fig. 3(c) — here $T = 15, 20, 25$ m, respectively. Empirical coefficients of error are in the $30\% - 40\%$ range for for *CountEm* and in the $5\% - 10\%$ for *CountEm2*. The Cavalieri estimator shows a reasonable performance in all the considered cases.

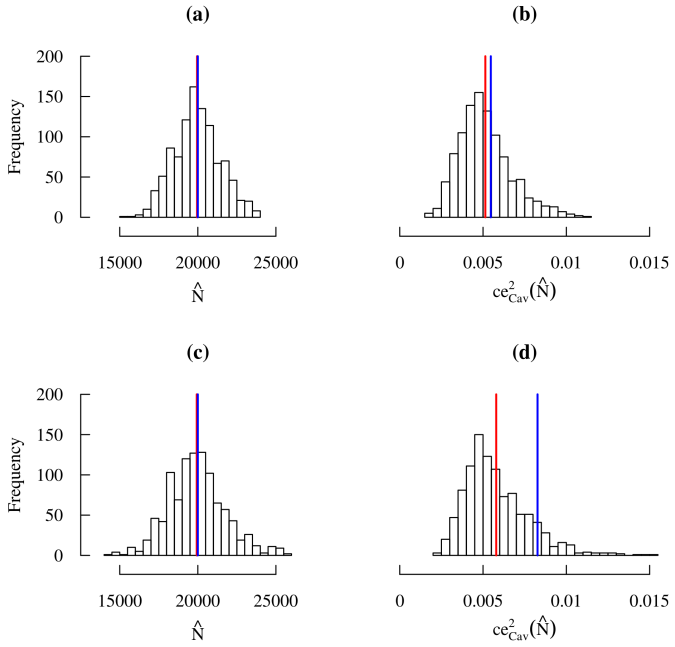

Fig. 5. Histograms of the Monte Carlo results for the 1024 replications for $T = 20$ m and $t = 1.5$ m using exact (a, b) and *simplified* (c,d) projection respectively. The mean of the 1024 $\widehat{N}$ Monte Carlo values is represented in (a) and (c) with a red line, whereas the blue line shows the "true" empirical value $N = 20000$. Analogously in (b) and (d) the red line stands for the mean Cavalieri predictor and the blue line for the empirical square coefficient of error.

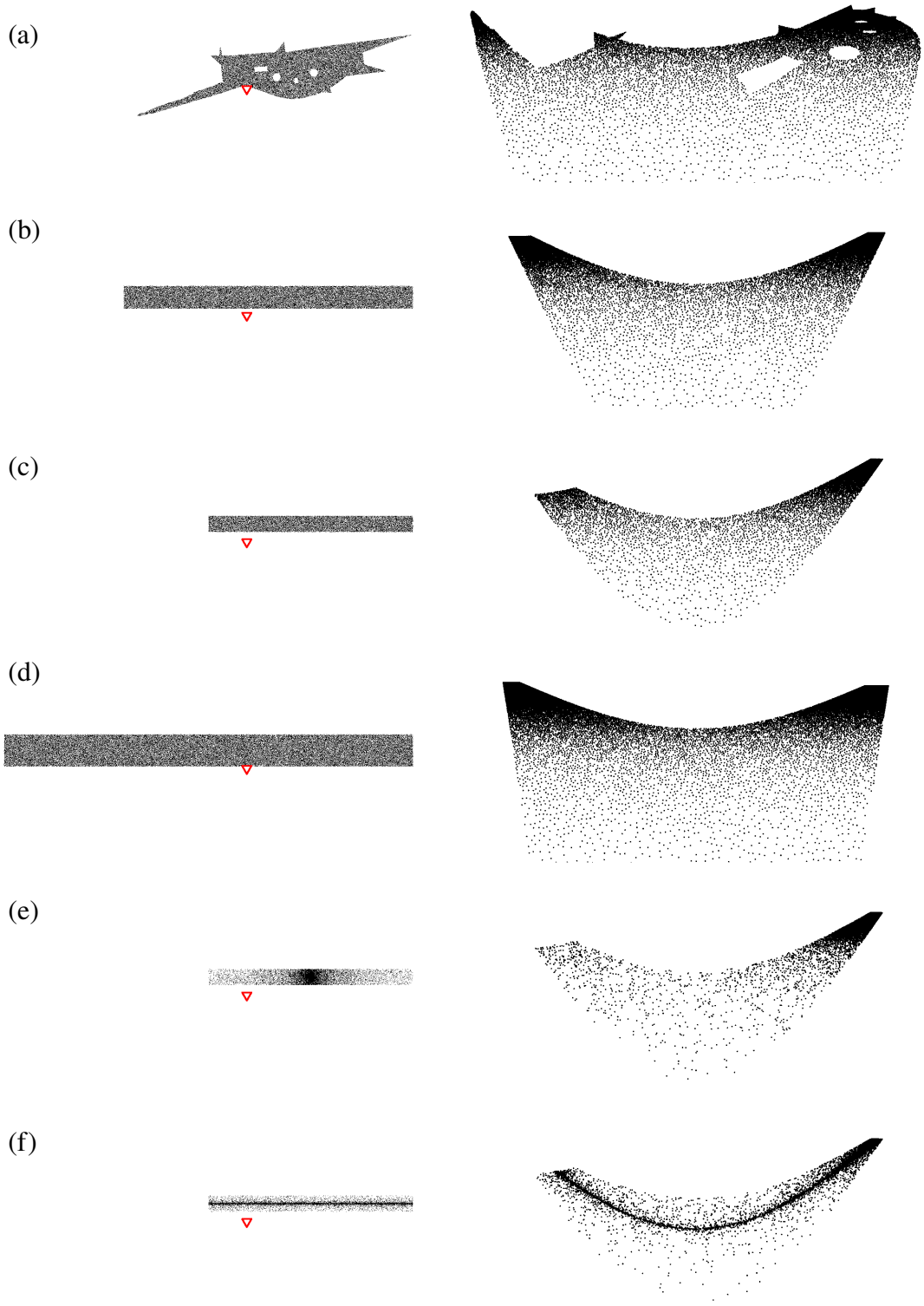

Fig. 6. Simulated point patterns as in Fig. 3, with different crowd sizes and spatial distributions. The left column represents the particles on a map based on Fig. 2(a), and the right column the unwrapped cylindrical projection of the point patterns. (a): $N = 2 \times 10^4$ points with hard core distribution as in Fig. 3. (b): Same as (a) but considering a rectangular area on the map. (c): Same as (b) but $N = 10^4$ and halving the area of the rectangle. (d): Same as (b) but $N = 4 \times 10^4$ and doubling the rectangular area. (e): Same as (c) but with inhomogeneous spatial distribution. The particle density decreases with the distance to the centre of the rectangle. (f): Same as (e) but the particle density decreases with the distance to the central x axis of the rectangle. The red triangle represents the position of the camera.

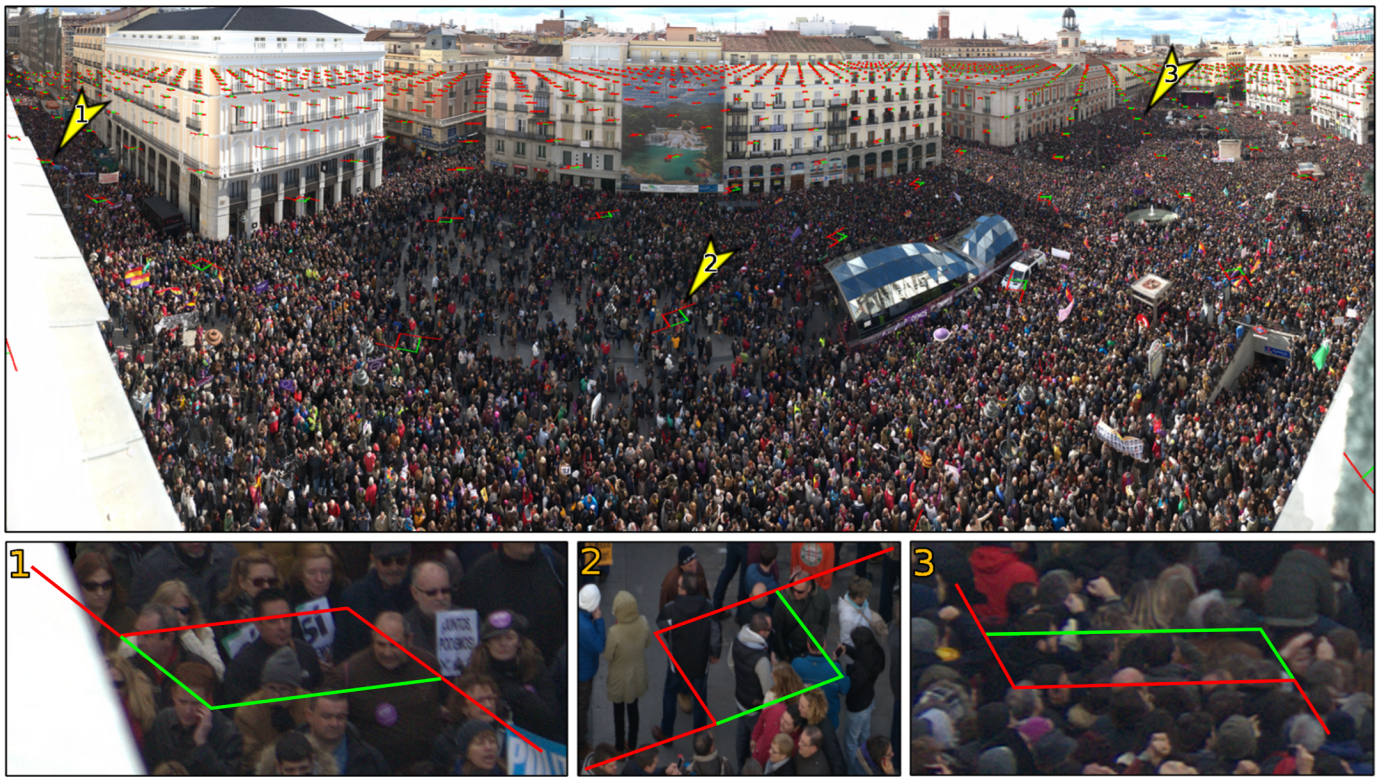

Fig. 7. Low resolution version of the $T = 20$ m and $t = 1.5$ m sampling grid projected onto the gigapixel image (top). The quadrats marked with yellow arrowheads are magnified in the bottom row.