Capturing layers in image collections with componential models: from the layered epitome to the componential counting grid

Alessandro Perina
Microsoft Research, Redmond, USA
alperina@microsoft.com

Nebojsa Jojic
Microsoft Research, Redmond, USA
jojic@microsoft.com

Abstract

Recently, the Counting Grid (CG) model [5] was developed to represent each input image as a point in a large grid of feature counts. This latent point is a corner of a window of grid points which are all uniformly combined to match the (normalized) feature counts in the image. Being a bag of word model with spatial layout in the latent space, the CG model has superior handling of field of view changes in comparison to other bag of word models, but with the price of being essentially a mixture, mapping each scene to a single window in the grid. In this paper we introduce a family of componential models, dubbed the Componential Counting Grid, whose members represent each input image by multiple latent locations*, rather than just one. In this way, we make a substantially more flexible admixture model which captures layers or parts of images and maps them to separate windows in a Counting Grid. We tested the models on scene and place classification where their componential nature helped to extract objects, to capture parallax effects, thus better fitting the data and outperforming Counting Grids and Latent Dirichlet Allocation, especially on sequences taken with wearable cameras.*

1. Introduction

The most basic Counting Grid (CG) model [5] represents each input image as a point $\mathbf{k}$ in a large grid of feature (SIFT, color, high level feature) counts. This latent point is a corner of a window of grid points which are all uniformly combined to form feature counts that match the (normalized) feature counts in the image. Thus, the CG model strikes an unusual compromise between modeling the spatial layout of features and simply representing image features as a bag of words where feature layout is completely sacrificed. The spatial layout is indeed forgone in the representation of any single image, as the model is simply concerned with modeling the feature histogram. However the spatial layout is present in the counting grid itself, which, by being trained on a large number of individual image histograms, recovers some spatial layout characteristics of the image collection to the extent needed to capture correlations among feature counts. For example, in a collection of images of a scene taken by a camera with a field of view that is insufficient to cover the entire scene, each image will capture different scene parts. Slight movement of the camera produces correlated changes in feature counts, as certain features on one side of the view disappear, and others appear on the other side. The resulting bags of features show correlations that directly fit the CG model. Ignoring the spatial layout in the image frees the model from having to align individual image locations, allowing for geometric defor-

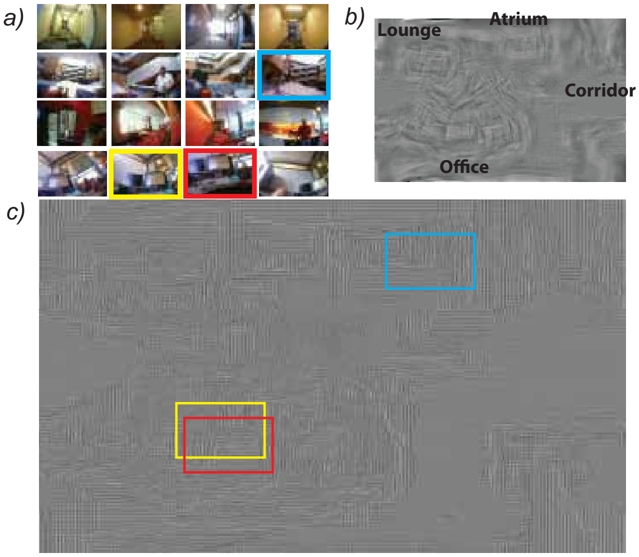

Figure 1. a) Images from 4 classes of the SenseCam dataset [6] (*Office*, *Atrium*, *Corridor*, *Lounge*) b-c) Visualization of the top words in each counting grid location. In c) in each location we show the texton that corresponds to the peak of the distribution (π_i) at the location, while in b), we overlap these textons by as much as the patches were overlapping during feature extraction process, and then average to create a clearer visual representation. We also show few windows and their mapping position on the Grid. Componential Counting Grids map each image in multiple locations, in this figure we only show a window in correspondence of the most likely location.

Table 1. Members of the Componential Counting Grid Family. $\mathbf{W}$ and $\mathbf{S}$ are respectively the Window size and the Tessellation size.

Model	Abbr.	$\mathbf{W}$	$\mathbf{S}$
Latent Dirichlet Allocation [7]	LDA	1×1	1×1
Componential Counting Grid	CCG	$> 1 \times 1$	1×1
Tessellated Compon. Counting Grid	tCCG	$> 1 \times 1$	$> 1 \times 1$
Layered Epitome	lEP	$N_x \times N_y$	$N_x \times N_y$

mations, while the grid itself reconstructs some of the 2D spatial layout needed for modeling feature count correlations.

As shown in [5] and as we demonstrate in Fig. 1, arranging counts on a topology that allows feature sharing through windowing can have representational advantages beyond this surprising possibility of panoramic scene reconstruction from bags of features. Counting Grids have been recently used in the context of scene classification [4] and video analysis in [19, 6].

In this paper we introduce the Componential Counting Grids (CCG), a family of models which extend the basic Counting Grid model so that each input image is represented by multiple latent locations in CG, rather than just one. Through admixing locations, CCG models become multi *-part* or *-object* models but, like their CG predecessor, they recreate only as much of spatial layout in the counting grid as necessary for capturing count correlations.

This family creates connections between two popular generative modeling strategies in computer vision, previously seen as very different: By varying the image tessellation and window size, we will get a variety of models among which we find latent Dirichlet allocation [7, 1] as well as flexible sprites [18]/Layered Epitomes at two ends, or rather corners, of the spectrum illustrated in Fig.2. In each of these corners, substantial research effort has been invested to refine and apply these basic approaches, but it turns out that the CCG models at neither end of the spectrum tend to perform best in our experiments. A summary of these models can be found in Tab.1.

Componential Counting Grids and Topic models [7] The original counting grid model shares its focus on modeling image feature counts (rather than feature layouts) with another category of generative models, the "topic models", such as latent Dirichlet allocation (LDA) [7, 1]. However, neither of these is a generalization of another. The CG model is essentially a mixture model, assuming only one source for all features in the bag, while the LDA model is an admixture model that allows mixing of multiple topics to explain a single bag. By using large windows to collate many grid distributions from a large grid, CG model can be a very large mixture of sources without overtraining, as these sources are highly correlated: Small shifts in the grid change the window distribution only slightly. LDA model does not have this benefit, and thus has to deal with a smaller number of topics to avoid overtraining. Topic mixing cannot quite appropriately represent feature correlations due to translational camera motion.

The basic Componential Counting Grid model, however, is a generalization of LDA, as it does allow multiple sources for each bag, in a mathematically identical way as LDA. But, the equivalent of LDA topics are windows in a counting grid, which allows the model to have a very large number of topics that are highly related, as shift in the grid only slightly refines any topic.

The most similar generative model to CCG comes from the statistic community. Dunson et al. [17] worked on sources positioned in a plane at real-valued locations, with the idea that sources within a radius would be combined to produce topics in an LDA-like model. They used an expensive sampling algorithm that aimed at moving the sources in the plane and determining the circular window size. The grid placement of sources of CCG yields much more efficient algorithms and denser packing. In addition, as illustrated below, CCG model can be run with various tessellations efficiently making it especially useful in vision applications.

Generative models for vision: Tessellated Componential Counting Grids and Layered Epitomes. In computer vision, instead of forming a single bag of words out of one image, separate bags are typically extracted from a uniform $\mathbf{S} = S_x \times S_y$ rectangular tessellation of the image [6, 8, 10]. The Tessellated extension of CCG (tCCG) is as straightforward as was the corresponding extension of CG [4]. All sections are mapped to the same grid, but, the corresponding window is tessellated in the same way as the image, and the feature histograms from corresponding rectangular segments are supposed to match. Even with as coarse tessellations as 2×2, training CG on image patches can result in panoramic reconstruction similar to that of the epitome model which entirely preserves the spatial layout.[1] When the Tessellation is equal to the image size $\mathbf{S} = N_x \times N_y$, every bag is composed by a single feature, and we obtain the Layered Epitome. Like regular Epitomes [2] and flexible sprites [18] preserves the spatial layout of features. However, differently from [2], tCCGs break each image into layers and maps them separately in the epitome space, and differently from [18], it does not assume a pre-defined number and an ordering between layers.

In Fig. 2, though, we show a variety of Componential models one can obtain by varying the tessellation and the window size for the mapping. The window size need not, and usually in our experiments does not match the size of the input image, except for the .

[1] Of course, when the data does not consist of patches from a single image, but from patches or images with more geometric deformation, CG or recently introduced SLCG [6] model typically have a significant advantage.

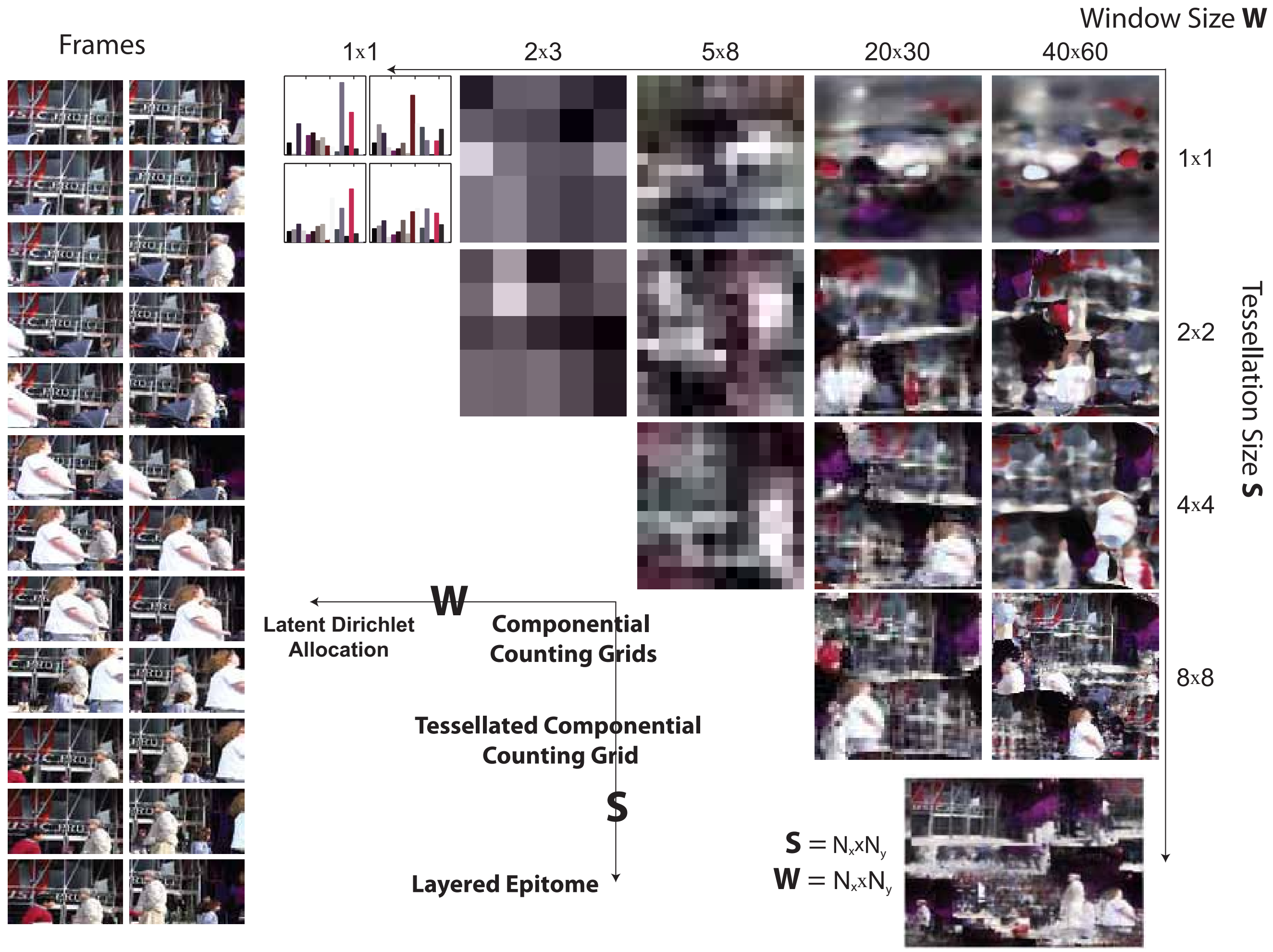

Figure 2. "CCG spectrum": Relationship of Componential Counting Grids Family with Layered Epitomes and Topic models. Although the Layered Epitome has the same capacity its height/width ratio is different as we are forced to set $\mathbf{W} = N_x \times N_y$.

Images used in training contain multiple objects and a background captured from a moving field of view, and a subset of frames is shown in the image.[2] Due to visualization advantages for this illustration, all models were trained using discretized colors rather than SIFT features, and they all have roughly the same *capacity* – the number of *independent* topics that can be created in the allotted space without overlapping the windows. This means that counting grids created with smaller windows have to be proportionally smaller, but for better visualization we enlarged all grids to the same size. Window overlaps create smooth interpolations among topics that compensate for camera motion. When 1×1 windows are used, there is no sharing of grid distributions among topics, and the model reduces to LDA shown in the corner with its histograms for its topics. As there is no sharing, the spatial arrangement of four topics onto the 2×2 grid has no meaning or value. Layered epitomes or flexible sprites are another extreme where both the window size and the tessellation match the resolution of input images[3], but the CCG models with as coarse a tessellation as 8×8 already look indistinguishable from epitome/flexible sprite results.

The video sequence features prominently a man and a women dressed in white clothing (see the Frames in Fig.2). While LDA color model will obviously confuse the white elements of the background with these foreground objects, the model with full tessellation has to learn multiple versions of each person to capture the scale changes due to their motion at an angle with the motion of the camera. The intermediate tessellations and window size provide more interesting tradeoffs. For example, we see a generalized representation of each object, where some of the original spatial layout of features is recovered, but the allowed rearrange-

[2] Frame time stamps are not used to create the models: No tracking!

[3] Note, however, that we do not learn object masks here, as was done in flexible sprites

ment of the features in the tessellation segments compensates for scale. When the model is forced to simplify further, through appropriate choice of window and tessellation size, the two persons dressed in white are generalized into a single object (though it may occur twice in one image).

While this illustration reinforces the naturally good fit of CCG models to images of scenes with multiple moving objects taken by a camera with a moving field of view, the applicability of the CCG models hardly stops there. Fig.1 illustrates the value of computing a grid of features in a very different context, where one large grid is computed from all images from 4 of the 32 class wearable camera dataset [6]. Each image was represented by a single bag of features (1×1 tessellation) and the counting grid is computed using 38×50 windows. A total of 200 feature centers were used, and in each spot in the grid, only the peak of the histogram is shown. The model tends to break up each bag into more topics, and instead of reflecting a panoramic reconstruction, the grid now models smaller scene parts, such as vertical and horizontal edges found in windows and building walls that the subject sees in his office and elsewhere. The choice of edges placed close together shows that the model makes sure that a window into the grid captures an appropriate feature mix found in some of the images in the training set. In multiple places in the grid we see that when the window is moved the orientation of the edges changes slightly and in concert. Thus, in this case the CG real-estate and window overlapping strategy was often used to model rotation, rather translation.

Next we mathematically describe the basic CG model, which bears a lot of similarity with representations in Fig. 2, but as opposed to these, it does not model multiple scene parts as mapped to different parts of the CG, but would rather have to try to learn all foreground-background combinations. Then, we formally define the CCG model and derive the learning algorithm for it. Finally, we demonstrate the CCG performance on various datasets.

2. From Counting Grids to Componential Models

The basic 2-D Counting Grid $\pi_{\mathbf{i},z}$ is a set of normalized counts of words/features indexed by z on the 2-dimensional discrete grid indexed by $\mathbf{i} = (i_x, i_y)$ where each $i_d \in [1 \dots E_d]$ and $\mathbf{E} = (E_x, E_y)$ describes the extent of the counting grid. Since π is a grid of distributions, $\sum_z \pi_{\mathbf{i},z} = 1$ everywhere on the grid. Each bag of words/features, is represented by a list of word $\{\mathbf{w}^t\}_{t=1}^T$; we will assume that all the samples have N words and each word w_n^t takes a value between 1 and Z.

Counting Grids assume that each bags follow a feature distribution found somewhere in the counting grid; In particular, using windows of dimensions $\mathbf{W} = (W_x, W_y)$, a bag can be generated by first averaging all counts in the window $W_{\mathbf{i}}$ starting at 2-dimensional grid location $\mathbf{i}$ and extending in each direction d by W_d grid positions to form the histogram $h_{\mathbf{i},z} = \frac{1}{\prod_d W_d} \sum_{\mathbf{j} \in W_{\mathbf{i}}} \pi_{\mathbf{j},z}$, and then generating a set of features in the bag. In other words, the position of the window $\mathbf{i}$ in the grid is a latent variable given which we can write the probability of the bag as

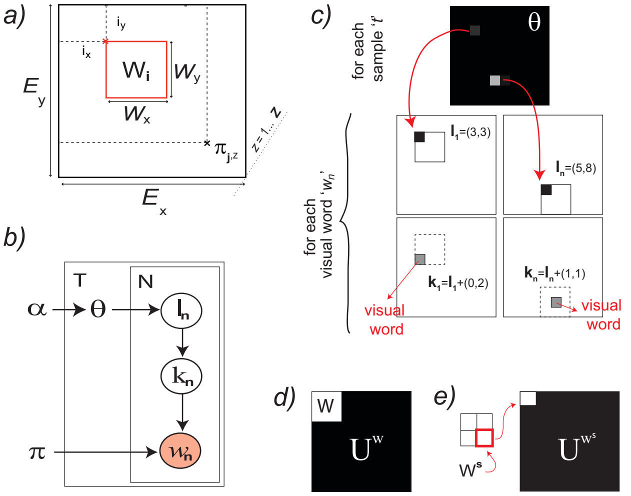

Figure 3. a) Counting Grid geometry. b) Componential Counting Grid/Layered Epitome generative model. c) CCGs generative process. d) Illustration of U^W; e) Illustration of U^{W_s} in the case of a $\mathbf{S} = 2 \times 2$.

$$p(\{\mathbf{w}\}|\mathbf{i}) = \prod_n h_{\mathbf{i},z}(w_n) = \prod_n \Big(\frac{1}{\prod_d W_d} \sum_{\mathbf{j} \in W_{\mathbf{i}}} \pi_{\mathbf{j},z}(w_n)\Big),$$

An example of CG geometry is shown in Fig.3a.

Relaxing the terminology, $\mathbf{E}$ and $\mathbf{W}$ are referred to as, respectively, the counting grid and the window size. The ratio of the two volumes, κ, is called the capacity of the model in terms of an *equivalent number of topics*, as this is how many non-overlapping windows can be fit onto the grid. Finally, with $W_{\mathbf{i}}$ we indicate the particular window placed at location $\mathbf{i}$.

Componential Counting Grids As seen in the previous section, Counting Grids generate words from a feature distribution in a window W, placed at location $\mathbf{i}$ in the grid. Locations close in the grid generate similar features. As we move the window on the grid, some new features appear while others are dropped. Learning the model that can generate this way produces panoramic reconstructions in the CG (as seen in Fig.1) or, at a higher level, captures (or infers new) spatial or topological relationships among features (i.e., features of the sea are *close to* sand, buildings are often *over* a street). On the other hand in standard componential models, [7], each feature can be generated by a different "process" or "topic." These models capture feature co-occurrences (e.g., sands often comes with sea), and by breaking the bag into topics can potentially segment the image into parts.

Componential Counting Grids get the best of both worlds: using the counting grid embedding through window overlapping, they can recover spatial layout, but like componential models they can also explain the bags as generated from multiple positions in the grid (called components), explaining away the foreground and clutter, or discovering parts that can be combinatorially combined in the image collection (e.g., grass, horse, ball, athlete, to explain different sports that may be created mixing these topics).

In the CCG generative model each bag is generated by mixing several windows in the grid following the location distribution θ. More precisely, each word w_n can be generated from a different window, placed at location l_n, but the choice of the window follows the same prior distributions θ_l for all words. Within the window at location l_n the word comes from a particular grid location k_n, and from that grid distribution the word is assumed to have been generated.
The Bayesian network is illustrated in Fig.3b) and it defines the following joint probability distribution

$$P = \prod_t \Big(p(\theta|\alpha) \prod_n \sum_{l_n,k_n} p(w_n|k_n,\pi) \cdot p(k_n|l_n) \cdot p(l_n|\theta) \Big)$$

where $p(w_n = z|k_n, \pi) = \pi_{k_n}(z)$ is a multinomial over the word indices, $p(k_n|l_n) = U^W_{k_n-l_n}$ is a distribution over the Counting Grid, equal to $(\frac{1}{W_x \cdot W_y})$ in the upper left window of size $\mathbf{W}$ and 0 elsewhere (see Fig.3d), $p(l_n|\theta) = \theta_l$ is a prior distribution over the windows location, and $p(\theta|\alpha) = Dir(\theta; \alpha)$ is a dirichlet distribution of parameters α.
The generative process (Fig.3c), is the following:

1. Sample a multinomial over the locations $\theta \sim \alpha$
2. For each of the N words w_n
   - **a)** Choose a location $l_n \sim \theta$ for a window $\mathbf{W}$
   - **b)** Choose a location within $\mathbf{W}_{l_n}$; $k_n \sim U^W_{k_n-l_n}$
   - **c)** Choose a word w_n from π_{k_n}

Since the posterior distribution $p(\mathbf{k}, \mathbf{l}, \theta|\mathbf{w}, \pi, \alpha)$ is intractable for exact inference, we learned the model using variational inference [16]. By introducing the posterior distributions q, and approximating the true posterior as $q^t(\mathbf{k}, \mathbf{l}, \theta) = q^t(\theta) \cdot \prod_n (q^t(k_n) \cdot q^t(l_n))$ [4] we can write the negative free energy $\mathcal{F}$, and use the iterative variational EM algorithm to optimize it.

$$\mathcal{F} = \sum_t \Big(\sum_n \sum_{l_n,k_n} q^t(k_n) \cdot q^t(l_n) \cdot \log \pi_{k_n}(w_n) \cdot U^W_{k_n-l_n} \cdot \theta_l \cdot p(\theta|\alpha) \Big) - \mathbb{H}(q) \quad (1)$$

where $\mathbb{H}(q)$ is the entropy of the posterior. Minimization of Eq. 1 reduces in the following update rules:

$$q^t(k_n) \propto \pi_{k_n}(w_n) \cdot \exp\big(\sum_{l_n} q^t(l_n) \cdot \log U^W_{k_n-l_n} \big) \quad (2)$$

$$q^t(l_n) \propto \theta^t_{l_n} \cdot \exp\big(\sum_{k_n} q^t(k_n) \cdot \log U^W_{k_n-l_n} \big) \quad (3)$$

$$\theta^t_l \propto \alpha_l - 1 + \sum_n q^t(l_n) \quad (4)$$

$$\pi_k(z) \propto \sum_t \sum_n q^t(k_n = k)^{[w_n = z]} \quad (5)$$

where $[w_n = z]$ is an indicator function, equal to 1 when w_n is equal to z. The minimization procedure described by Eqs.2-5 must be iterated until convergence and can be carried out efficiently in $\mathcal{O}(N \log N)$ time using FFTs.

Tessellated Componential Counting Grids The procedure described in the previous section does not require information about the spatial layout of features in the bag and can be in principle applied to any kind of data. In computer vision, it is useful to enrich the model and its E and M rules to deal with image representations that consist not of one, but several $\mathbf{S} = s_x \times s_y$ bags of words, each corresponding to a section of the image [8, 10]. When inferring the mapping of each "section" bag, the window $W_{\mathbf{k}}$ is tessellated into section $W^{\mathbf{S}}_{\mathbf{k}}$ in the same way images are tessellated and the histogram comparisons are done accordingly. Moreover $U^W_{k_n-l_n}$ becomes $U^{W_s}_{k_n-l_n}$, where $\mathbf{W}_s$ is a window of the same size and it is shown in Fig.3e. In a similar way non-uniform and most descriptive image layout patterns can be used [20].

Layered Epitomes In the limit, when $\mathbf{S} = N_x \times N_y$ each bag contains a single feature and the model becomes the Layered (or componential) Epitome. In this case, n indexes a pixel in the image coordinate $\mathbf{i}$ (e.g., $w_n = z_{\mathbf{i}}$) and $U^{\mathbf{i}}$ highlights now the single pixel $\mathbf{i}$. The E-Step thus becomes:

$$q^t(k_{\mathbf{i}}) \propto \pi_{k_{\mathbf{i}}}(z_{\mathbf{i}}) \cdot \exp\big(\sum_{l_{\mathbf{i}}} q^t(l_{\mathbf{i}}) \cdot [k_{\mathbf{i}} - l_{\mathbf{i}} = \mathbf{i}] \big) \quad (6)$$

$$q^t(l_{\mathbf{i}}) \propto \theta_{l_{\mathbf{i}}} \cdot \exp\big(\sum_{k_{\mathbf{i}}} q^t(k_{\mathbf{i}}) \cdot [k_{\mathbf{i}} - l_{\mathbf{i}} = \mathbf{i}] \big) \quad (7)$$

where $[\cdot]$ is the indicator function. In both the layered epitome and tessellated case θ and π are updated as in Eq.4 and Eq.5.

[4] $q(k_n)$ and $q(l_n)$ multinomials over the locations, and $q(\theta)$ a Dirac function centered at the optimal value $\hat{\theta}$

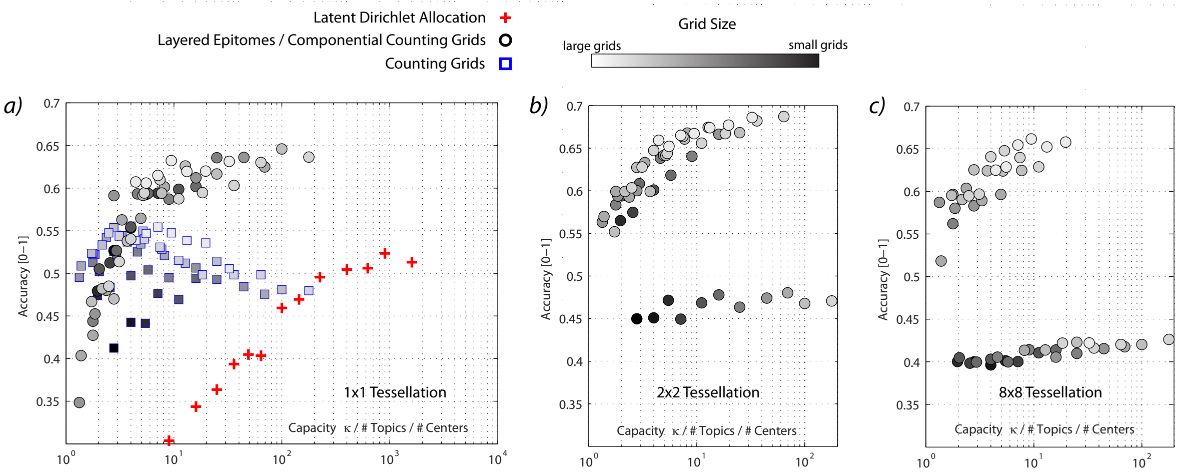

Figure 4. Results on SenseCam (Mean results over 5 repetitions). As the same κ can be obtained with different choice of **E** and **W**, multiple results may be reported for the some values of κ. For (Componential) Counting Grids, we colored the markers based on the size of **E**. a) Single Bag model and comparison with [4] and [7]. b) Moderate Tessellation results c) Fine Tessellation results.

3. Experiments

In all the experiments as visual words we used SIFT features, extracted from 16×16 patches spaced 8 pixels apart, clustered in Z=200 visual words. In each task, unless specified, we employed the dataset author's training/testing/validation partition and protocol; if this information was not available, we used 10% of the training data as validation set.

We considered squared grids of various complexities $\mathbf{E} = [\mathbf{2}, \mathbf{3}, \ldots, \mathbf{10}, \mathbf{15}, \mathbf{20}, \ldots, \mathbf{40}]$ and window size $\mathbf{W} = [\mathbf{2}, \mathbf{4}, \mathbf{6}, \ldots]$ but limiting the tests only to the combinations with capacity $\kappa = \frac{E_x \cdot E_y}{W_x \cdot W_y}$ between 1.5 and $T/2$, where T is the number of training samples. We tried single bag models (1×1 tessellation), tessellated models 2×2, 4×4 and the layered epitome ($N_x \times N_y$).

Place Classification on SenseCam: Recently in [6] a 32-classes dataset have been proposed. This dataset is a subset of the whole visual input of a subject who wore a wearable camera for few weeks. Images in the dataset exhibit dramatic viewing angle, scale, illumination variations and a lot of foreground objects, and clutter.

We compared CCGs with LDA [7] and CGs [4], learning a model per class and assigning test samples to the class that gives the lowest free energy. The capacity κ is roughly equivalent to the number of LDA topics as it represents the number of independent windows that can be fit in the grid; we compared the results using this parallelism [4, 6].

Results are shown in Fig.4: the Componential Counting Grid model outperforms LDA and CGs across the choices of model complexity considered. Like [7], it breaks each image into parts and, like regular CGs, it maps these onto a bigger real estate, trying to recover their panoramic nature, by laying out the features into a 2D window and stitching overlapping windows. This fits both the panoramic and componential qualities of the data acquired by a wearable camera.

Table 2. Comparison with state of the art on SenseCam dataset. We reported accuracies from [6], where comparisons with other methods can be found.

CCG	[11]	[10]	[6]	[8]
64.03%	43.65%	57.47%	60.12%	56.45%

Moderate tessellations (up to 4×4) significantly helped, except for very small grid/window sizes, where the model reduces itself to a very low resolution layered epitome, or for high κs, where it probably overtrains. Layered epitomes did not perform well ($\leq 40\%$) as the training data is limited and images are too diverse for panoramic stitching.

The overall accuracy after crossevaluation is $64\% \pm 1.7$ strongly outperforming recent advances in scene recognition [11, 10, 6] and setting a new state-of-the-art by a large margin (See Tab.4).

Scene Recognition: We tested our models on the video sequences introduced in [9]. In addition to the comparison with the original method [9], we also compared with Epitomes [3], as epitomic location recognition [3] was, among recognition applications of epitome, one of the most successful. The trick was to use low resolution epitome with each low res image location represented by a histogram of features. Results are presented in Fig.5; the improvement is significant and once again, CCGs set a new state-of-the-art.

We finally considered the UIUC Sports dataset [12], this dataset is particularly challenging as composing elements

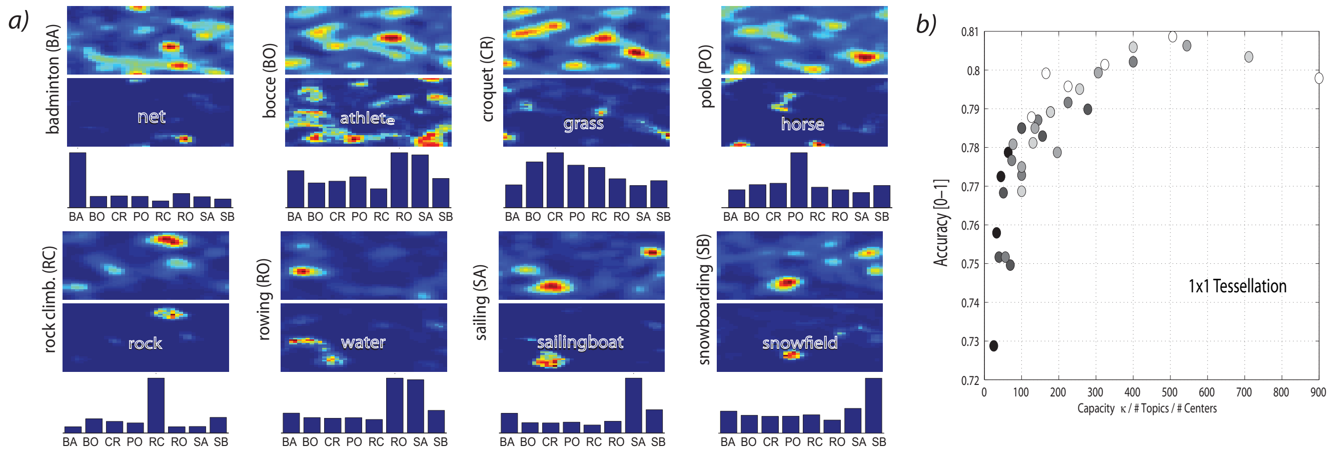

Figure 6. a) First/Fourth row: $p(\mathbf{l}|\theta, c)$, Second/Fifth row: Embedding of few words in the Grid (π_w^W). Third/Sixth row: Reciprocal of KL-"similarity" between the $p(\mathbf{i}|\theta, c)$ and π_w^W for each class. b) Results on UIUC Sports dataset across the complexities.

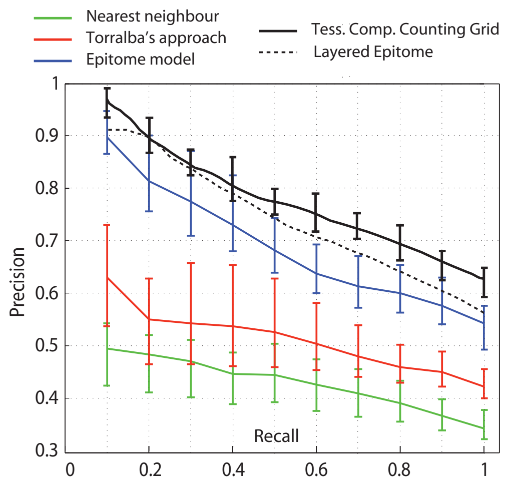

Figure 5. Results on Torralba sequences [9]. Our approach strongly outperforms Nearest Neighbor, and [3, 9]. We also reported the result of the layered epitome.

and objects must be identified in order to correctly classify the sport event [15].

For this task, we learned a single model pooling the images from all the classes together. We considered models of complexity $\mathbf{E} = [\mathbf{40}, \mathbf{50}, \ldots, \mathbf{90}]$ and $\mathbf{W} = [\mathbf{2}, \mathbf{4}, \mathbf{6}, \mathbf{8}]$ and we used training set's θ^t as feature to learn a discriminative classifier (We used SVM with histogram intersection kernel). The rationale here is that different classes share some elements, like "water" for sailing and rowing classes, but they also will have peculiar elements that distinguish them. This is shown in Fig.6a where we depicted $p(\mathbf{i}|\theta, c) = \sum_{t_c} \theta_{\mathbf{i}}^{t_c}$, where the sum is carried out separately on the samples of each class. After learning a model, we embedded the textual annotations available for this dataset, simply iterating the M-step using textual words as observations. In Fig. 6a we show where some selected words are embedded in the grid.

Table 3. Comparison with other componential models after crossevaluation. We did not use the annotations in the classification task.

CCG	CG	LDA[5]	[14]	[15]	[12]
80.02%	43%	36% - 68%	78%	76.3%	73%

Numerical accuracies on the test set are shown in Tab.3, while in Fig.6b we reported the accuracy across κ. As expected, CGs [4] fail as they stick to classify the scene in which the event takes place, but so does LDA [7]. CCGs, similar in spirit to [15] (but somewhat simpler), look and extract object/texture/feature combinations to classify images and reach compelling accuracies (see Fig.6b).

The variation in spatial layout of the objects here was sufficient to render tessellations beyond 1×1 unnecessary: They do not improve classification results (but increase in the window size is needed).

4. Discussion

The componential models introduced here can be seen as a generalization of both LDA and template-based models such as flexible sprites [18] or epitomes [3, 2]. As opposed to the basic CG model, it allows for source (object, part) admixing in a single bag of words. In addition, by partially decoupling the feature layout modeling in the image from the layout modeling in the latent space (the grid of feature distributions as in the CG model), it empowers the modeler to strike balance between layout following and transformation invariance in substantially different and more diverse ways than these previous models, simply by varying the tessellation and the mapping window size (which is typically not linked to the original image size).

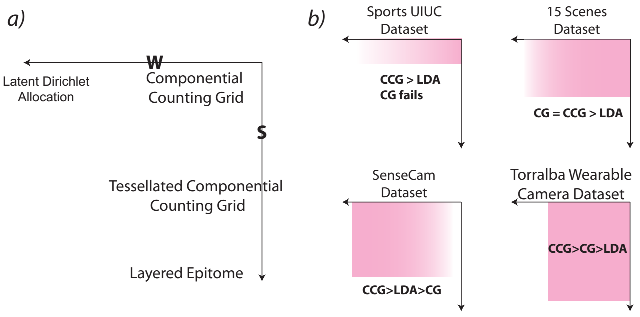

Figure 7. a) CCG Spectrum b) Performance across the spectrum on the database used in the experiments.

Keeping the capacity κ fixed, the increase in window size incurs the proportional increase in the computational cost, but provides for smoother reconstruction in the spatial layout. As experiments show, once the $\mathbf{W}$ is "sufficiently big", recognition accuracies raise with κ. The tessellation $\mathbf{S}$ guides the rough positioning of the features from different image quadrants and moderate tessellations never hurt. In our experiments we invariably find that the basic LDA and epitome-like models, which are at opposite corners of the model organization by tessellation and window size, underperform the CCG models from somewhere in the middle of the triangle illustrated on the toy data in Fig. 2.

It is also interesting to analyze the performance of the Componential Counting Grid family, Counting Grids [4] and LDA [7] for various datasets. In Fig.7, for each dataset considered in this paper, we colored the area where we reached "reasonably good" results. To correctly classify UIUC Sports images, objects/parts/athletes must be extracted and recognized. Componential models (CCG, LDA) break the image and perform well, while CGs fails as they classify the scene in which the event take place. Tessellations finer than $\mathbf{S} = 2 \times 2$, hurt the result as they made CCGs stick to the scene. SenseCam images and Torralba sequences are collected with a wearable camera and in principle the spatial layout can be at least piecewise reconstructed. Here all methods perform well and the tessellations significantly helped. Torralba sequences was the only dataset where layered epitomes were found to perform well. The lack of training data made small windows (and grids) preferable on SenseCam. Finally we also analyzed the 15-Scenes dataset [8][6] where Counting Grids and CCG outperformed LDA. Tessellation helped up to $\mathbf{S} = 5 \times 5$.

A number of refinements previously added to generative models can be added to CCG, e.g., the mask model akin to the ones used by flexible sprites and layered epitomes, modeling the spatial layout changes in tessellation segments as in the spring lattice CG model [6], exotic priors and added hierarchies as in LDA-based models, or as in any generative model, addition of other hidden variables that relate to other modalities or higher-level variables.

[6] We did not report the results in the paper for lack of space

References

[1] D.Blei, A.Ng, M.Jordan Latent Dirichlet Allocation J.Machine Learning Research, 2003

[2] N.Jojic, B.J. Frey, A.Kannan Epitomic analysis of appearance and shape ICCV 2003

[3] K. Ni, A. Kannan, A.Criminisi, J. M. Winn Epitomic location recognition CVPR 2008

[4] A.Perina, N.Jojic Image Analysis by Counting on a grid CVPR 2011

[5] N.Jojic, A.Perina Multidimensional Counting Grids UAI 2011

[6] A.Perina, N.Jojic Spring Lattice Counting Grids: Scene recognition using deformable positional constraints ECCV 2012

[7] Fei-Fei Li, P.Perona A Bayesian Hierarchical Model for Learning Natural Scene Categories CVPR 2005

[8] S. Lazebnik, C.Schmid, J.Ponce Beyond Bags of Features: Spatial Pyramid Matching for Recognizing Natural Scene Categories CVPR (2) 2006: 2169-2178

[9] A.Torralba, K.P.Murphy, W.T.Freeman and M.A.Rubin Context-based vision system for place and object recognition ICCV 2003: 273-280

[10] S.N. Parizi, J. Oberlin, P.F. Felzenszwalb Reconfigurable models for scene recognition CVPR 2012

[11] M.Pandey, S.Lazebnik Scene recognition and weakly supervised object localization with deformable part-based models ICCV 2011

[12] L.J.Li, L.Fei-Fei What, where and who? Classifying event by scene and object recognition. ICCV 2007

[13] C.Wang, D.Blei, L.Fei-Fei Simultaneous Image Classification and Annotation CVPR 2009

[14] Z.Niu, G.Hua, X.Gao, Q.Tian Context Aware Topic Model for Scene Recognition CVPR 2012

[15] L.J.Li, H.Su, E.Xing, L.Fei-Fei Object Bank: A High-Level Image Representation for Scene Classification and Semantic Feature Sparsification NIPS 2010

[16] R.Neal and G.E.Hinton A View Of The Em Algorithm That Justifies Incremental, Sparse, And Other Variants Learning in Graphical Models, Kluwer Academic Publishers, 1998

[17] D.Dunson, J-H.Park Kernel Stick-Breaking Processes Biometrika 2008

[18] N.Jojic, B.J. Frey Learning Flexible Sprites in Video Layers CVPR 2011

[19] M.R.Amer, S.Todorovic Sum-product networks for modeling activities with stochastic structure CVPR 2012

[20] Y.Jiang, J.Yuan, G.Yu Randomized Spatial Partition for Scene Recognition ECCV 2012