Summarization and classification of wearable camera streams by learning the distributions over deep features of out-of-sample image sequences

Alessandro Perina
WDG Core Data - Microsoft Corp.
Redmond, WA alperina@microsoft.com

Sadegh Mohammadi
PAVIS Dept. - Italian Institute of Technology
Genova, Italy sadegh.pub@gmail.com

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

Vittorio Murino
PAVIS Dept. - Italian Institute of Technology
Genova, Italy vittorio.murino@iit.it

1. Summary of the material

In the zip file, the reader will find

- High resolution image describing the embedding of SenseCam AIHS data in a 64×64 counting grid (Fig. 2 in main text). In each location, an image with the highest likelihood is shown. (CGembedding.jpg)
- A video of the random walk over the counting grid (TheAmericanDream.avi)
- The high resolution image of the tSNE embedding of the same SenseCam data using the same features (tSNE.pdf)
- The accompanying text including comparison with tSNE, as well as the list of all classes in the SenseCam dataset (this text)
- Classification in case of abundance of training data.

2. CG embedding illustrated

Although Counting Grids (CGs) can be trained in higher dimensions than two, and have been shown to benefit from extra dimensionality in several applications, we focused on 2D embeddings as we expect that visualization and browsing will play an important role in adoption of wearables. In Fig. 2 of the main text we illustrated the 2D CG mapping of 43516 images from the SenseCam data. The high resolution of this image is available in CGembedding.jpg. While this 64×64 tiling only shows less than 10% of all data, it does allow for quick discovery of typical scenes. For example, a large contiguous area close to the left edge contains images taken in the office. Interestingly the variation in the vertical direction corresponds largely to the angle of view change (as we move up in the grid, the camera angle points more and more towards the ceiling). Just below this area is an area filled with a few images of the subject's kitchen, with most images taken in the morning with kids around. The living room images are mostly found at the top and bottom (CGs are mapped on a torus, and so the top and the bottom correspond to the same area in the mod 64 sense on a 64-cell-tall grid). Thus this embedding can be a starting point for powerful visualization-driven tools. Suppose that the user wants to set a reminder that should go off at dinner time (e.g., to take a pill with food, or to discuss an interesting story he or she heard on NPR on the way to work). Then, instead of searching for an image taken in the dining room by flipping through images in temporal order, such an image can be quickly spotted in the upper left area of the grid where a cluster of dining room images occupy a contiguous area. Furthermore, instead of using just a couple of images from there and attaching a reminder to them, the user could lasso the entire area, specify the length of time that needs to be spent in this area before the reminder goes off and thus create a very reliable just-in-time notification trigger. To further illustrate the clustering of the images, we use the fact that the images are taken in temporal order (albeit very sparsely, every 20 seconds or so), and compute a transition statistic

$$r_\ell(\Delta) = \frac{\sum_t q(\ell_t = \ell) q(\ell_{t+1} = \ell + \Delta)}{\sum_t q(\ell_t = \ell)}, \qquad (1)$$

for $\Delta \in \{-1, 0, 1\} \times \{-1, 0, 1\}$. In Fig. 1-Left), we show the negative entropy of this distribution, $-H(\ell) = \sum_\Delta r_\ell(\Delta) \log r_\ell(\Delta)$ next to the tiled visualization of the embedding Fig. 1-Right. The entropy image reveals 'walls' among areas of high visual similarity whenever there is a small likelihood of jumping from one 'room' to the neighboring one, as we move from one time point in the acquisition to the next. For example, the office area is broken into two sections, one taken at work, and the other in the home offices of the subject. (Examining closely the ver-

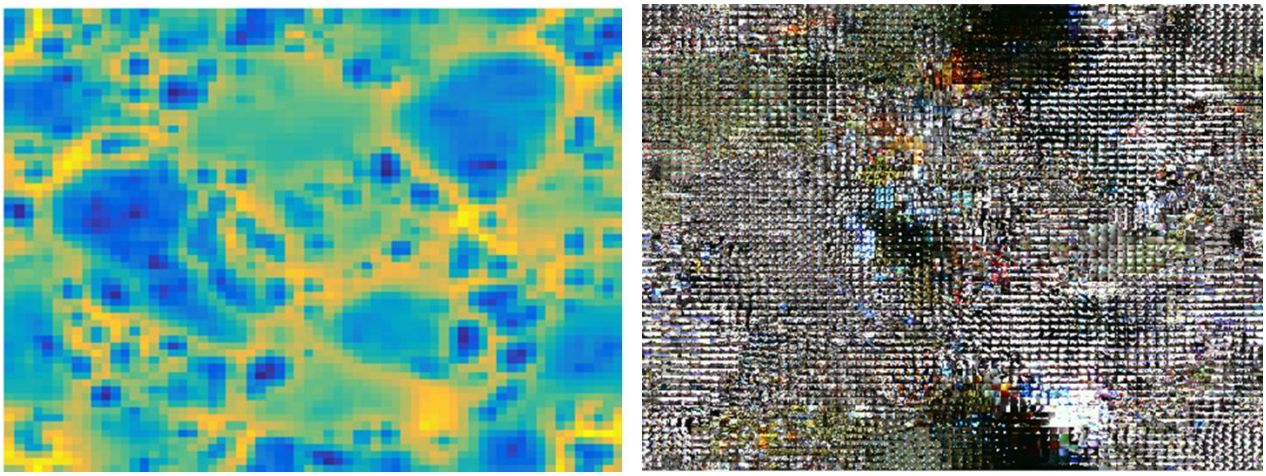

Figure 1. Left: The negative entropy $-H(\ell)$ of the transition statistic at each location ℓ reveals the transition 'walls' among visually consistent areas. The dark spots indicate natural clusters. Right: The most likely image in each location ℓ.

tical boundary, we see that the images on either side are structurally very similar, with the same pattern of vertical variation reflecting angle of view change on both sides of the 'wall,' i.e. both at work and at home. The boundary, however, indicates that these two are different clusters as the subject's images are typically mapped on one side of the boundary or the other).

We also created a video sequence TheAmericanDream.avi, where a random walk according to $r_\ell(\Delta)$ is run for 1650 frames. The video starts at a location ℓ in the subject's living room, after which $r_\ell(\Delta)$ is sampled and the location moved to $\ell + \Delta$. The process is repeated with occasional random jumps to avoid spending too much time in one area. The left panel shows the locations of the last few frames in the 'wall map' $-H(\ell)$described above. The central frame shows an image that maps to the current location. As each location represents multiple (in some cases hundreds of) images, the image to show is sampled based on the quality of fit $p(x_t|\ell)$. The right panel is a 7×24 rectangle in which the current frame's capture time is shown as (day, hour), with days starting with Sunday and ending with Saturday. The one-minute video should be viewed several times at various playback speeds and also with pausing to reveal that this sampling procedure pulls highly similar images taken on very different days. It also quickly reveals natural patterns, such as the scattering of images of office across different days in earlier hours of the day; the grill coming into visual field with a variety of food across different days in the evening just before the dining room images are typically taken; the biking scenes and garden images mostly captured on weekends (Sunday on the top and Saturday on the bottom).

Thus, the embedding can indeed lead to a variety of interesting applications, e.g., correlating the time spent in any of the 'visual rooms' with health indicators (e.g. sleep quality pulled from the FitBit); adding creative notifications (like 'remind me to check if I turned off the grill 5 minutes after it was last spotted') which the hardware on camera or the phone can flash after appropriate visual detection in real time; visual search ('where did I leave that sweater?'); tracking the growth of children; placing high res photos taken with phone camera in the context of the more frequent lower quality acquisition by something like a SenseCam, etc. In addition, GPS, accelerometers, and the context of smart phone use can all be combined with the visual stream, and streams can be shared. For example, when the subject's GPS location indicates that he is in the grocery store, he could be reminded to get yogurt not by a phone call from a family member, but by an addition (in real time) of a reminder into the 'dairy section of the store' visual room, as the subject always buys milk but tends to forget the yogurt. The notification can be added by a family member complete with the photo of the exact brand that is needed.

Te visualizations here are just a starting point, of course. Instead of an image tiling, a larger image could be chosen to represent an entire room (or surfaced on top of the tiling). Mouse-over or touch can be used to pop-up the mapped images, and the timeline graph can be used to indicate all the times when these images were collected so that the user can go back and forth between temporal and embedding browsing, etc.

The point is that a 2D embedding is relatively easy to browse, and that the quality of the embedding is good enough to surface the natural clusters even early in the subject's use of the wearable camera, making the low-labeling-effort scenarios possible. Importantly, the ability to start such applications without a lot of labeled data would ensure faster adoption, as discussed in the main paper.

Table 1. kNN classification with small number of labeled exemplars. In multiframe cases, the majority vote over three frames are used to classify the middle frame.

	Exemplars					
Method	1	2	3	4	5	10
CG-HMM	0.3157	0.3894	0.4495	0.4828	0.5169	0.5561
fc6	0.09081	0.2092	0.2911	0.3473	0.4107	0.4784
tSNE	0.1708	0.3203	0.4016	0.4650	0.5102	0.5517
CG-HMM-Multiframe	0.3412	0.4236	0.4855	0.5118	0.5283	0.6013
fc6-Multiframe	0.1134	0.2277	0.3218	0.3615	0.3902	0.5007
tSNE-Multiframe	0.1718	0.3167	0.4255	0.4838	0.5238	0.5790

3. Comparison with tSNE embedding

In analyzing wearable camera images, we focused on CG-based models as previous work on the SenseCam dataset had most success with embeddings based on this model (alternatives include panoramic epitome models with structure elements instead of colors). Approaches using pairwise distances (LLE, ISOMAP), and various linear embeddings such as PCA usually did not match CG models in performance. However, given that this work uses new, more powerful features, the question is if popular embedding methods may also work well with these features. Here we compare our embedding with tSNE [6], currently perhaps the most popular embedding tool. This tool shares some appealing properties with CGs. The data is less likely to end up clumped in distant clusters, allowing for potentially better visualizations than LLE and ISOMAP. Although the method is based on pairwise distances, the tool is practical for large datasets as accelerated algorithms that estimate the embedding in $O(T \log T)$ time for T samples have been developed (CGs are of still lower, $O(T)$ complexity).

The embedding of 43516 images from the SenseCam data is illustrated in Fig. 2 Left) where each image is represented as a circle to illustrate the mapping spread and density, and Fig. 2 Right) where a uniform tessellation of the space is created and cells each filled with one of the images that fell in the cell (high res version is attached in tSNE.pdf). This allows a visual comparison with CGembedding.jpg (note that in tSNE embedding there is no notion of wraparound as is the case in CGs which are mapped on a torus). While the embedding is quite reasonable, identifying some main classes such as the work office, the home office, car, and dining room, there are also undesirable effects such as splitting the living room based on the lighting condition, and scattering kitchen images, as well as the less uniform use of the space (Fig. 2 left) compared to CGs which update all positions during learning and use the entire space to increase the likelihood of the data. Numerically, these differences contribute to lower kNN classifications in low-labeling regime with 1-10 labeled exemplars per class, as shown in Table 1. The numbers are especially low compared to CGs when only 1-3 exemplars per class are provided as a training set.

3.1. Google Glass life logging experiment

The objective of this experiment is to analyze the performance of our approach in abundance of annotated data. The recent Google Glass dataset consists of $660,000$ seconds of egocentric video streams collected by three subjects named A, B, and C. Differently from the main paper, in the additional material we used the same experimental set-up suggested in [4]. We compared our method with all the baseline algorithms and DMA [5], which is state-of-the-art in this dataset. In particular, it used two network architectures a CNN and a shallow network, that tries to manage domain shift between source and new target data in an online manner.

Table 2 summarizes the results. Once again we observed that CG models consistently outperform the competitors by a large margin. The technique, introduced here (e.g., CG^{CW}) help to improve significantly the quality of the latent space, and thus the accuracy.

One can also observe that for a large amount of training samples, high dimensional autoencoders outperform other methods, which yet gives satisfactory results. However, higher dimensional embeddings can not serve as a reliable visualization tool for visual lifelog which is key in large data visualization.

4. The list of labeled classes

The SenseCam contains 43516 images, of which 5860 were labeled manually. The 45 labels are:

1 - Bathroom Home
2 - Bedroom Home
3 - Biking
4 - Cafeteria Work
5 - Car
6 - School A Inside
7 - Conference Room
8 - Corridors Work
9 - Dining Room Home
10 - Bakery
11 - Garage Home

Table 2. Comparison of average accuracy on Google Glass with baselines. The final accuracy is computed as the average of three available annotated levels, namely "location", "sub-location" and "Activity", for each subject A, B, and C. 10-folds with 20 times of repetition using kNN classifier is used for this task.

	Subject		
Method	A	B	C
CG [2]	0.8256	0.7864	0.7951
CG^{CW}	0.8337	0.8012	0.8103
$tSNE$ [6]	0.7859	0.7052	0.7993
$fc6$ [3]	0.6117	0.5579	0.732
DMA [5]	0.6702	0.588	0.7757
$2D$-Autoencoder [1]	0.6096	0.695	0.6193
$50D$-Autoencoder [1]	0.9308	0.893	0.9254
$100D$-Autoencoder [1]	0.9384	0.8963	0.9269
200D-Autoencoder [1]	**0.9390**	**0.9037**	**0.9277**

12 - Atrium
13 - Entry
14 - Hiking
15 - Ice palace
16 - Kids Bedroom Home
17 - Kids Game Room Home
18 - Kitchen Home
19 - Living Room Home
20 - Lounge
21 - Office Home
22 - Campus
23 - Parking Work
24 - Patio Home
25 - Playground
26 - Restroom Work
27 - Small Bathroom Home
28 - Small Home Office
29 - Tennis Court
30 - Food Court
31 - Grocery Store 1
32 - Work office
33 - Coffee House 2
34 - Garden
35 - School M Inside
36 - Front Home
37 - School A Outside
38 - FuseBall Room
39 - Dance
40 - Coffee House
41 - Post Room
42 - Hallway 1st Floor
43 - Fred Meyer grocery store
44 - Print Room
45 - School M Outside

References

[1] Y. Bengio et al. Learning deep architectures for ai. *Foundations and trends® in Machine Learning*, 2(1):1–127, 2009. 4

[2] N. Jojic and A. Perina. Multidimensional counting grids: Inferring word order from disordered bags of words. *arXiv preprint arXiv:1202.3752*, 2012. 4

[3] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012. 4

[4] S.-W. Lee, C.-Y. Lee, D. H. Kwak, J. Kim, J. Kim, and B.-T. Zhang. Dual-memory deep learning architectures for lifelong learning of everyday human behaviors. *International Joint Conference on Artificial Intelligence (IJCAI 2016)*, 2016. 3

[5] S. Song, V. Chandrasekhar, B. Mandal, L. Li, J.-H. Lim, G. Sateesh Babu, P. Phyo San, and N.-M. Cheung. Multimodal multi-stream deep learning for egocentric activity recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 24–31, 2016. 3, 4

[6] L. van der Maaten and G. Hinton. Visualizing high-dimensional data using t-sne. *Journal of Machine Learning Research*, 9:2579–2605, 2008. 3, 4

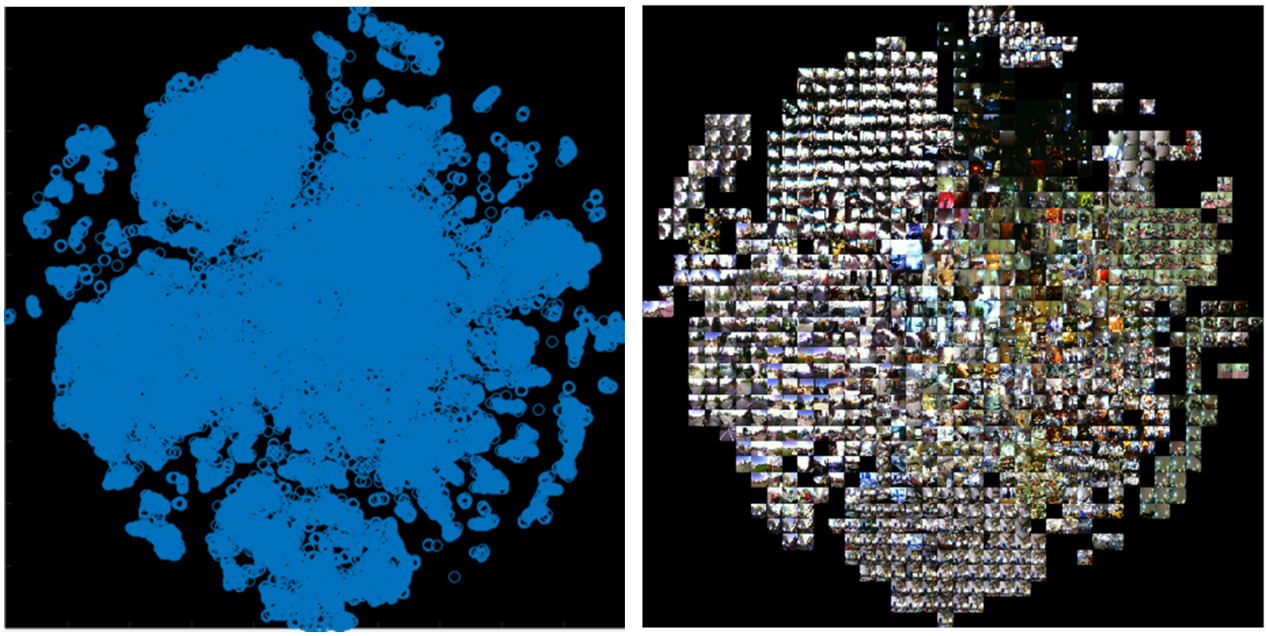

Figure 2. Left: The embedding of 43516 images from the SenseCam data, each image is represented as a circle to illustrate the mapping spread and density. Right: a uniform tessellation of the space is created and cells each filled with one of the images that fell in the cell.