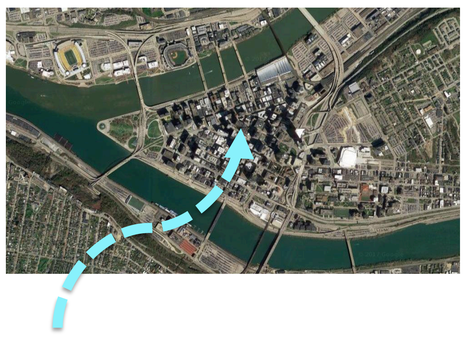
Panorama to panorama matching for location recognition

Ahmet Iscen[1], Giorgos Tolias[2], Yannis Avrithis[1], Teddy Furon[1], Ondřej Chum[2]
[1]Inria, [2]Visual Recognition Group, CTU in Prague

Problem formulation

Query (image)

query descriptor $\mathbf{Q}$



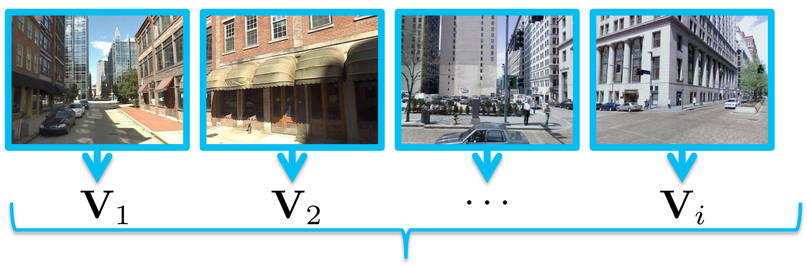
$$i^\star = \arg\max_{i \in 1 \dots N} \; s(\mathbf{Q}, \mathbf{V}_i)$$

Location recognition by NN search

Self driving cars

Query (image set from the same location)

query descriptor $\mathbf{Q}$



Descriptors for database images

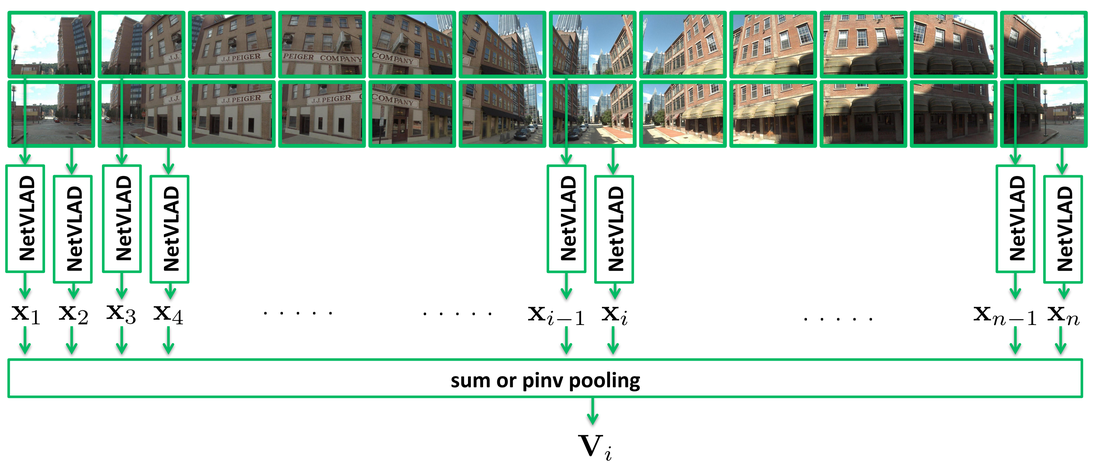
Image to image matching

Descriptors for database image sets → each set from the same location

Panorama to Panorama matching

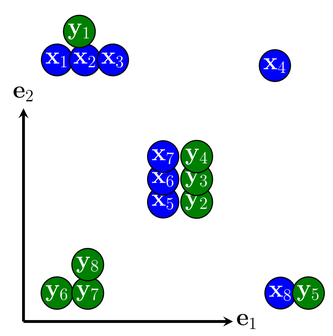
Contribution 1: Implicit Panorama Construction — pan2pan/sum & pan2pan/pinv

- Use NetVLAD [1] for single image descriptor
- Joint representation of image set by pooling in the descriptor space [3]
- Index one vector (joint representation) per location

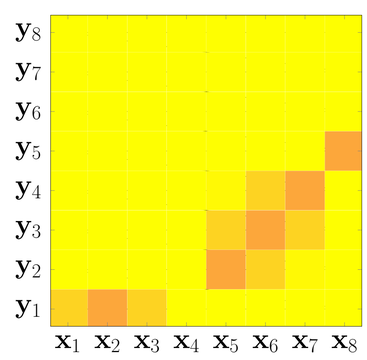


- $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_i, \dots, \mathbf{x}_n]$
- pan2pan/sum pooling: $m(\mathbf{X}) = \mathbf{X}\mathbf{1}_n$
- pan2pan/pinv pooling: $m^+(\mathbf{X}) = (\mathbf{X}^+)^\top \mathbf{1}_n = \mathbf{X}(\mathbf{X}^\top\mathbf{X})^{-1}\mathbf{1}_n = \mathbf{X}(\mathbf{G}_\mathbf{X})^{-1}\mathbf{1}_n$
- **sum** matching is straightforward: $s(\mathbf{X}, \mathbf{Y}) = m(\mathbf{X})^\top m(\mathbf{Y}) = \mathbf{1}_n^\top \mathbf{X}^\top \mathbf{Y} \mathbf{1}_k$
- **pinv** matching is "democratized" cross-matching: $s^+(\mathbf{X}, \mathbf{Y}) = m^+(\mathbf{X})^\top m^+(\mathbf{Y}) = \mathbf{1}_n^\top \mathbf{G}_\mathbf{X}^{-1} \mathbf{X}^\top \mathbf{Y} \mathbf{G}_\mathbf{Y}^{-1} \mathbf{1}_k$

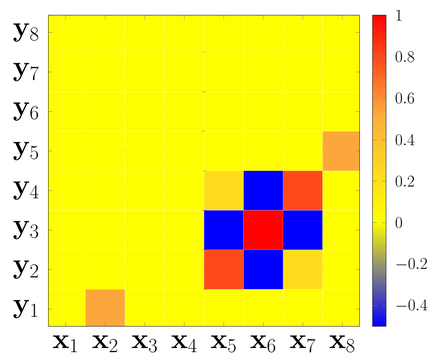
2D toy example



sum matching

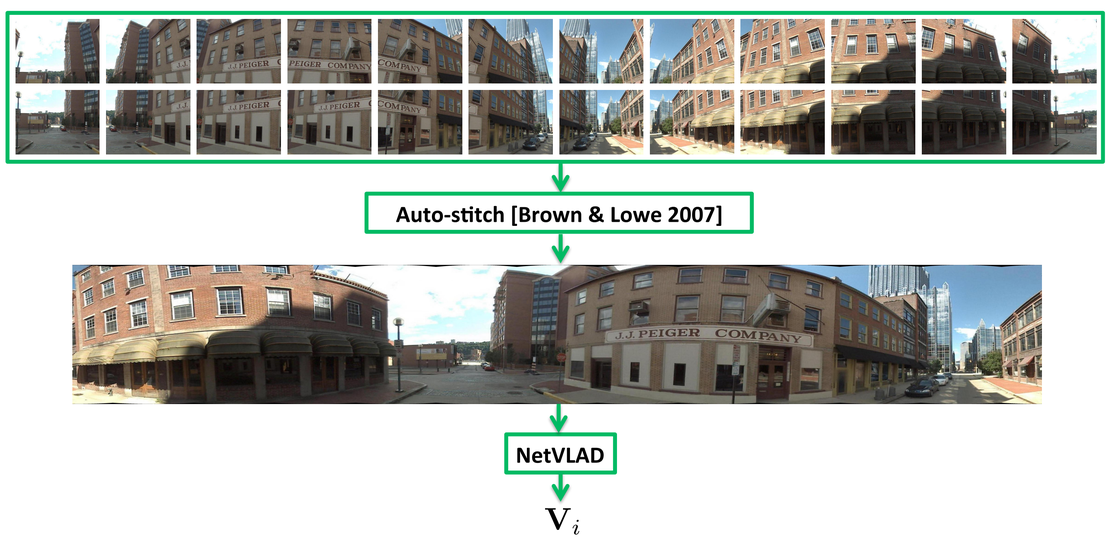


pinv matching

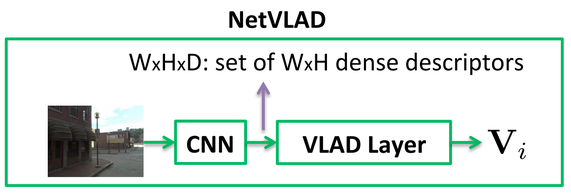


Contribution 2: Explicit Panorama Construction — pan2pan/net

- Auto-stitch images of the same location [2]
- Use NetVLAD on the stitched (panoramic) image
- Index one NetVLAD descriptor per location



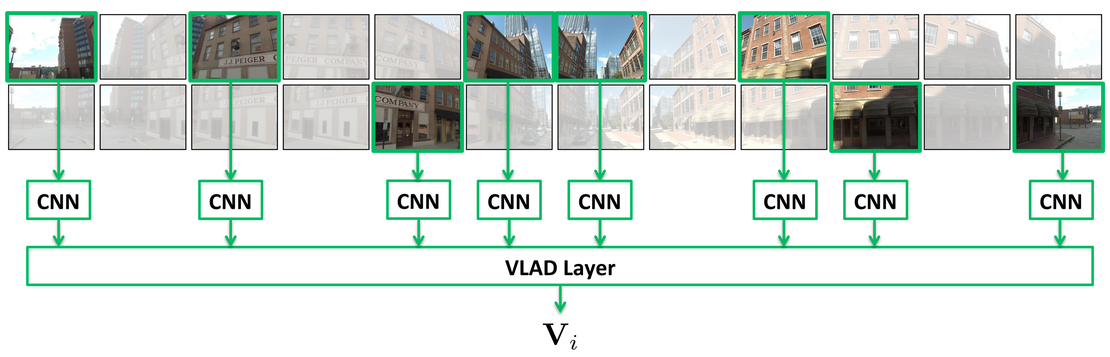
- Feature extraction with NetVLAD



Sparse panorama:

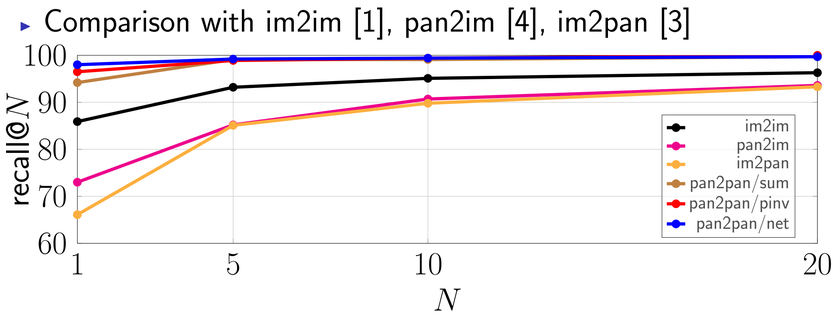
- Sparse image sequence → stitching not possible
- We adopt an *intermediate* approach

- Extract CNN activations from each image
- VLAD layer input = union of all activations

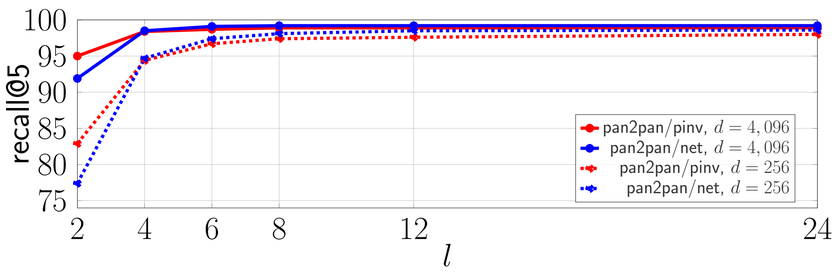


Results

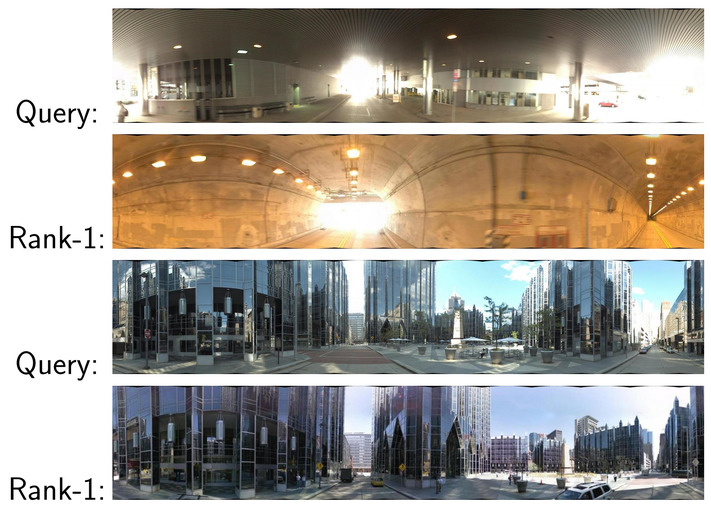
- Comparison with im2im [1], pan2im [4], im2pan [3]



Performance comparison on Pittsburgh dataset with $d = 4096$.



Performance comparison on Pittsburgh dataset with sparse panoramas of l images.



Example failures with *pan2pan/net*

- Conceptually simple method
- Significant memory savings
- Query speed-up for multiple queries
- Near optimal performance
  only 7 queries missed on Pittsburgh
- Multiple query views help
  but only 4 views seem enough

References:

[1] R. Arandjelović, P. Gronat, A. Torii, T. Pajdla, and J. Sivic. NetVLAD: CNN architecture for weakly supervised place recognition. In *CVPR*, 2016.

[2] M. Brown and D. G. Lowe. Automatic panoramic image stitching using invariant features. *IJCV*, 74:59–73, 2007.

[3] A. Iscen, T. Furon, V. Gripon, M. Rabbat, and H. Jégou. Memory vectors for similarity search in high-dimensional spaces. *IEEE Trans. Big Data*, 2017.

[4] R. Sicre and H. Jégou. Memory vectors for particular object retrieval with multiple queries. In *ICMR*, 2015.

Contact: ahmet.iscen@inria.fr