Scale-less Dense Correspondences

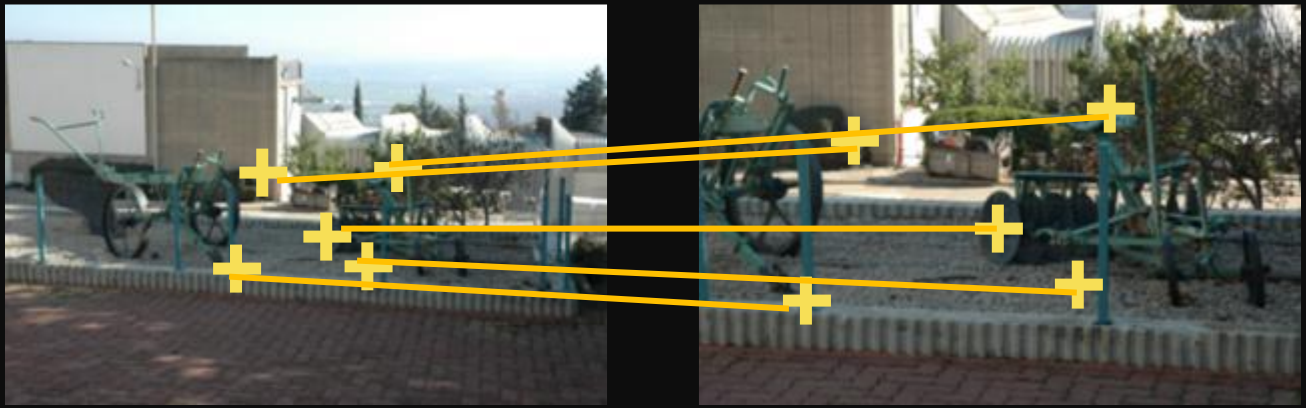
Tal Hassner
The Open University of Israel

ICCV'13 Tutorial on

Dense Image Correspondences for Computer Vision

Matching Pixels

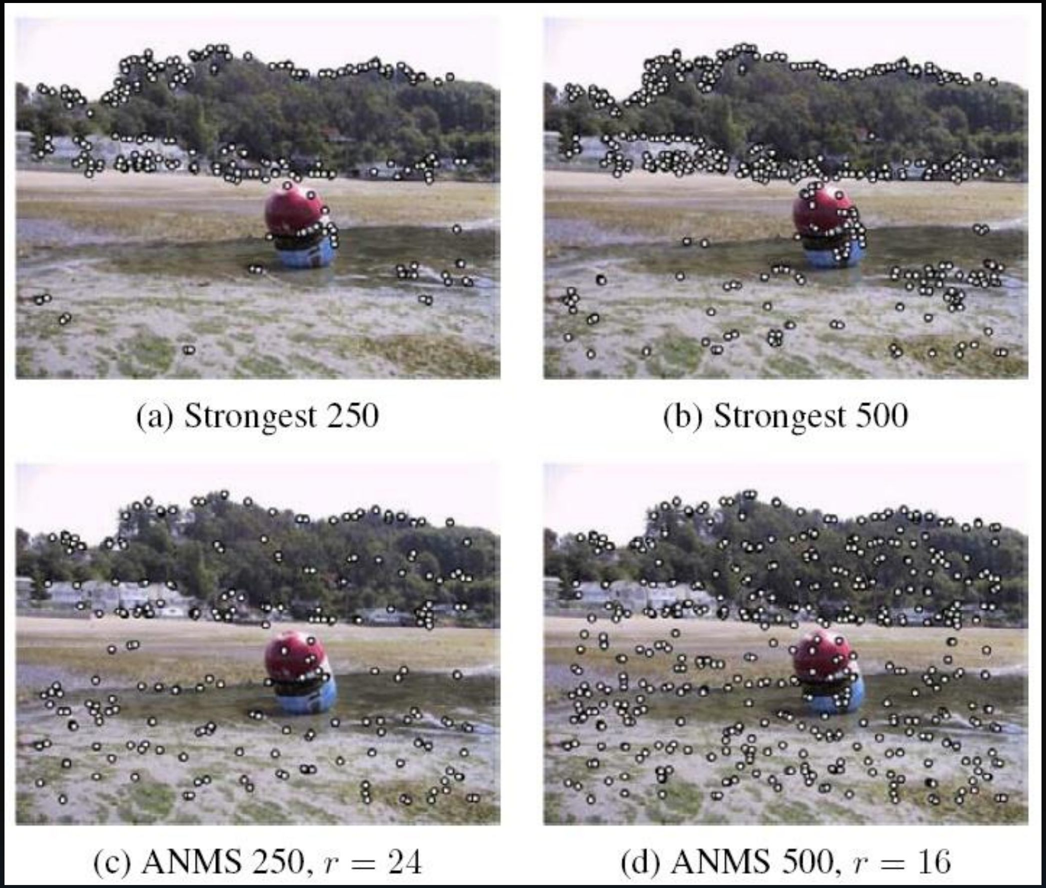
In different views, scales, scenes, etc.



Invariant detectors +
robust descriptors +
matching

Observation:

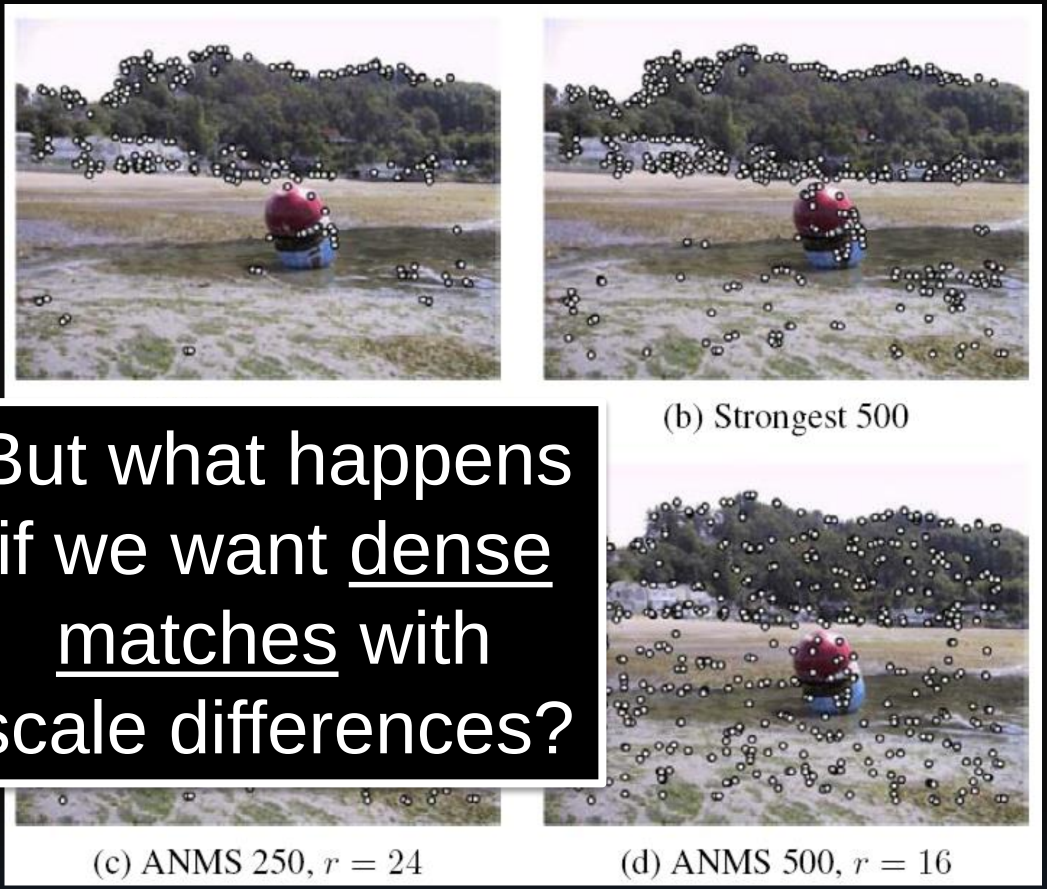
Invariant detectors require dominant scales
BUT
Most pixels do not have such scales



[Szeliski's book]

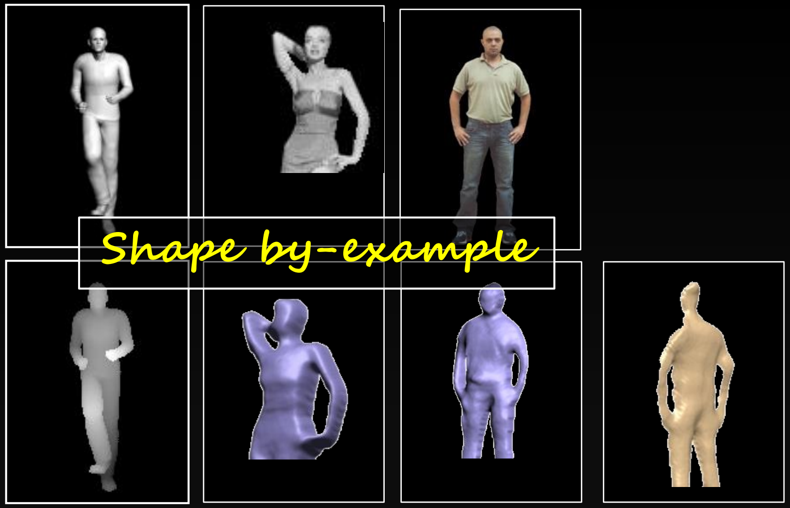
Observation:

Invariant detectors require dominant scales
BUT
Most pixels do not have such scales



But what happens if we want <u>dense matches</u> with scale differences?

[Szeliski's book]



[Hassner&Basri ’06a, ‘06b,’13]

Why is this
useful?

[Hassner&Basri ’06]

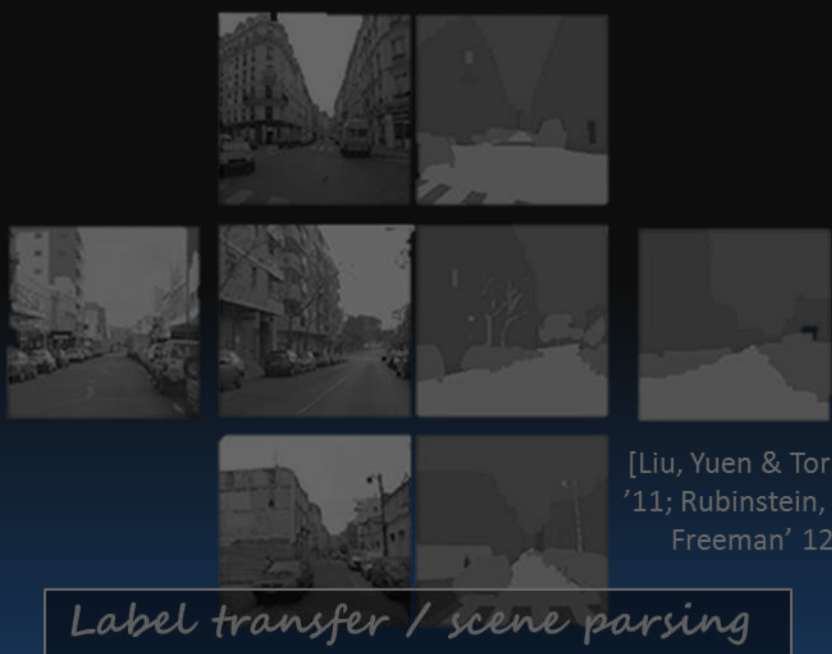
Why is this useful?

[Liu, Yuen & Torralba ’11; Rubinstein, Liu & Freeman’ 12]

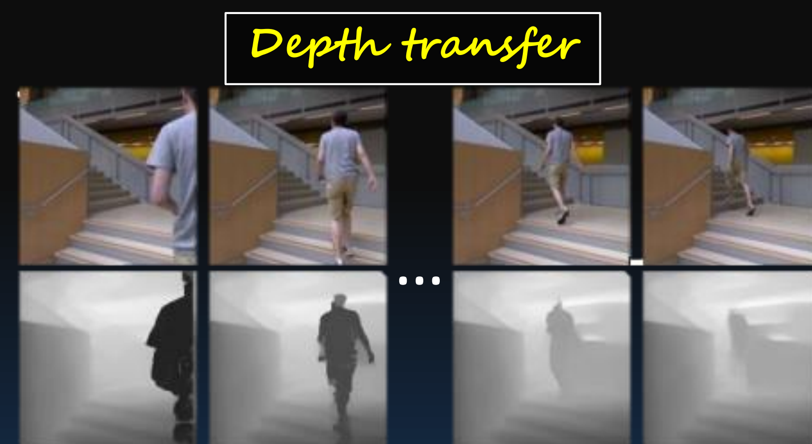
Label transfer / scene parsing

[Hassner&Basri '06]

Why is this useful?



[Liu, Yuen & Torralba '11; Rubinstein, Liu & Freeman' 12]



[Karsch, Liu & Kang '12]

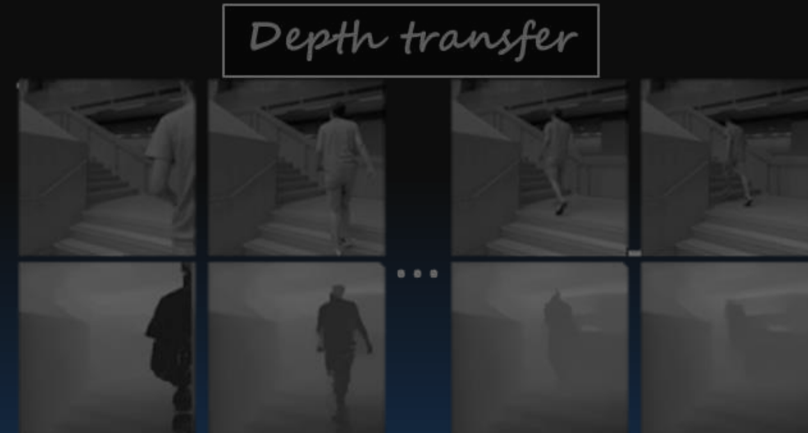
[Hassner&Basri ’06]

[Liu, Yuen & Torralba ’11]

Why is this useful?

[Hassner, Saban & Wolf]

[Liu, Yuen & Torralba ’11; Rubinstein, Liu & Freeman’ 12]



[Karsch, Liu & Kang ’12]

Shape by-example

[Hassner&Basri '06]

New view synthesis

[Hassner '13]

Face recognition

[Liu, Yuen & Torralba '11]

Fingerprint recognition

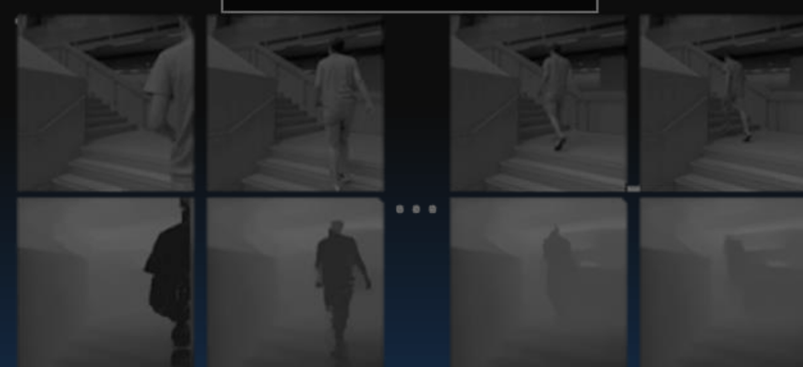
[Hassner, Saban & Wolf]

Why is this useful?

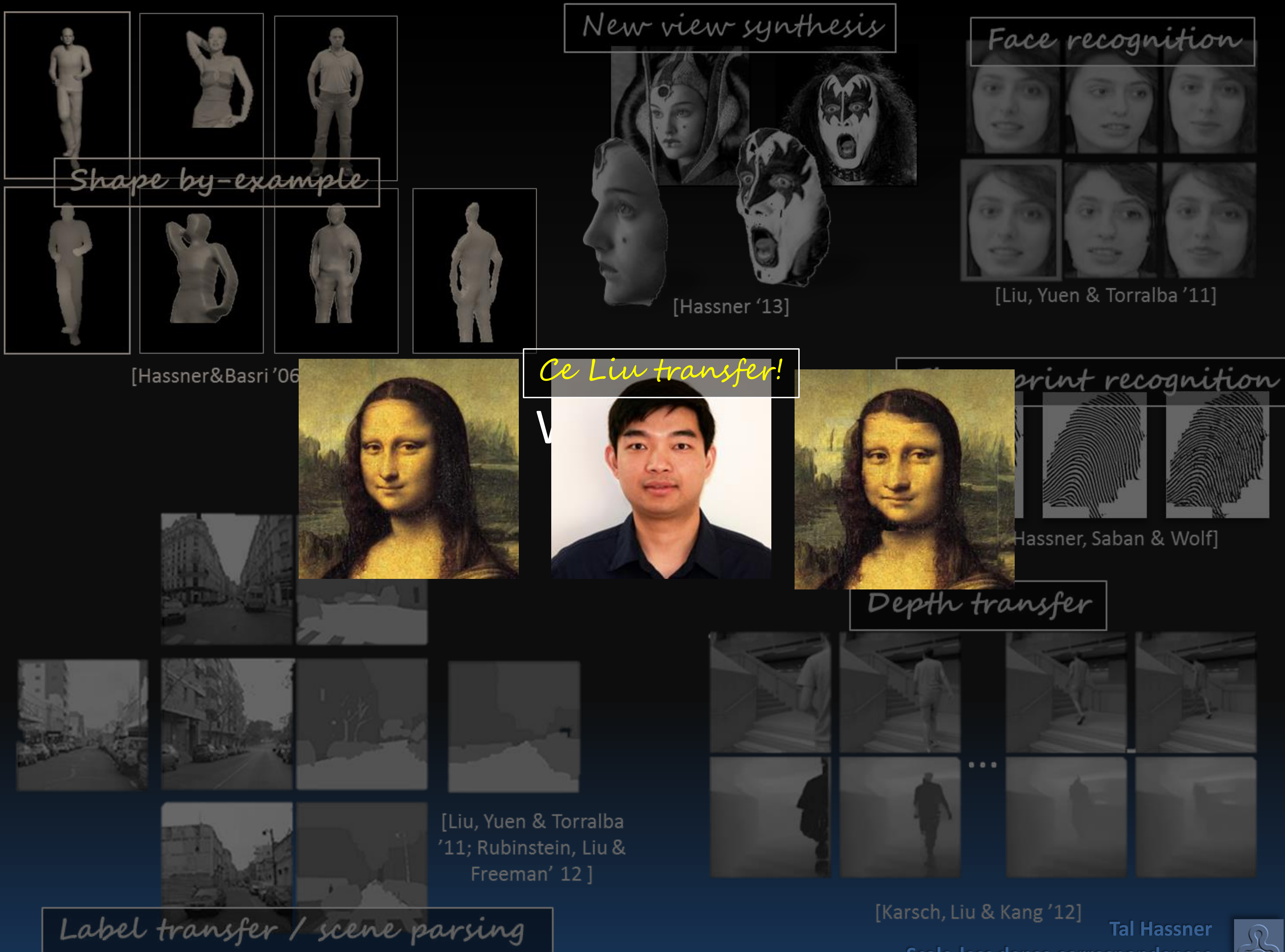
[Liu, Yuen & Torralba '11; Rubinstein, Liu & Freeman' 12]

Label transfer / scene parsing

Depth transfer



[Karsch, Liu & Kang '12]


Shape by-example
[Hassner&Basri '06
New view synthesis
[Hassner '13]
Face recognition
[Liu, Yuen & Torralba '11]
Ce Liu transfer!
print recognition
Hassner, Saban & Wolf]
Depth transfer
[Liu, Yuen & Torralba '11; Rubinstein, Liu & Freeman' 12]
[Karsch, Liu & Kang '12]
Label transfer / scene parsing

Dense matching with scale differences

Solution 1:

Ignore scale differences – Dense-SIFT

Dense SIFT (DSIFT)

Arbitrary scale selection

VLFeat.org

Google™ Custom Search

Tutorials - DSIFT/PHOW

- **Home**
- **Download**
- **Documentation**
- **Tutorials**
  - SIFT
  - DSIFT/PHOW
  - MSER
  - IKM
  - HIKM
  - AIB
  - Quick shift
  - kd-tree
  - Distance transf.
  - Utils
- **Applications**

VLFeat implements a fast dense version of SIFT, called `vl_dsift`. The function is roughly equivalent to running SIFT on a dense gird of locations at a fixed scale and orientation. This type of feature descriptors is often uses for object categorization.

Dense SIFT as a faster SIFT

The main advantage of using `vl_dsift` over `vl_sift` is speed. To see this, load a test image

```
I = imread(fullfile(vl_root,'data','a.jpg')) ;
I = single(vl_imdown(rgb2gray(I))) ;
```

To check the equivalence of `vl_disft` and `vl_sift` it is necessary to understand in detail how the parameters of the two descriptors are related.

- **Bin size vs keypoint scale**. DSIFT specifies the descriptor size by a single parameter, `size`, which controls the size of a SIFT spatial bin in pixels. In the standard SIFT descriptor, the bin size is related to the SIFT keypoint scale by a multiplier, denoted `magnif` below, which defaults to `3`. As a consequence, a DSIFT descriptor with bin size equal to 5 corresponds to a SIFT keypoint of scale 5/3=1.66.

[Vedaldi and Fulkerson'10]

SIFT-Flow

[Liu et al. ECCV'08, PAMI'11]

Left photo

Right photo

Left warped onto Right

"The good": Dense flow between *different scenes*!

SIFT-Flow

[Liu et al. ECCV’08, PAMI’11]

Left photo

Right photo

Left warped onto Right

“The bad”: Fails when matching different scales

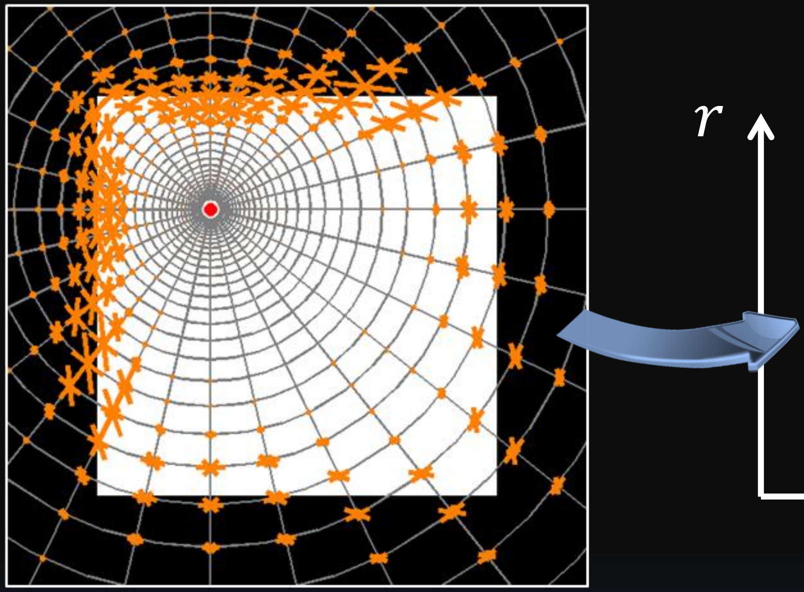
Dense matching with scale differences

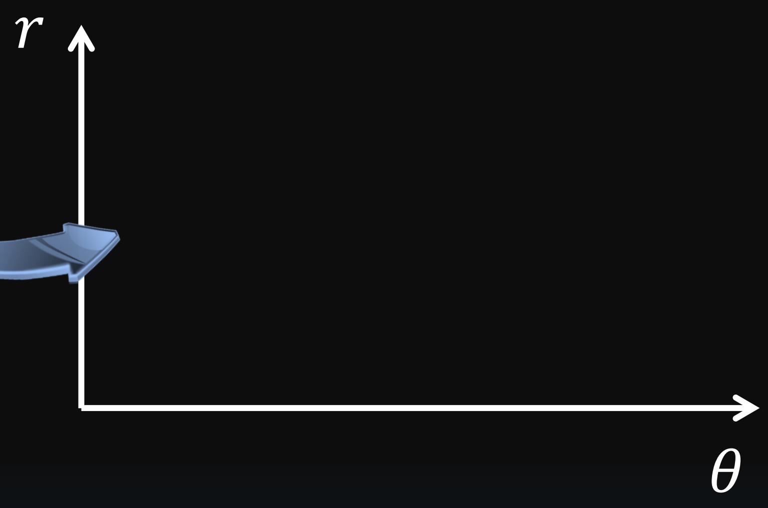
Solution 2:

Scale Invariant Descriptors (SID)*

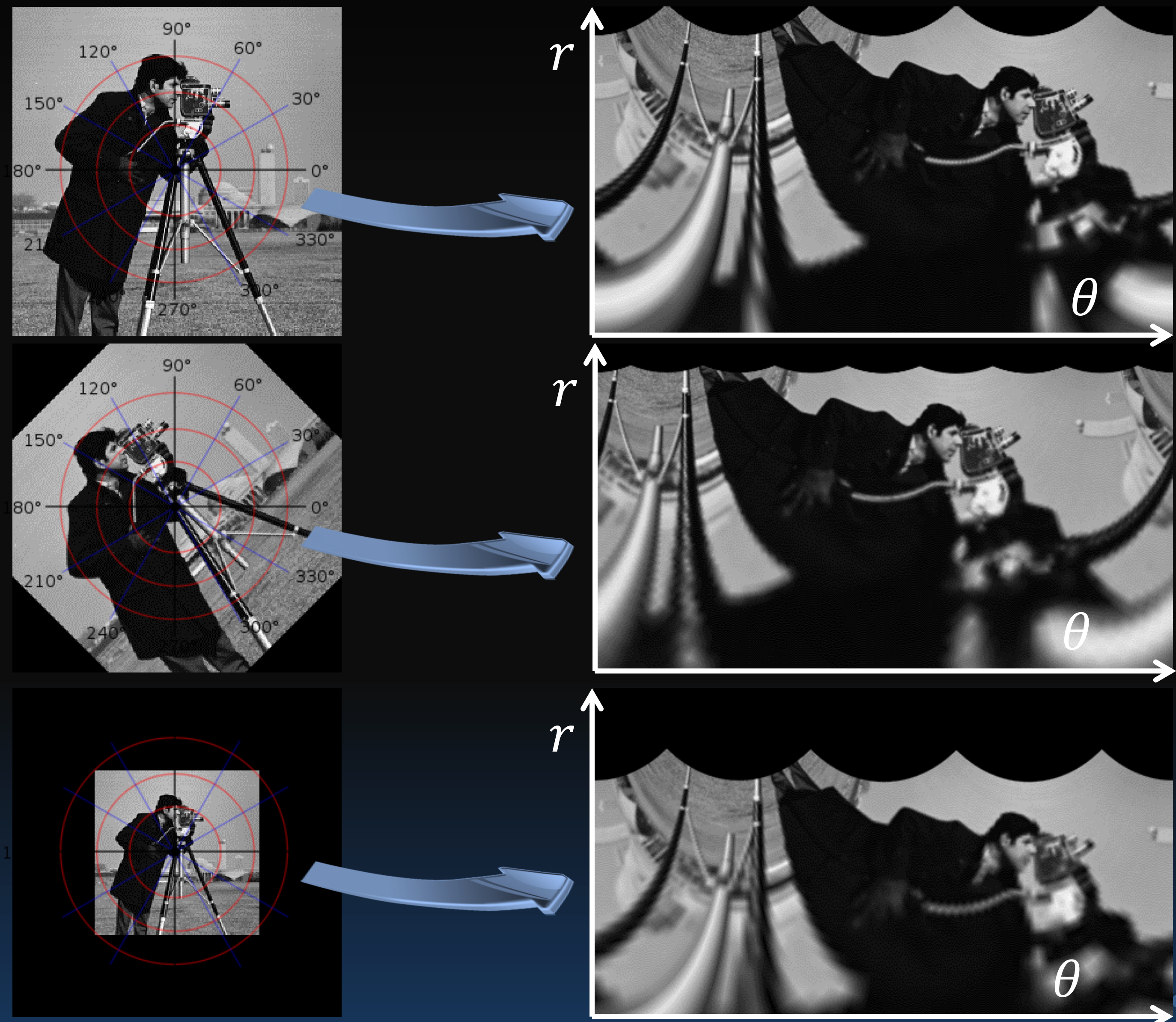
* Kokkinos and Yuille, *Scale Invariance without Scale Selection*, IEEE Conf. on Computer Vision and Pattern Recognition (CVPR), 2008

Log-Polar sampling



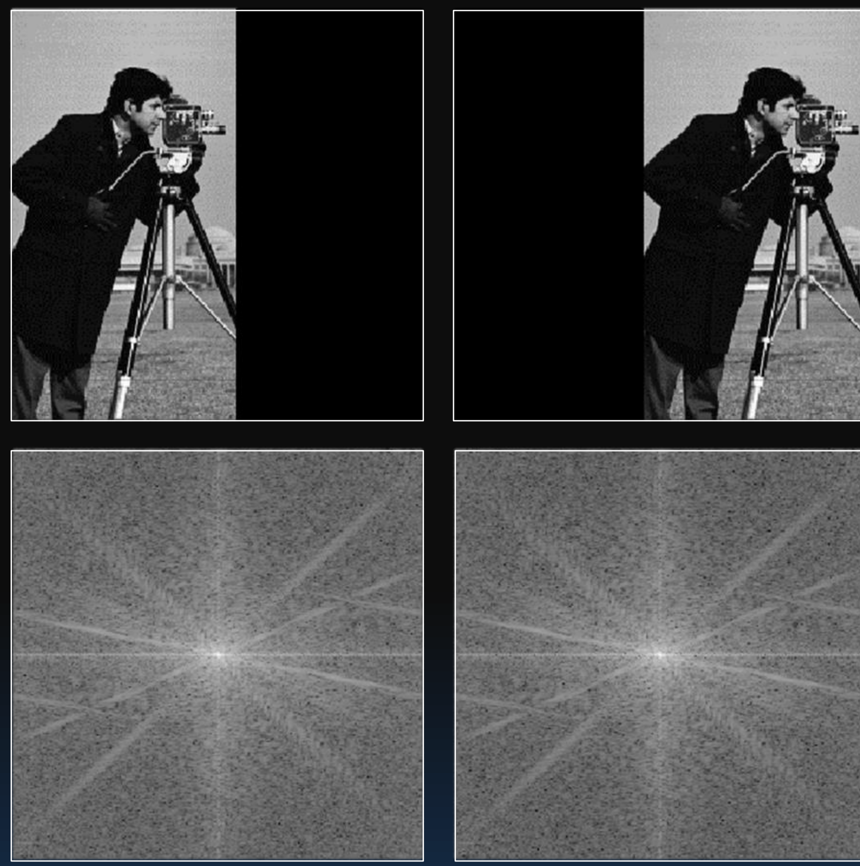


From Rotation + Scale to translation



Translation invariance

Absolute of the Discrete-Time Fourier Transform



SID-Flow

Left

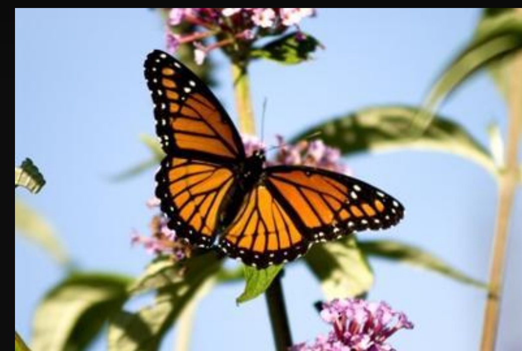
Right

DSIFT

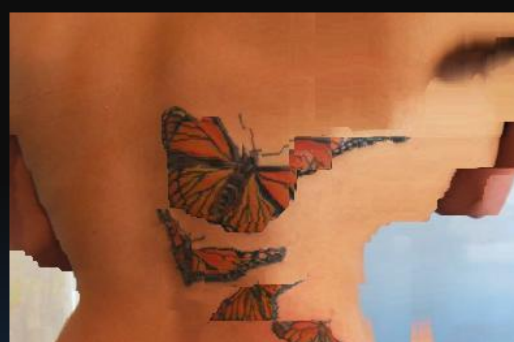
SID

SID-Flow

Left



Right



DSIFT

SID

Dense matching with scale differences

Solution 3:

Scale-Less SIFT (SLS)*

Joint work with
Viki Mayzels and Lihi Zelnik-Manor and

* T. Hassner, V. Mayzels, and L. Zelnik-Manor, *On SIFTs and their Scales*, IEEE Conf. on Computer Vision and Pattern Recognition (CVPR), Rhode Island, June 2012

SIFTs and Multiple Scales

SIFTs and Multiple Scales

$\sigma_1, \ldots, \sigma_k$

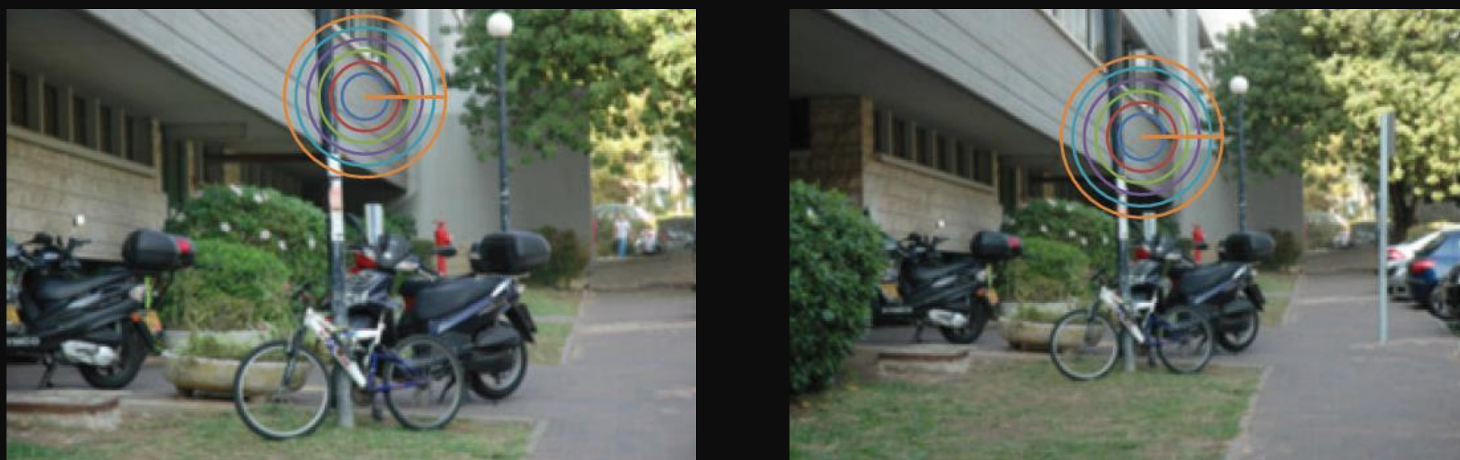
$\sigma_1, \ldots, \sigma_k$

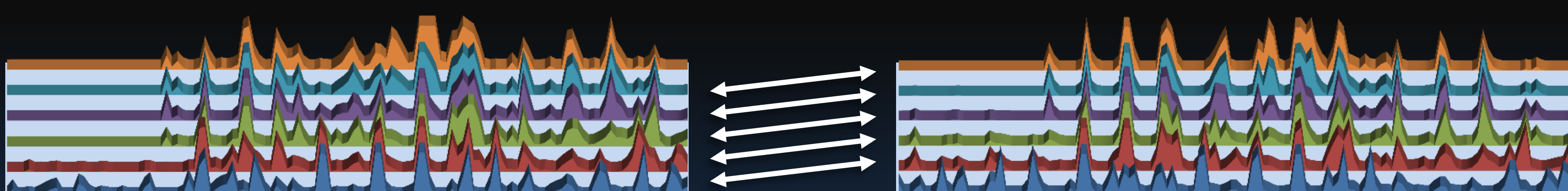
$$\left[\mathbf{h}_{\sigma_1}, \ldots, \mathbf{h}_{\sigma_k}\right]$$

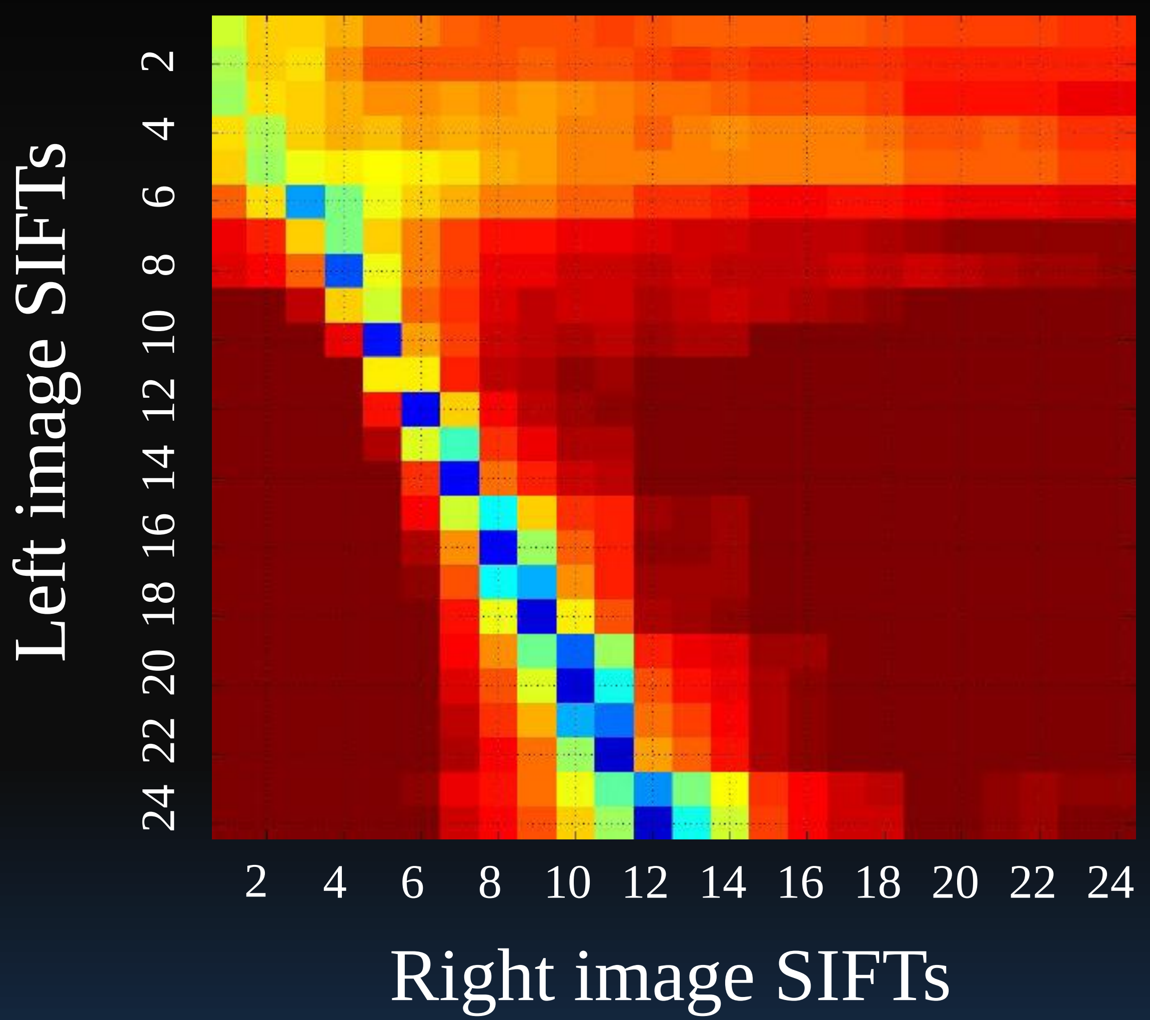
$$\left[\mathbf{h}'_{\sigma_1}, \ldots, \mathbf{h}'_{\sigma_k}\right]$$

Observation 1

Corresponding points have multiple SIFT matches at multiple scales






Left image SIFTs
2
4
6
8
10
12
14
16
18
20
22
24
2 4 6 8 10 12 14 16 18 20 22 24
Right image SIFTs

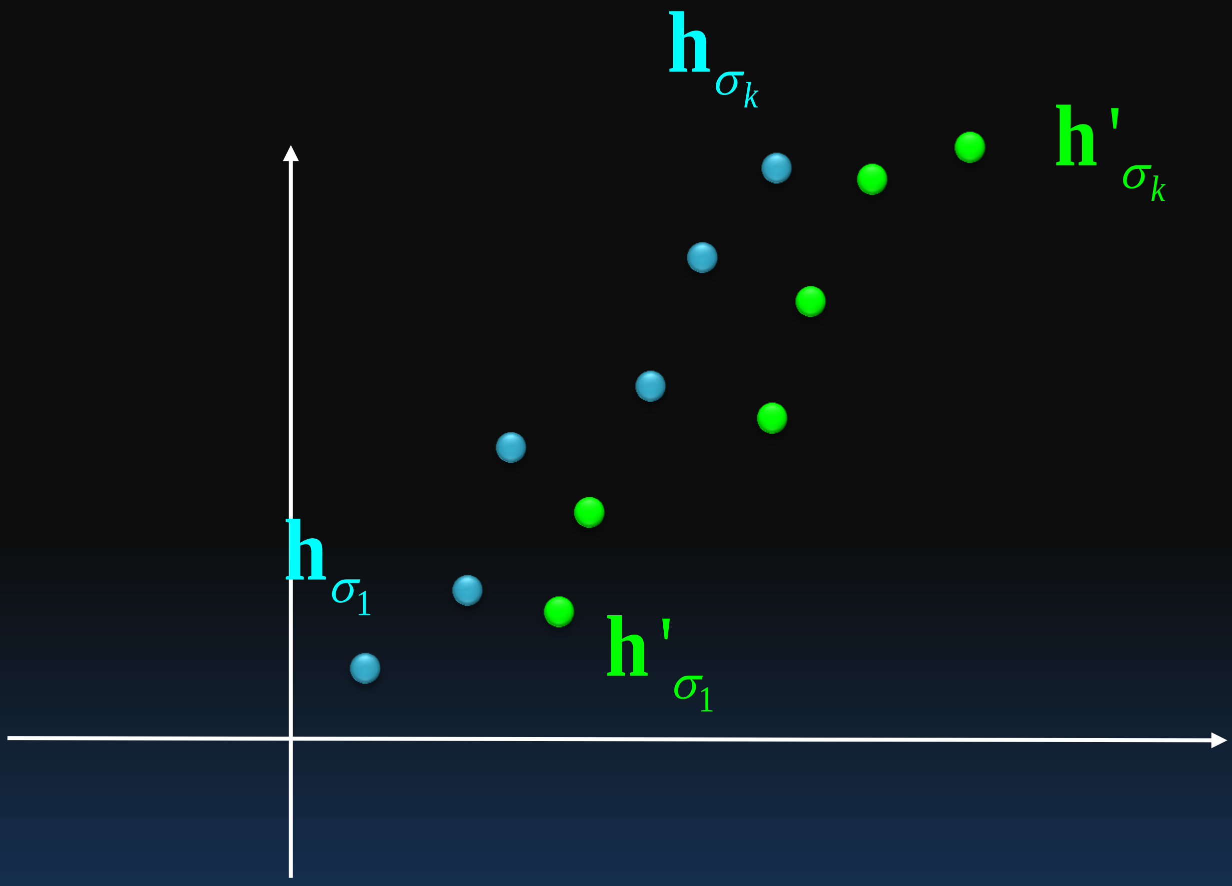
Matching ver.1

Use set-to-set distance:

$$dist(\mathbf{p},\mathbf{p}') = \min dist(\mathbf{h}_{\sigma_i},\mathbf{h}'_{\sigma_j})$$

To Illustrate

…if SIFTs were 2D



Observation 2

SIFT changes gradually across scales

Suggests they reside on manifold

Main Assumption

SIFTs in multi-scales lie close to a linear subspace

Fixed local statistics : $$\mathbf{h}_{\sigma_i} = \mathbf{h}_{\sigma_j}$$

Gradual changes across scales: $$\mathbf{h}_{\sigma_i} = \sum_j w_{ij} \mathbf{h}_{\sigma_j}$$

$$\mathbf{H} = \left[\mathbf{h}_{\sigma_1}, \ldots, \mathbf{h}_{\sigma_k} \right] = \underbrace{[\mathbf{h}_1, \ldots, \mathbf{h}_b]}_{\text{basis}} \mathbf{W} = \hat{\mathbf{H}} \mathbf{W}$$

So, for each pixel…

Extract SIFTs at multi-scales

$$\left[\mathbf{h}_{\sigma_1}, \ldots, \mathbf{h}_{\sigma_k}\right]$$

Compute basis (e.g., PCA)

$$\hat{\mathbf{H}} = \left[\mathbf{h}_1, \ldots, \mathbf{h}_b\right]$$

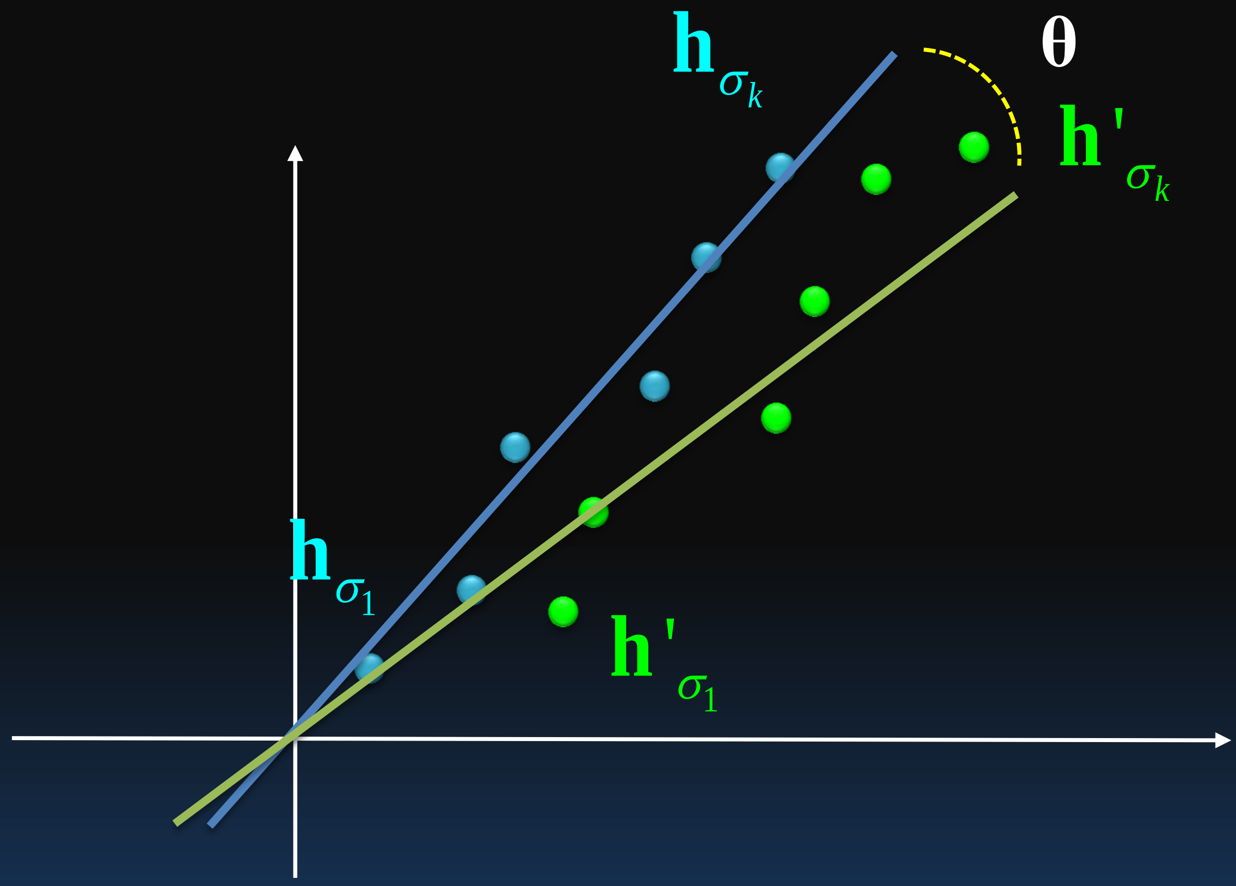
This low-dim subspace reflects SIFT behavior through scales

Matching ver.2

Use subspace to subspace distance:

$$dist(\mathbf{p},\mathbf{q}) = dist(\hat{\mathbf{H}}_{\mathbf{p}},\hat{\mathbf{H}}_{\mathbf{q}}) = \|\sin\boldsymbol{\theta}\|_2^2$$

To Illustrate



The Scale-Less SIFT (SLS)

Map these subspaces to points!

[Basri, Hassner, Zelnik-Manor, CVPR'07, ICCVw'09, TPAMI'11]

For each pixel $\mathbf{p}$

$$\mathbf{A}_{\mathbf{p}} = \hat{\mathbf{H}}_{\mathbf{p}} \hat{\mathbf{H}}_{\mathbf{p}}^{T}$$

$$SLS(\mathbf{p}) = Vec(\mathbf{A}_{\mathbf{p}}) = \left[\frac{a_{11}}{\sqrt{2}}, a_{12}, \ldots, a_{1D}, \frac{a_{22}}{\sqrt{2}}, a_{23} \ldots, \frac{a_{DD}}{\sqrt{2}} \right]$$

$$\left\| SLS(\mathbf{p}) - SLS(\mathbf{q}) \right\|^{2} = \mu \cdot dist^{2}\left(\hat{\mathbf{H}}_{p}, \hat{\mathbf{H}}_{q} \right)$$

The Scale-Less SIFT (SLS)

Map these subspaces to points!

[Basri, Hassner, Zelnik-Manor, CVPR'07, ICCVw'09, TPAMI'11]

A point representation for the subspace spanning SIFT's behavior in scales!!!

$$SLS(\mathbf{p}) = Vec(\mathbf{A_p}) = \left[\frac{a_{11}}{\sqrt{2}}, a_{12}, \ldots, a_{1D}, \frac{a_{22}}{\sqrt{2}}, a_{23} \ldots, \frac{a_{DD}}{\sqrt{2}}\right]$$

$$\left\|SLS(\mathbf{p}) - SLS(\mathbf{q})\right\|^2 = \mu \cdot dist^2\left(\hat{\mathbf{H}}_p, \hat{\mathbf{H}}_q\right)$$

SLS-Flow

Left Photo

Right Photo

DSIFT

SID [Kokkinos & Yuille, CVPR'08]

Our SLS

Dense-Flow with SLS

Using SIFT-Flow to compute the flow

Left Photo

Right Photo

DSIFT

SID [Kokkinos & Yuille, CVPR'08]

Our SLS

Dense-Flow with SLS

Using SIFT-Flow to compute the flow

Left Photo

Right Photo

DSIFT

SID [Kokkinos & Yuille, CVPR'08]

Our SLS

What we saw

Dense matching, even when scenes and scales are different

Thank you!

hassner@openu.ac.il

www.openu.ac.il/home/hassner

References

- **[Basri, Hassner & Zelnik-Manor ’07]** Basri, Ronen, Tal Hassner, and Lihi Zelnik-Manor. "Approximate nearest subspace search with applications to pattern recognition." Computer Vision and Pattern Recognition, 2007. CVPR'07. IEEE Conference on. IEEE, 2007.
- **[Basri, Hassner, Zelnik-Manor ’09]** Basri, Ronen, Tal Hassner, and Lihi Zelnik-Manor. "A general framework for approximate nearest subspace search." Computer Vision Workshops (ICCV Workshops), 2009 IEEE 12th International Conference on. IEEE, 2009.
- **[Basri, Hassner, Zelnik-Manor ’11]** Basri, Ronen, Tal Hassner, and Lihi Zelnik-Manor. "Approximate nearest subspace search." Pattern Analysis and Machine Intelligence, IEEE Transactions on 33.2 (2011): 266-278.
- **[Hassner ’13]** Hassner, Tal. "Viewing real-world faces in 3D." ICCV, 2013.
- **[Hassner & Basri ’06a]** Hassner, Tal, and Ronen Basri. "Example based 3D reconstruction from single 2D images." Computer Vision and Pattern Recognition Workshop, 2006. CVPRW'06. Conference on. IEEE, 2006.
- **[Hassner & Basri ’06b]** Hassner, T., and R. Basri. "Automatic depth-map colorization." Proc. Conf. Eurographics. Vol. 2006. 2006.
- **[Hassner & Basri ’13]** Hassner, Tal, and Ronen Basri. "Single View Depth Estimation from Examples." arXiv preprint arXiv:1304.3915 (2013).
- **[Hassner, Mayzels & Zelnik-Manor’12]** Hassner, Tal, Viki Mayzels, and Lihi Zelnik-Manor. "On sifts and their scales." Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on. IEEE, 2012.
- **[Hassner, Saban & Wolf]** Hassner, Tal, , Gilad Saban, and Lior Wolf, In submission
- **[Karsch, Liu & Kang]** Karsch, Kevin, Ce Liu, and Sing Bing Kang. "Depth extraction from video using non-parametric sampling." Computer Vision–ECCV 2012. Springer Berlin Heidelberg, 2012. 775-788.
- **[Kokkinos & Yuille’08]** Kokkinos, Iasonas, and Alan Yuille. "Scale invariance without scale selection." Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on. IEEE, 2008.
- **[Liu, Ce, et al.’08]** Liu, Ce, Jenny Yuen, Antonio Torralba, Josef Sivic, and William T. Freeman. "SIFT flow: dense correspondence across different scenes." Computer Vision–ECCV 2008. Springer Berlin Heidelberg, 2008. 28-42.
- **[Liu, Yuen & Torralba’ 11]** Liu, Ce, Jenny Yuen, and Antonio Torralba. "Sift flow: Dense correspondence across scenes and its applications." Pattern Analysis and Machine Intelligence, IEEE Transactions on 33.5 (2011): 978-994.
- **[Rubinstein, Liu & Freeman’12]** Rubinstein, Michael, Ce Liu, and William T. Freeman. "Annotation propagation in large image databases via dense image correspondence." Computer Vision–ECCV 2012. Springer Berlin Heidelberg, 2012. 85-99.
- **[Szeliski’s book]** Szeliski, Richard. Computer vision: algorithms and applications. Springer, 2011.
- **[Vedaldi and Fulkerson’10]** Vedaldi, Andrea, and Brian Fulkerson. "VLFeat: An open and portable library of computer vision algorithms." Proceedings of the international conference on Multimedia. ACM, 2010.

Resources

- SIFT-Flow
  - http://people.csail.mit.edu/celiu/SIFTflow/
- DSIFT (vlfeat)
  - http://www.vlfeat.org/
- SID
  - http://vision.mas.ecp.fr/Personnel/iasonas/code.html
- SLS
  - http://www.openu.ac.il/home/hassner/projects/siftscales/
- Me
  - http://www.openu.ac.il/home/hassner
  - hassner@openu.ac.il