Visual Correspondences: Matching Descriptors and Discrete Optimization

Dongbo Min

Department of Computer Science and Engineering

Chungnam National University, Korea

E-mail: dbmin@cnu.ac.kr **Web:** http://cvlab.cnu.ac.kr/

Acknowledgement: Dr. Jiangbo Lu (ADSC), Dr. Yu Li (ADSC), Prof. Minh N. Do (UIUC)

Contents

- Primarily focus on two images, i.e. pairwise matching (e.g. stereo, optical flow, wide-baseline, cross scene), though sometimes with more images

What we learn today
= ICME 2015 tutorial (3 hours) + Work on discrete labeling optimization – Many works on applications

Outline

- **Part 0**: Introduction (5 min)
- **Part 1**: Evaluating matching evidence (40 min)
- **Part 2**: Regularizing the estimates: labeling optimization (40 min)
- **Part 3**: Conclusions and future directions (5 min)
- Q & A

[1] Indeed, one of the oft-told stories is that when a student asked Takeo Kanade what are the three most important problems in computer vision, his reply was: "Alignment, alignment, alignment!". [Aubry et al., CVPR'14]

Correspondence, correspondence, correspondence

- Image alignment
- Image registration
- Image matching
- Optical flow
- Stereo

A number of challenges

- Large displacement
- Non-rigid motion
- Independent object motion
- Small objects

→ Robust

- Photometric differences (e.g. exposure, tone, sharpness)
- Weakly textured regions

- Matching across different scene contents

→ Dense

- Motion coherence vs. boundary/detail preserving
- Precision vs. recall, density, spatial coverage/distribution

- Computational load
- Memory cost
- Large hypothesis space

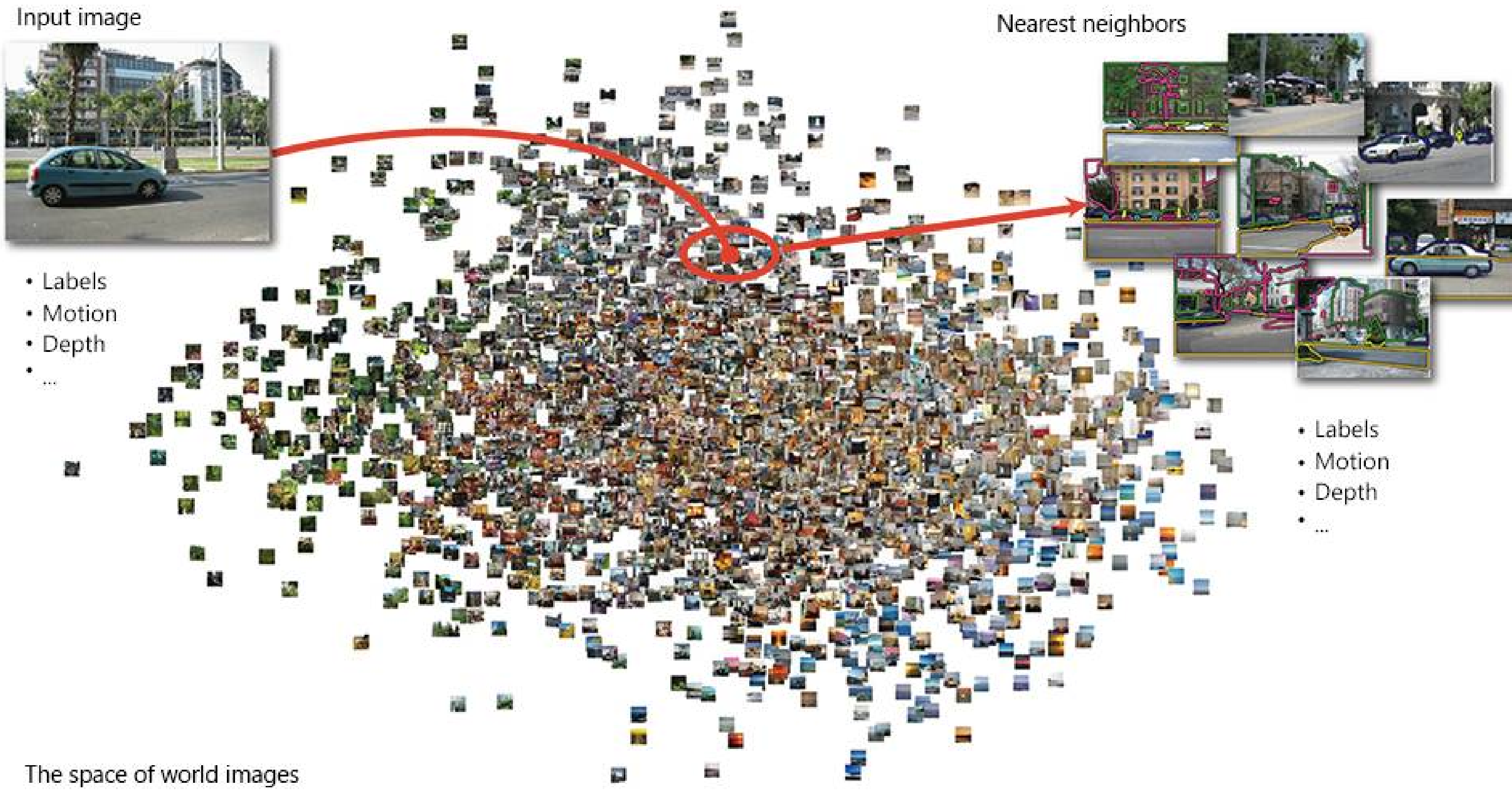
→ Fast

Applications of Dense Correspondences

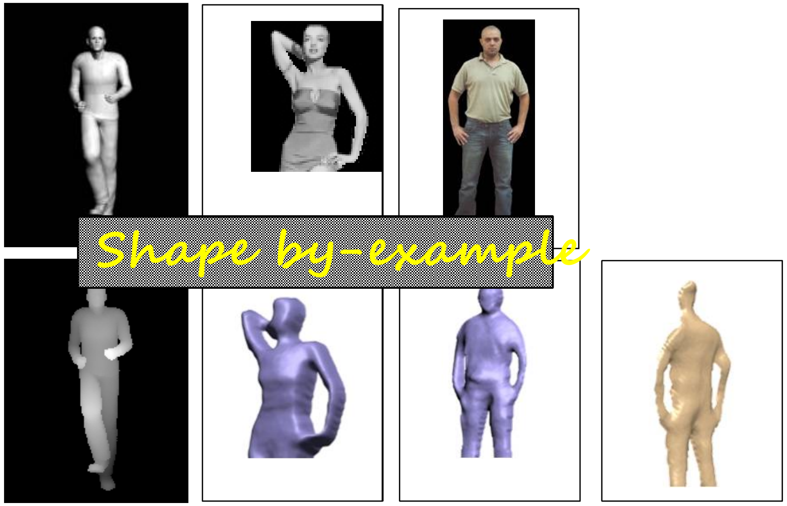
CVPR 2014 Tutorial

Dense Image Correspondences for Computer Vision

Ce Liu[1] Michael Rubinstein[1] Jaechul Kim[2] Zhuowen Tu[3]
[1]Microsoft Research [2]Amazon [3]UCSD



Shape by-example



[Hassner&Basri '06a, '06b,'13]

New view synthesis

[Hassner '13]

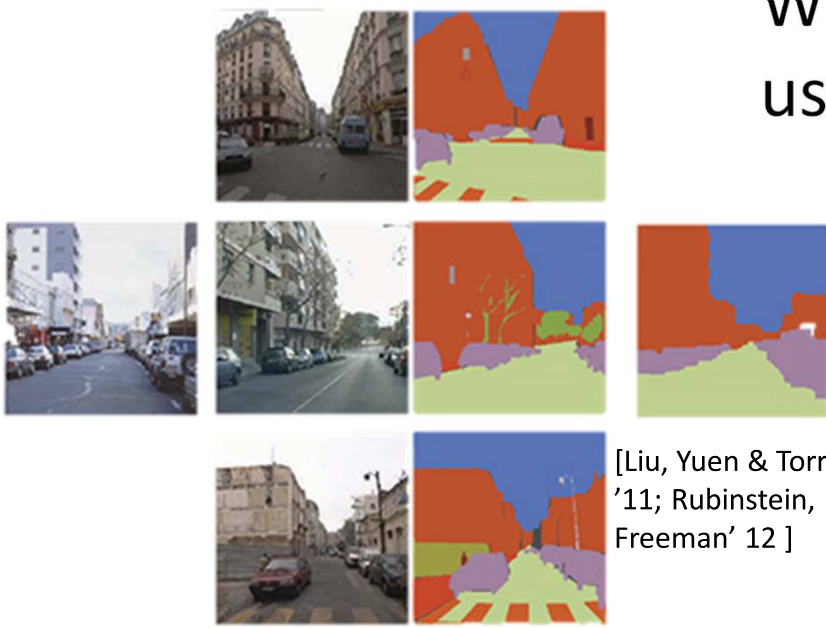
Face recognition

[Liu, Yuen & Torralba '11]

Fingerprint recognition

[Hassner, Saban & Wolf]

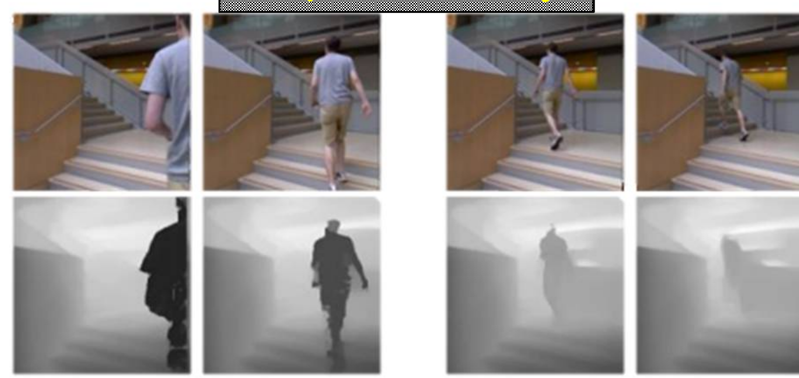
Why is this useful?



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

Label transfer / scene parsing

Depth transfer



Slide courtesy T. Hassner

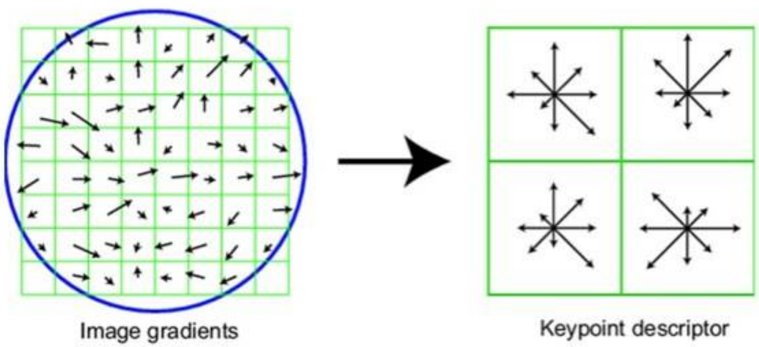
Taxonomy (a matrix form)
Typical MAP setup: Matching evidence term with build-in coherence or smoothness regularization

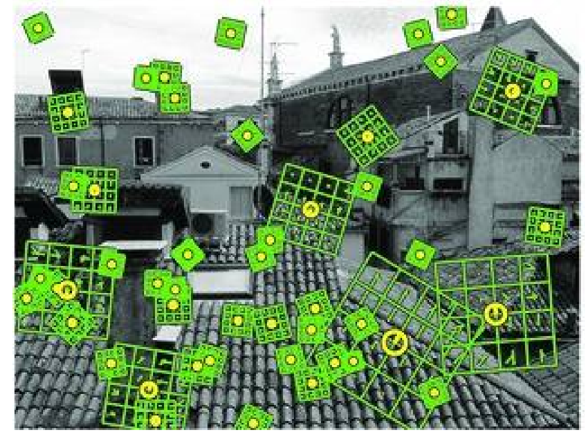
- Matching evidence evaluation
  - General local features
  - Specific tuned features
  - Similarity measures
  - Learned features/measures
- Regularization
  - Local aggregation
  - Non-local/semi-global aggregation or regularization
  - Global discrete/continuous labeling optimization
  - Continuous variational models
  - Non-parametric motion models

What decides the performance of visual correspondence?

1. **How well can we describe input images in a local manner?**

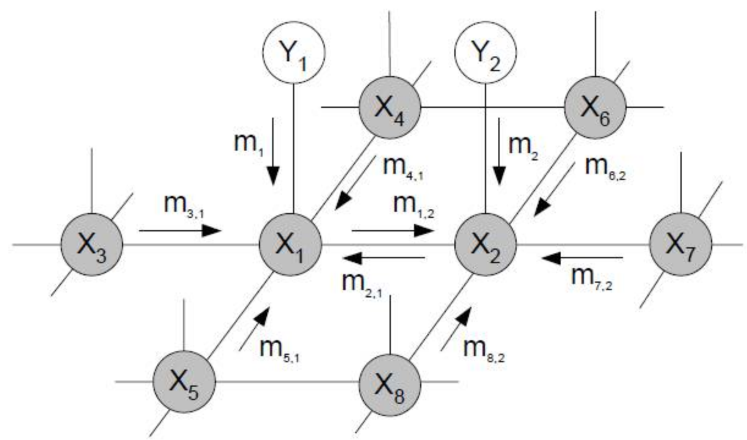
   Ex) SIFT (Scale-invariant feature transform)





2. **How well can we optimize an objective defined for estimating visual correspondence?**

   Ex) Belief Propagation
   (message passing algorithm)



General Formulation

- Find the label l_p for each pixel p, for instance, by minimizing the following objective consisting of the data fidelity E_p and the prior term E_{pq}

$$E = \sum_p E_p(l_p; W) + \sum_p \sum_{q \in \mathcal{N}_p} E_{pq}(l_p, l_q)$$

Evaluating matching evidences with local image descriptors or matching similarity measures

Enforcing the spatial smoothness constraint

General Formulation: Local vs. Global?

- **Local approaches**
  - Using the data fidelity term only
  - Typically, aggregating the data cost with edge-aware filtering

$$E = \sum_p E_p(l_p; W)$$

Cost Volume Filtering, CVPR 2012
PatchMatch Filter, CVPR 2013

- **Global approaches**
  - Using both the data fidelity and prior terms
  - **Optionally**, aggregating the data cost with edge-aware filtering for stronger performance

$$E = \sum_p E_p(l_p; W) + \sum_p \sum_{q \in \mathcal{N}_p} E_{pq}(l_p, l_q)$$

$|W| = 1$, No cost aggregation

Belief Propagation, IJCV 2006
PatchMatch Belief Propagation, IJCV 2014

PART 1: EVALUATING MATCHING EVIDENCES:

LOCAL IMAGE DESCRIPTORS AND MATCHING SIMILARITY MEASURES

Part 1: Evaluating matching evidences: local image descriptors and matching similarity measures

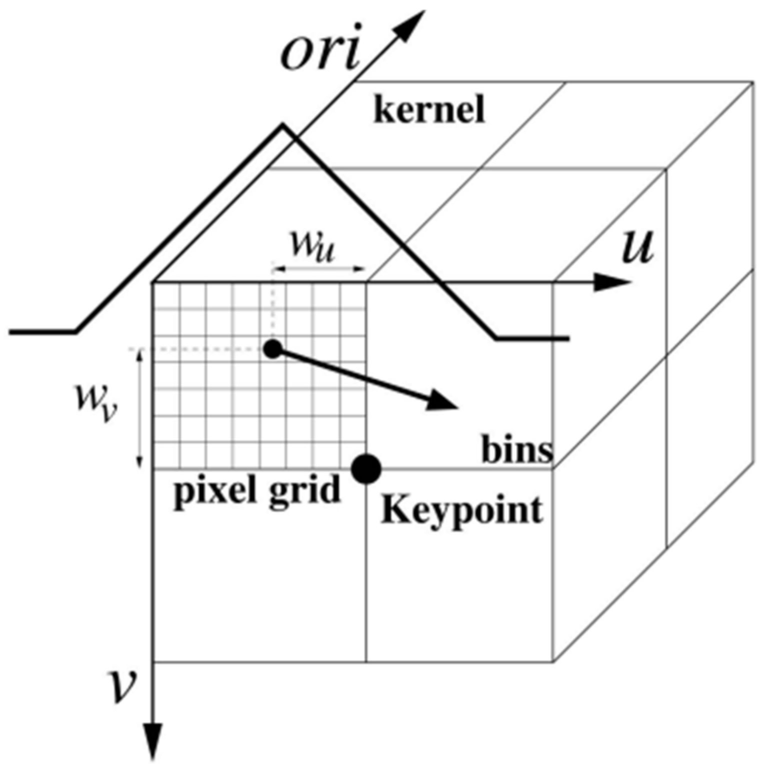
- **Descriptors for matching (sparse) interest points**
  - SIFT [1], BRISK [2], BRIEF [3], Affine SIFT (ASIFT) [4]
- **Descriptors for dense wide-baseline matching**
  - DAISY [5]
- **Descriptors for semi-dense large displacement matching**
  - Deep Matcher [6]
- **Descriptors for matching semantically similar image parts (e.g. cross-domain matching)**
  - Local Self-Similarity (LSS) [7], Locally Adaptive Regression Kernels (LARK) [8]
- **Similarity measures for handling photometric and multi-modal variations**
  - Rank Transform, Census transforms [9], Mutual Information (MI) [10], Normalized Cross-Correlation (NCC) [11], Zero-mean Normalized Cross-Correlation (ZNCC) [12], Dense Adaptive Self-Correlation (DASC) [13]
- *Future work/trend*: Learned matching similarity from CNN models, e.g. [CVPR'15]
  - Computing the Stereo Matching Cost With a Convolutional Neural Network [full paper] [ext. abstract]
    Jure Žbontar, Yann LeCun

Reference - Descriptor

1. D. Lowe, “Distinctive image features from scale-invariant keypoints,” Int. Journal of Computer Vision, 2004.
2. S. Leutenegger, et al., “BRISK: Binary robust invariant scalable keypoints,” ICCV 2011.
3. M. Calonder, et al., “BRIEF: Computing a local binary descriptor very fast,” IEEE Trans. on Pattern Analysis and Machine Intelligence, 2012.
4. J. M. Morel and G. Yu, “ASIFT: A new framework for fully affine invariant image comparison,” SIAM Journal on Imaging Sciences, 2009.
5. E. Tola, V. Lepetit, and P. Fua, “DAISY: An efficient dense descriptor applied to wide-baseline stereo,” IEEE Trans. Pattern Analysis and Machine Intelligence, 2010.
6. P. Weinzaepfel, J. Revaud, Z Harchaoui, and C. Schmid, “DeepFlow: Large displacement optical flow with deep matching,” ICCV 2013.
7. E. Schechtman and M. Irani, “Matching local self-similarities across images and videos,” CVPR 2007.
8. H. Takeda, S. Farsiu, and P. Milanfar, “Kernel regression for image processing and reconstruction,” IEEE Trans. on Image Processing, 2007.
9. R. Zabih and J. Woodfill, “Non-parametric local transforms for computing visual correspondence,” ECCV 1994.
10. H. Hirschmuller, “Stereo processing by semi-global matching and mutual information,” IEEE Trans. on Pattern Analysis and Machine Intelligence, 2008.
11. Y. S. Heo, K. M. Lee, and S. U. Lee, “Robust stereo matching using adaptive normalized cross-correlation,” IEEE Trans. on Pattern Analysis and Machine Intelligence, 2011.
12. X. Shen, L. Xu, Q. Zhang, and J. Jia, “Multi-modal and multi-spectral registration for natural images,” ECCV 2014.
13. S. Kim, D. Min, B. Ham, S. Ryu, M. N. Do, and K. Sohn, “DASC: Dense Adaptive Self-Correlation Descriptor for Multi-modal and Multi-spectral Correspondence,” CVPR 2015.
14. H. Hirschmuller and D. Scharstein, “Evaluation of stereo matching costs on images with radiometric differences,” IEEE Trans. on Pattern Analysis and Machine Intelligence, 2009.
15. C. Vogel, S. Roth, and K. Schindler, “An Evaluation of Data Costs for Optical Flow,” GCPR 2013.

SIFT Descriptor: matching sparse points

- SIFT descriptor: 3-D histogram
  - 2D spatial dimension + 1D dimension (image gradient direction)
  - Each bin contains a **weighted sum of the norms of image gradients**.



SIFT Descriptor: matching sparse points

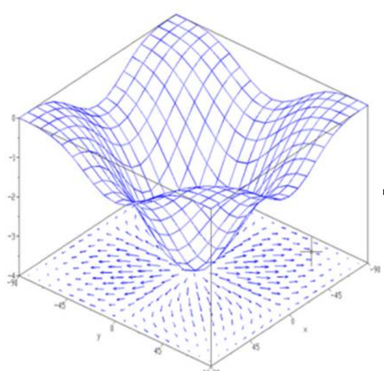
- SIFT (Scale Invariant Feature Transform) [Lowe'2004'IJCV]
  - **The most popular descriptor thanks to distinctiveness and invariance to a variety of common image deformation.**

  - **Step 1. Image Gradient Magnitude & Orientation Computation**

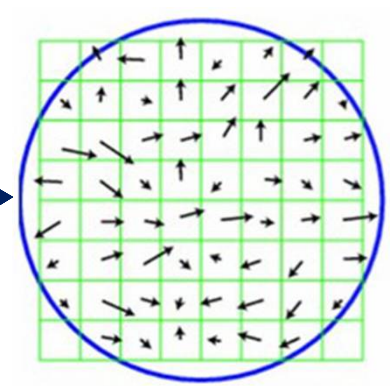
$$\boldsymbol{m(x,y)} = \sqrt{(L(x+1,y)-L(x-1,y))^2 + (L(x,y+1)-L(x,y-1))^2}$$

$$\boldsymbol{\theta(x,y)} = tan^{-1}((L(x,y+1)-L(x,y-1))/(L(x+1,y)-L(x-1,y)))$$

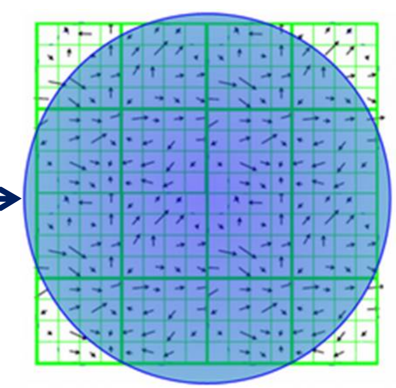
  - **Step 2. Gaussian Weighting Function**
    - **To avoid sudden changes in the descriptor with small changes.**



Gradient computation



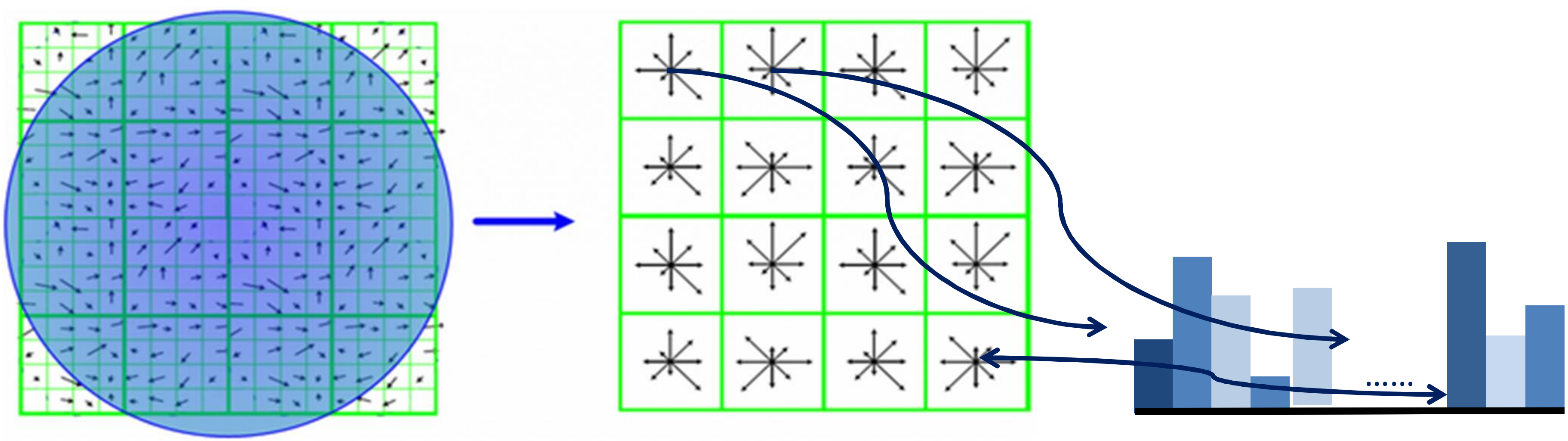
Gradient distribution



Gaussian weighting function

SIFT Descriptor: matching sparse points

- SIFT (Scale Invariant Feature Transform) [Lowe'2004'IJCV]
  - Step 3. Gradient Orientation Histogram
    - Divides its neighborhood (e.g. 16×16) into a 4×4 cell array, quantizes the orientation into 8 bins in each cell, and obtains a $4 \times 4 \times 8 = 128$ dimensional vector as the SIFT for a pixel.
  - Step 4. Feature Vector Normalization
    - To reduce the effects of illumination change.



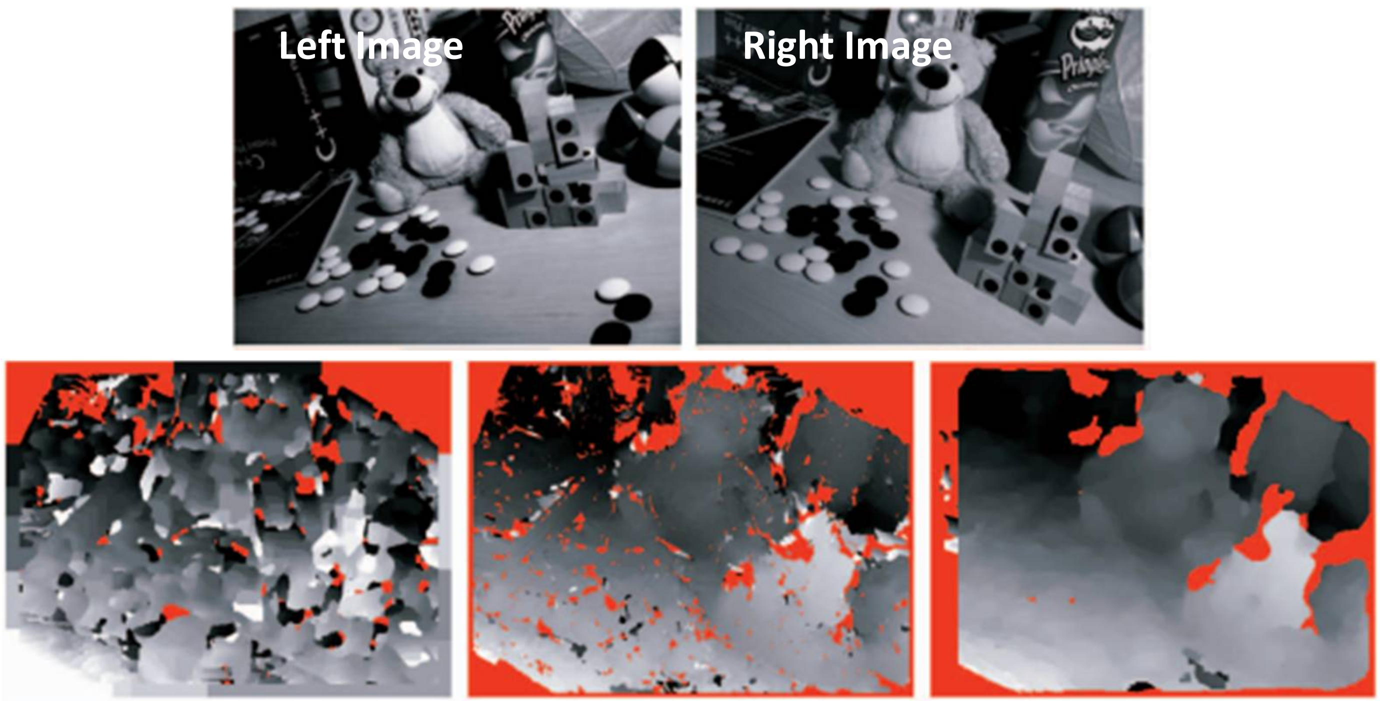
Gaussian Weighting Gradients Distribution

Gradient Orientation Grid-Square Binning

SIFT Feature Descriptor

DAISY Descriptor

- DAISY [Tola’2010’TPAMI]
  - SIFT works well for **sparse** wide-baseline matching, but it is very **SLOW** for **dense** matching tasks.
  - DAISY retains the **robustness** of SIFT and be computed **efficiently**.



NCC results **SIFT results** **DAISY results**

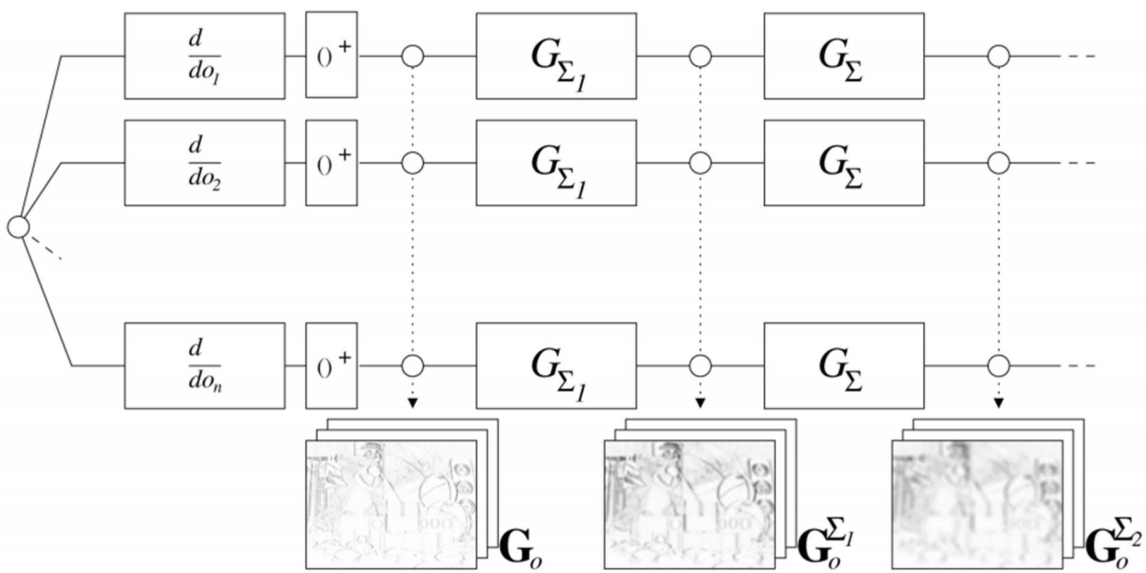
NCC: Normalized Cross Correlation

DAISY Descriptor

- Gaussian convolved orientation maps

$$G_o{}^{\Sigma} = G_{\Sigma} * (\partial I/\partial o)^{+}$$

  - G_{Σ}: Gaussian convolution filter with variance Σ
  - $\partial I/\partial o$: image gradient in direction o.



DAISY Descriptor

Step 1. Compute histograms for each pixel

$$h_{\Sigma}(u,v) = [{G_1}^{\Sigma}(u,v), {G_2}^{\Sigma}(u,v), \dots, {G_8}^{\Sigma}(u,v)]^T$$

$h_{\Sigma}(u,v)$: histogram at (u,v)

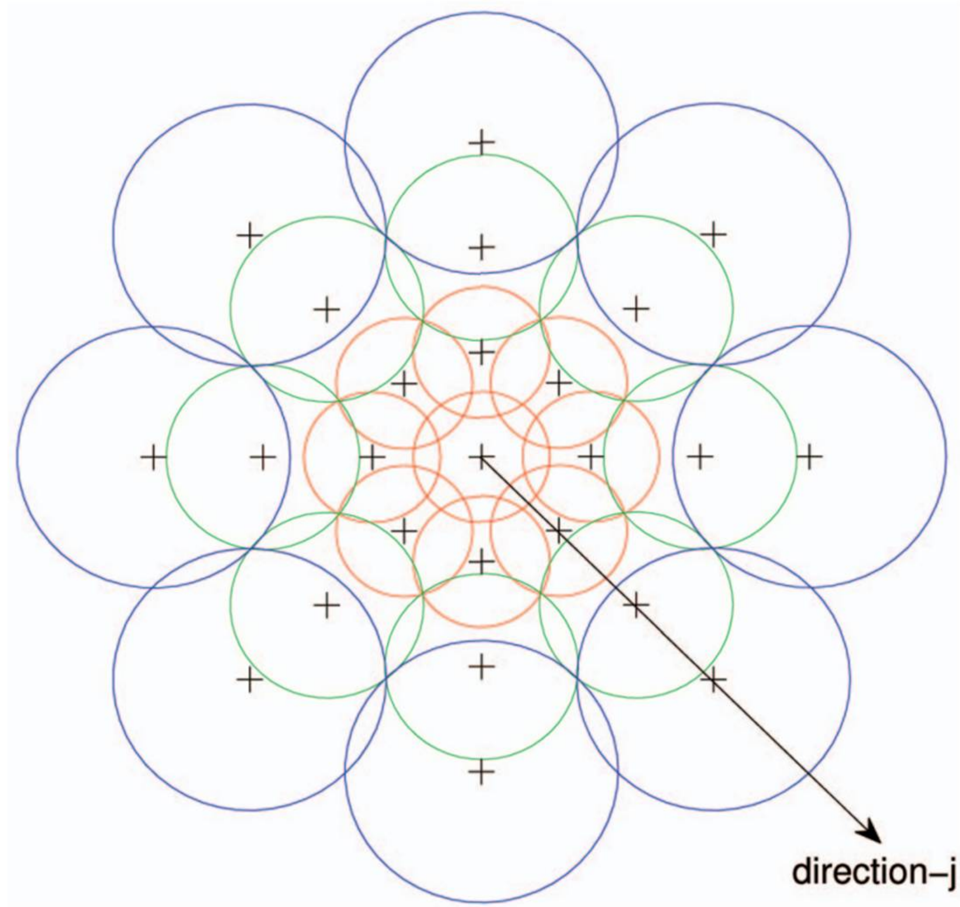
${G_1}^{\Sigma}(u,v)$: Gaussian convolved orientation maps

Step 2. Normalize histograms to unit norm

Step 3. DAISY descriptor is computed as

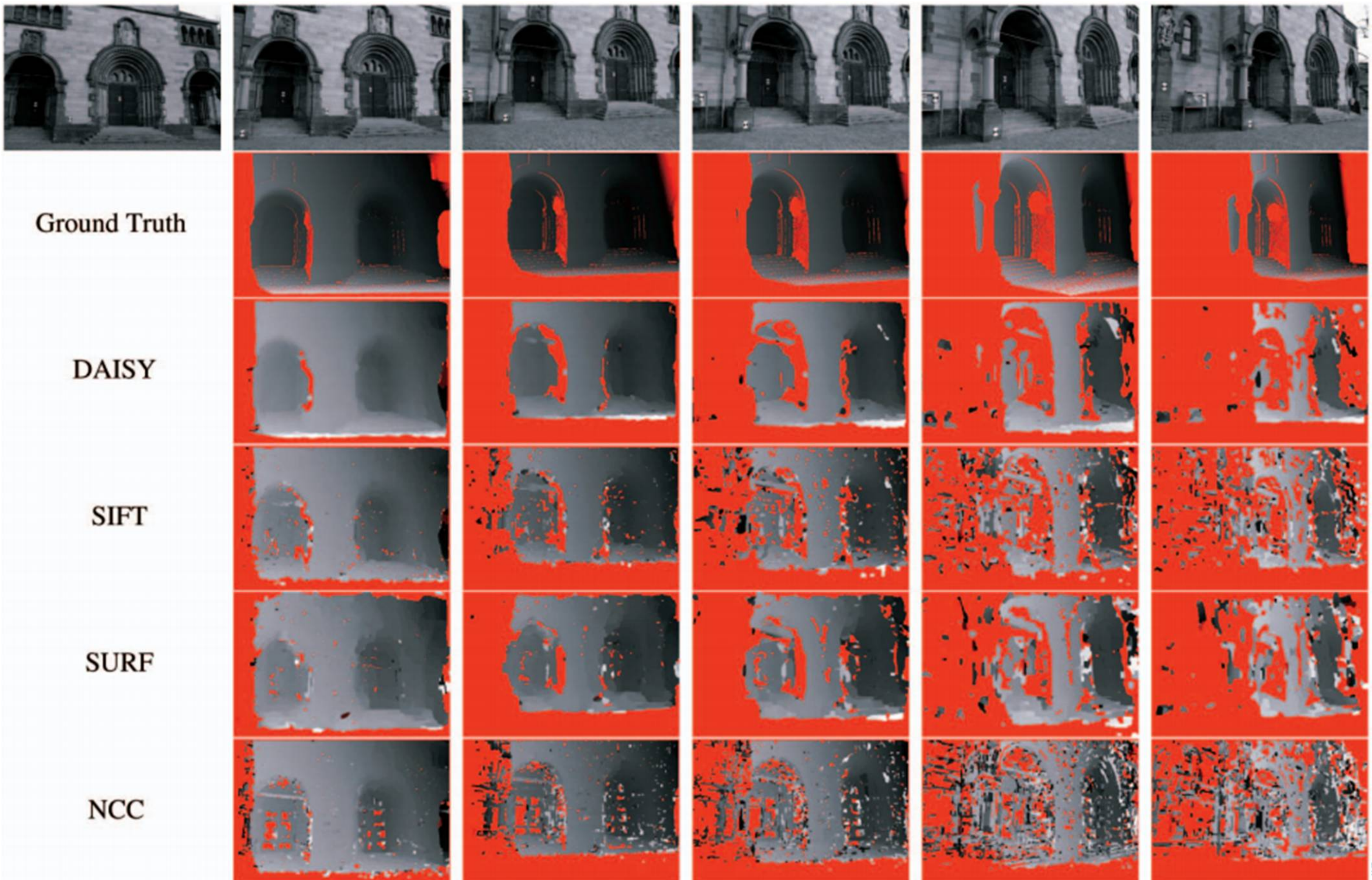
$$D(u_0,v_0) = \begin{matrix} [h_{\Sigma_1}(u,v), \\ h_{\Sigma_1}(I_1(u,v)), \dots, h_{\Sigma_1}(I_N(u,v)), \\ h_{\Sigma_2}(I_1(u,v)), \dots, h_{\Sigma_2}(I_N(u,v)), \\ h_{\Sigma_3}(I_1(u,v)), \dots, h_{\Sigma_3}(I_N(u,v))] \end{matrix}^T$$

DAISY Descriptor



DAISY Feature Descriptor

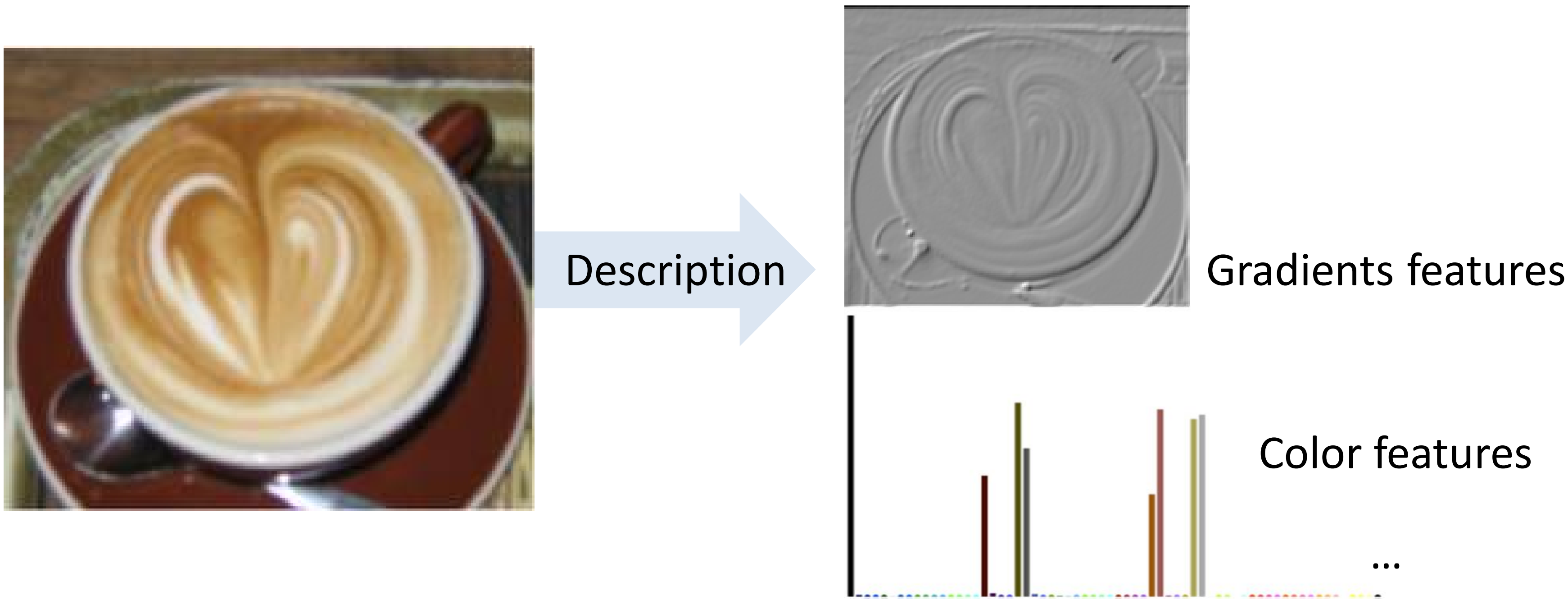
SIFT & SURF & DAISY Comparison



PART 1.2: LOCAL SELF-SIMILARITY

Conventional Image Descriptors

- **Measuring image properties from images.**
  - Gradient, edge, or spatial structures



Does It describes underlying visual Property?

Conventional Descriptors vs. Self-Similarity

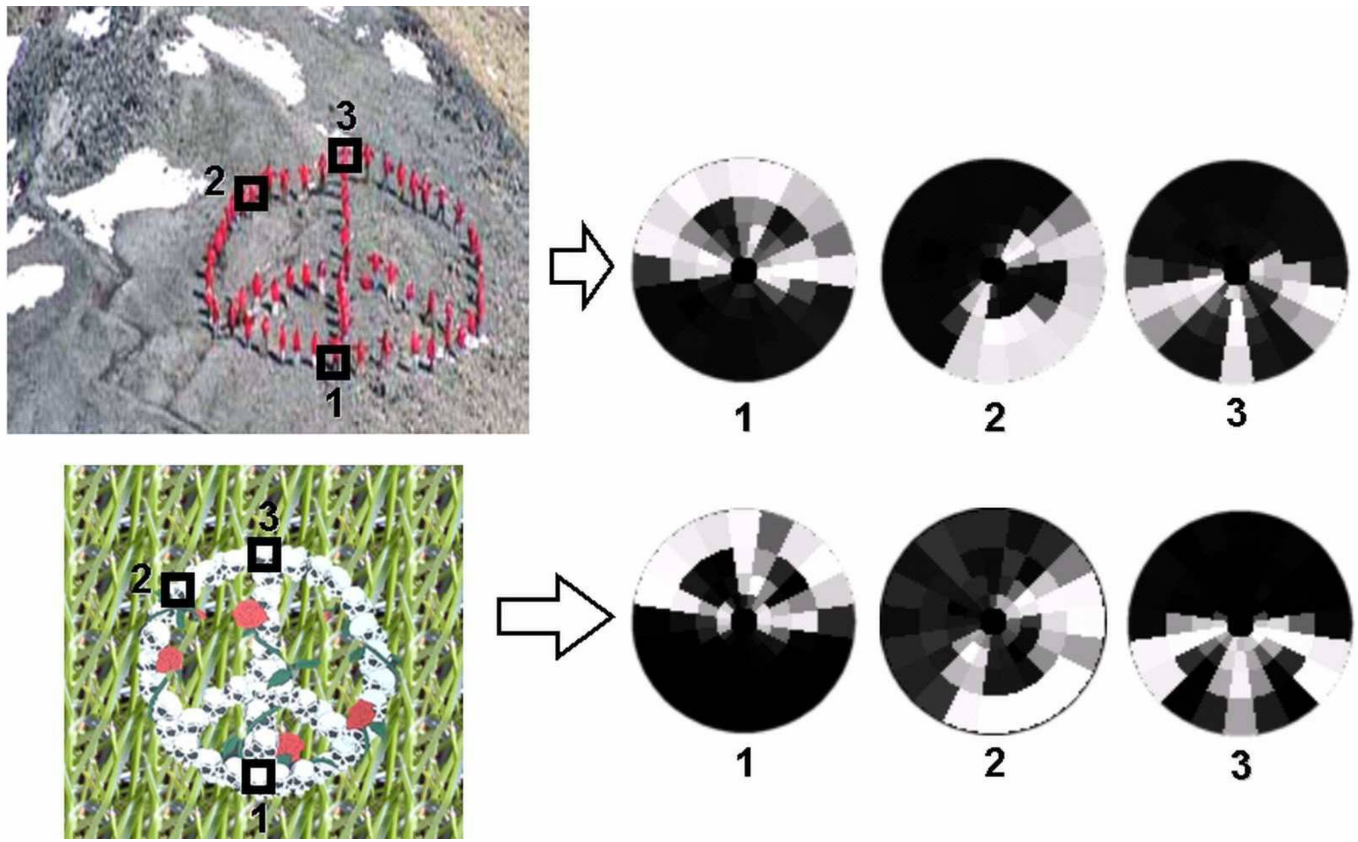
- **Conventional Descriptors**
  - **Direct visual properties** shared by images of the same class (e.g. colors, gradients,…)

- **Self-Similarity**
  - **Indirect property**: Geometric layout of repeated patches within an image

Do Not share common image properties (colors, textures, edges), but Do share a similar geometric layout of local internal self-similarities.

Local Self-Similarity (LSS) Descriptor

- Explore **local internal layouts of self-similarities**



Local Self-Similarity (LSS) Descriptor

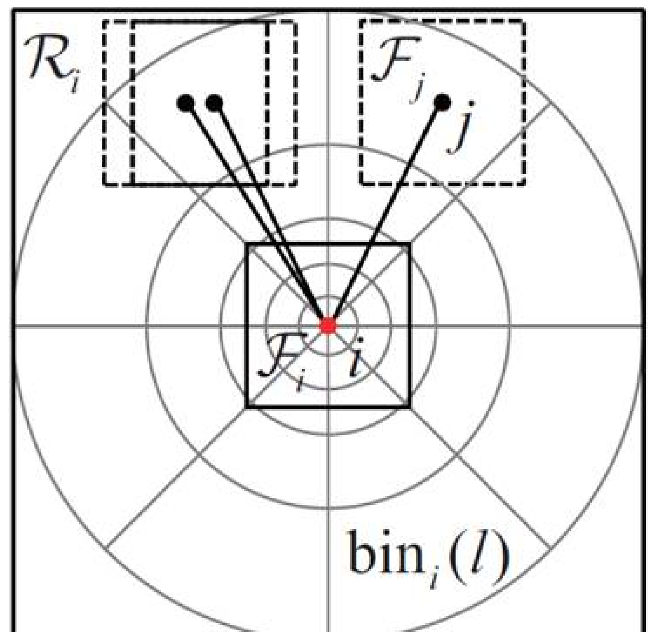
- **Step 1: Compute self-similarity on correlation surface**
  - Determine $N \times N$ correlation surface $C(i, j)$

$$\mathcal{C}(i,j) = \exp\left(-\mathrm{SSD}(\mathcal{F}_i, \mathcal{F}_j)/\sigma_s\right)$$

- **Step 2: Transform into log-polar coordinates, and select the maximal correlation value in each bin**

$$d_{i,l}^{\mathrm{LSS}} = \max_{j \in \mathrm{bin}_i(l)} \{\mathcal{C}(i,j)\}$$

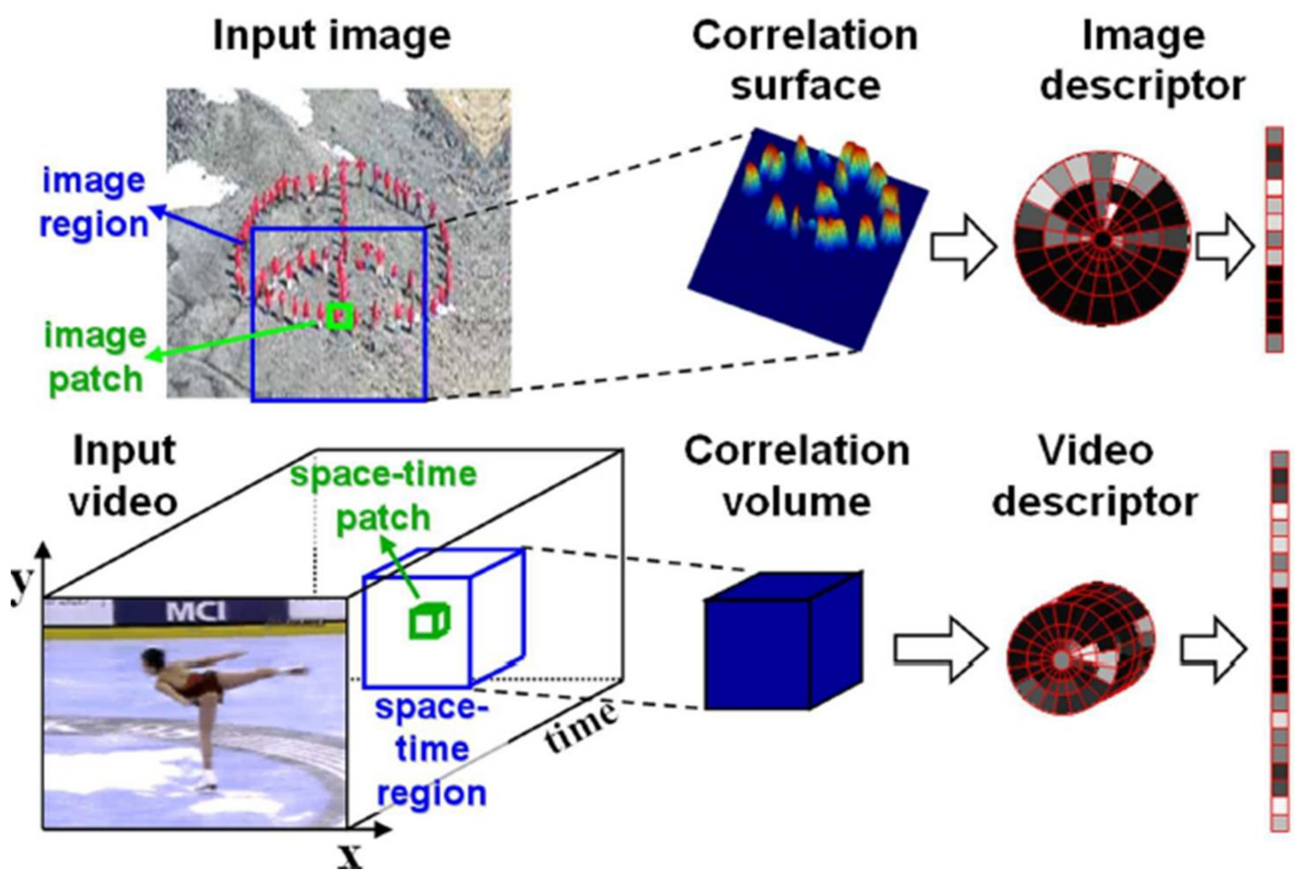
This descriptor vector is normalized by linearly stretching its values to the range [0..1]



Local Self-Similarity (LSS) Descriptor

Step 1: Compute correlation surface.

Step 2: Transform into log-polar coordinates, and select the maximal correlation value in each bin.



Properties and Benefit of LSS Descriptor

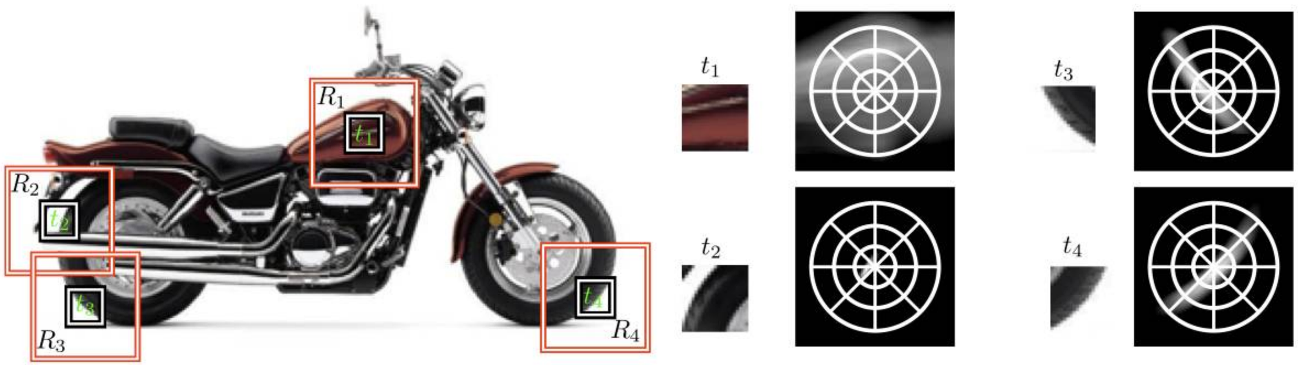
- **Locality**
  - Self-similarities are treated as a local image property, and are accordingly measured locally (within a surrounding image region).

- **Robust to Affine Deformation**
  - The log-polar representation accounts for local affine deformation in the self-similarities.

- **Robust to Non-Rigid Deformation**
  - Insensitive to the exact position of the best matching patch within that bin (similar to the observation used for brain signal modelling).

- **Meaningful Image Patterns**
  - The use of patches (at different scales) captures more meaningful image patterns than individual pixels.

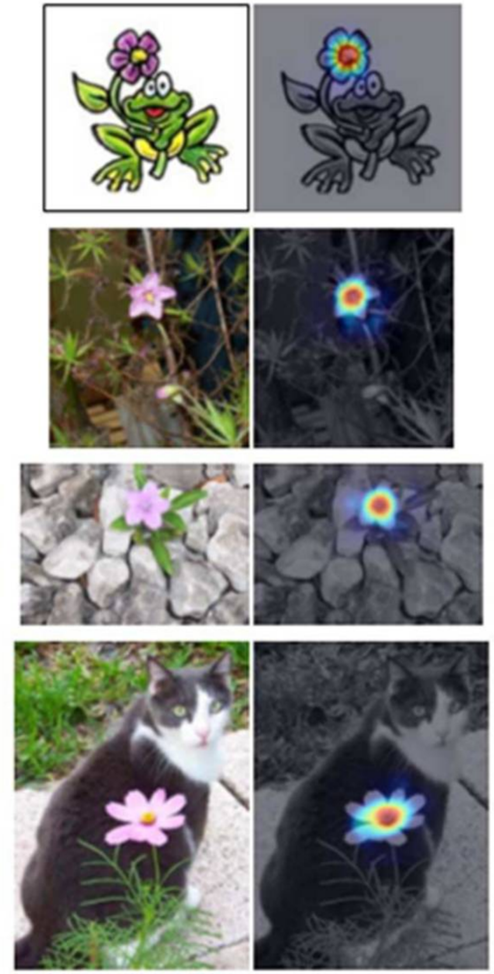
LSS Descriptor Applications



- **Object Recognition, Image Retrieval, Action Recognition**
  - Ensemble matching [Shechtman CVPR 07]
  - Nearest neighbor matching [Boiman CVPR 08]
  - Bag of Local Self-Similarities [Gehler ICCV09, Vedaldi ICCV09, Horster ACMM08, Lampert CVPR09, Chatfield ICCV09]
    1. Compute LSS descriptors for an image.
    2. Assign the LSS descriptors to a codebook.
    3. Represent the image as a histogram of LSS descriptors.

Interest Object Detection in Images

Single template image



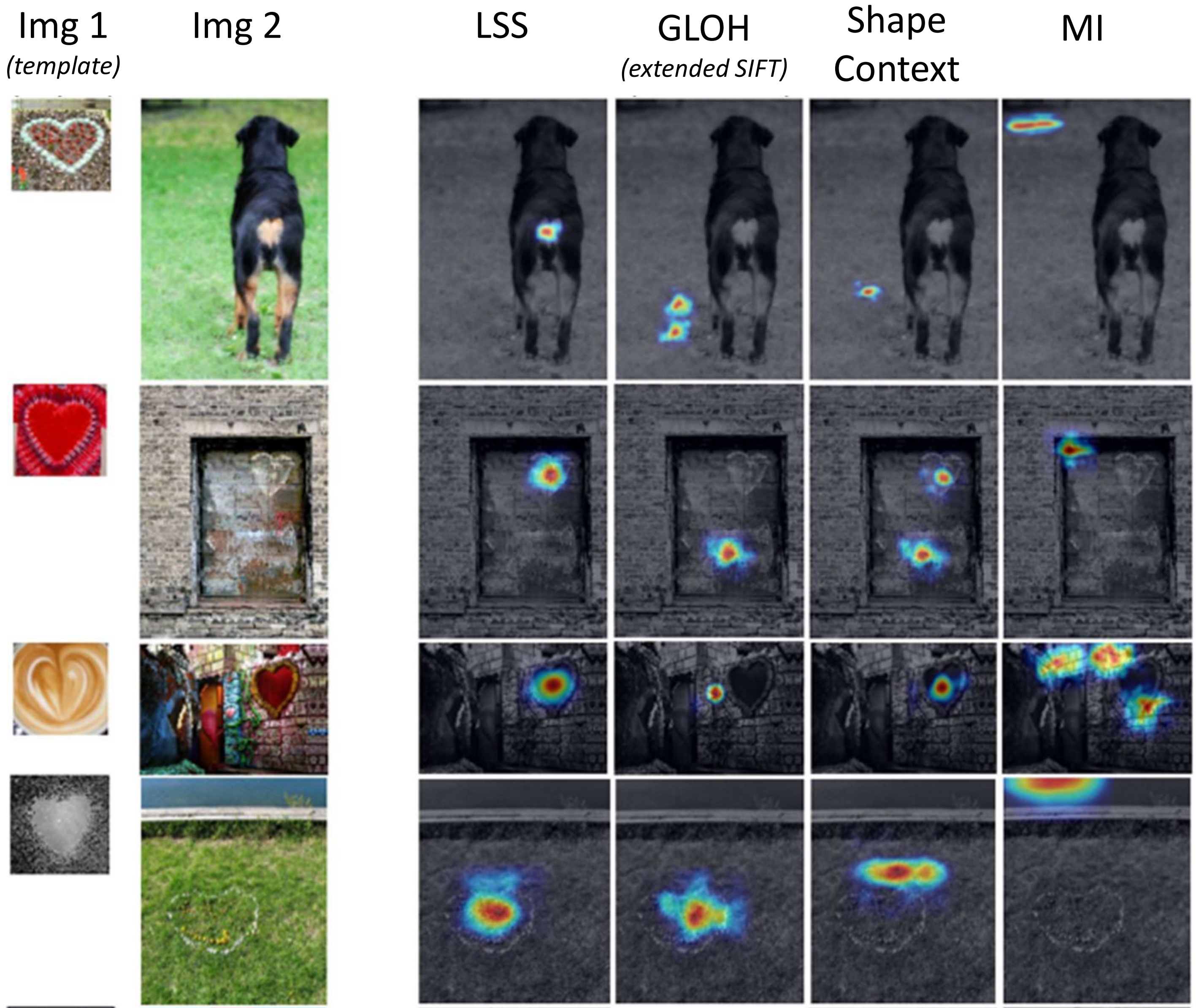
The images against which it was compared with the corresponding detections.

Image Retrieval by “Sketching”

Hand-sketched template

The images against which it was compared with the corresponding detections.

Comparison to Other Descriptors



EXTENSION OF LSS DESCRIPTOR TO MULTI-MODAL MATCHING

Can we find correspondences in the images below?

Yes! It is possible using our new descriptor (DASC).

DASC: Dense Adaptive Self-Correlation Descriptor for Multi-modal and Multi-spectral Correspondence, CVPR 2015

- RGB-NIR, Radiometric distortion, Motion Blur

Image Descriptor Matters!

- DASC: Dense Adaptive Self-Correlation Descriptor for Multi-modal and Multi-spectral Correspondence, CVPR 2015

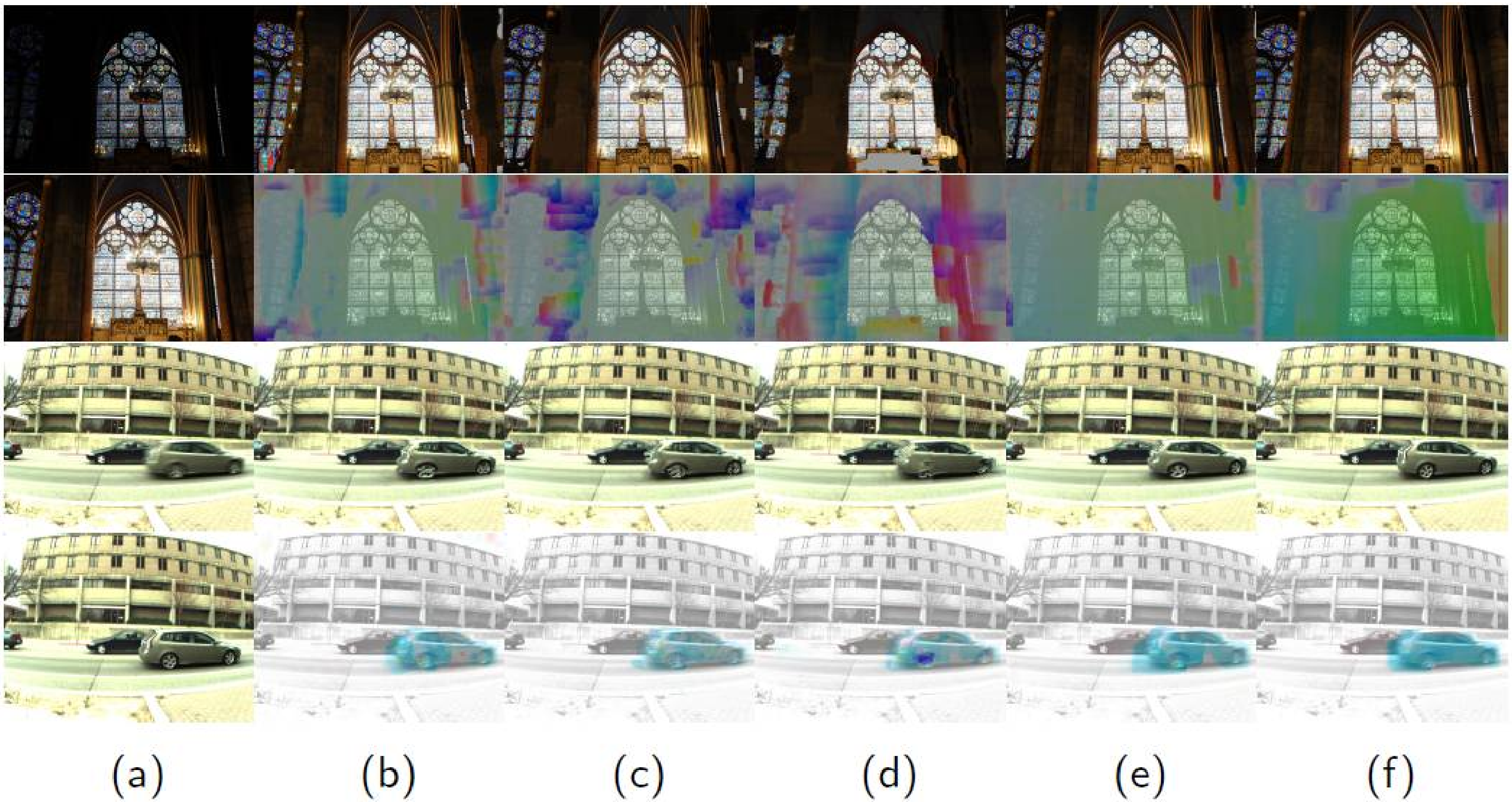


Figure: Comparison of dense correspondence for different exposure images and blurred-sharpen images for (a) input image pairs, (b) RSNCC, (c) BRIEF, (d) DAISY, (e) LSS, (f) DASC. The results consist of warped color images and 2-D flow fields.

Our Goal

Our Goal

1) **Addressing photometric distortions** in multi-modal and multi-spectral images
2) The descriptor should be **dense**, and be computed very **efficiently**

Contribution

1. **A patch-wise receptive field pooling** with sampling patterns optimized via a discriminative learning.
2. An efficient scheme using edge-aware filtering (EAF) to compute dense descriptors for all pixels
3. An intensive comparative study with state-of-the-art methods using various datasets.

Problem of Existing Descriptors

- Challenging limitations in multi-modal and multi-spectral images
  - **Nonlinear photometric deformation** even within a small window, e.g., gradient reverses and intensity variation.
  - **Outliers** including structure divergence caused by shadow or highlight.

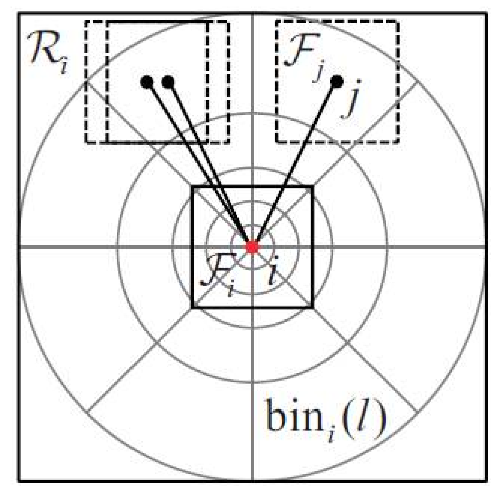
➡ **Most of the existing descriptors** may fail to compute a reliable descriptor in the images below.

Existing Work: Local Self-Similarity (LSS) Descriptor

- The local self-similarity (LSS) [1] may be useful in overcoming many inherent limitations of existing descriptors in establishing correspondence between multi-modal images.

- An input image $f_i : \mathcal{I} \to \mathbb{R}$ or $\mathbb{R}^3$, a dense descriptor $\mathcal{D}_i : \mathcal{I} \to \mathbb{R}^L$ is defined on a local support window centered at each pixel i

Key idea: The **local internal layout of self-similarities** is less sensitive to photometric distortions



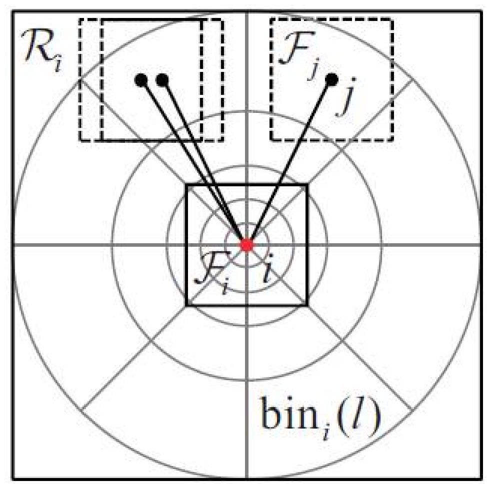
[1] E. Schechtman and M. Irani. Matching local self-similarities across images and videos, CVPR, 2007.

Existing Work: Local Self-Similarity (LSS) Descriptor

Formally, $\mathcal{D}_i^{\mathrm{LSS}} = \bigcup_l d_{i,l}^{\mathrm{LSS}}$ for $l = 1, ..., L^{\mathrm{LSS}}$ is a $L^{\mathrm{LSS}} \times 1$ feature vector, and can be written as follows:

$$\boxed{d_{i,l}^{\mathrm{LSS}} = \max_{j \in \mathrm{bin}_i(l)} \{\mathcal{C}(i,j)\},} \qquad \mathcal{C}(i,j) = \exp\left(-\mathrm{SSD}(\mathcal{F}_i, \mathcal{F}_j)/\sigma_s\right)$$

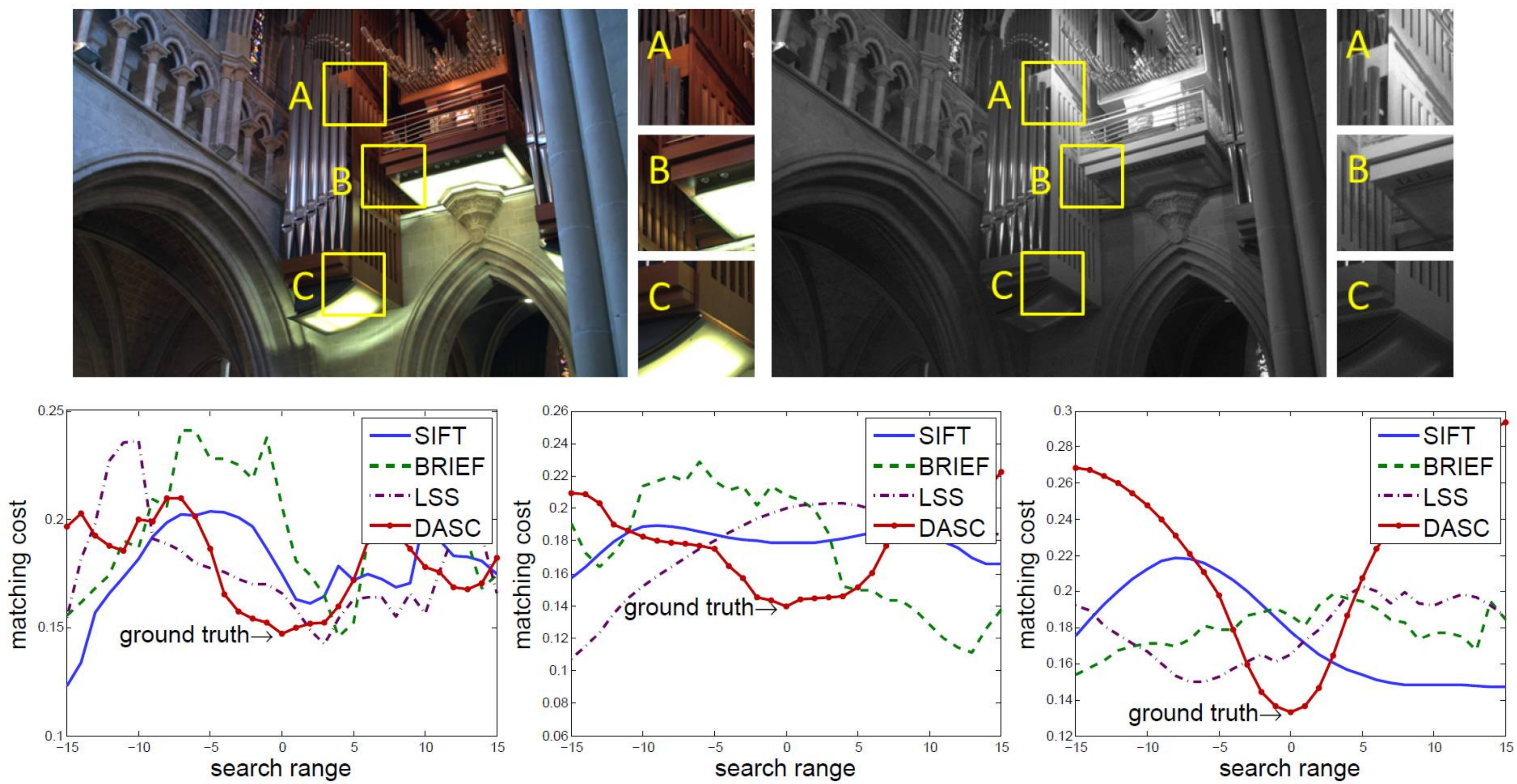
where $\mathrm{bin}_i(l) = \{j | j \in \mathcal{R}_i, \rho_{r-1} < |i - j| \leq \rho_r, \theta_{a-1} < \angle(i-j) \leq \theta_a\}$.



[1] E. Schechtman and M. Irani. Matching local self-similarities across images and videos, CVPR, 2007.

Problem of Existing Descriptors (including LSS)

However, even LSS often produces inaccurate correspondence.



(a) Matching cost in A (b) Matching cost in B (c) Matching cost in C

Figure: Examples of matching cost comparison. Multi-spectral RGB and NIR images have locally non-linear deformation as depicted in A, B, and C.

Dense Adaptive Self-Correlation (DASC)

Limitation of the LSS descriptor

1) The **center-biased max pooling** is very sensitive to the degradation of a center patch.

2) **No efficient computational scheme** designed for computing dense descriptor

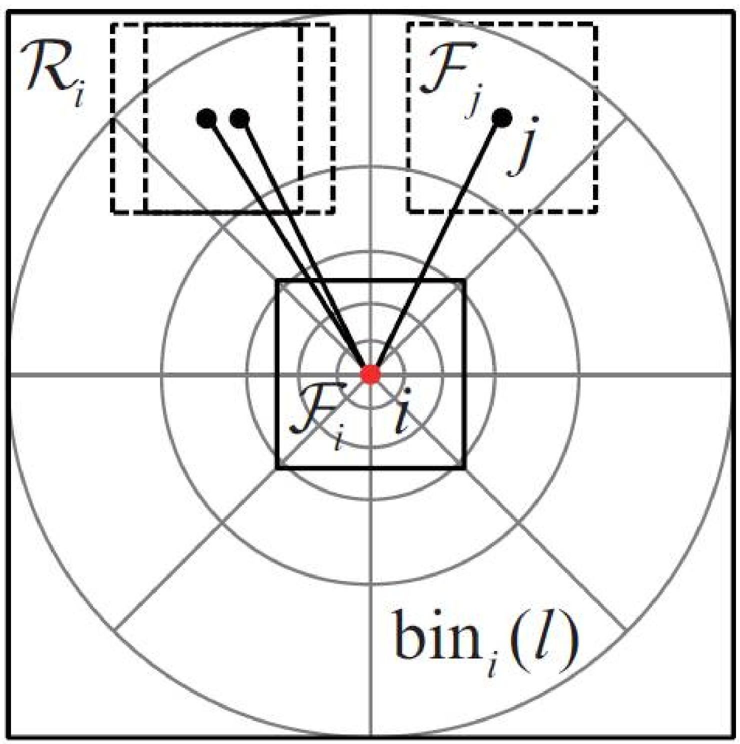
Intuitions for the DASC Descriptor

1) There frequently exist **non-informative regions** which are locally degraded, e.g., shadows or outliers.

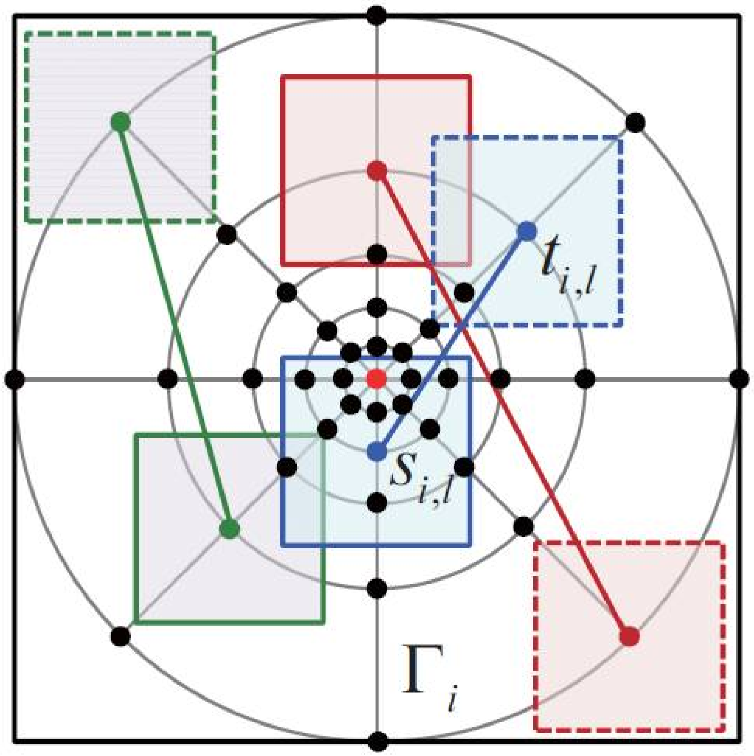
2) The **randomness** enables a descriptor to encode structural information more robustly.

LSS vs. DASC

- Center-biased dense <u>max</u> pooling vs. Randomized pooling
  – Note that the DASC descriptor does NOT use the max operation.
  – The max operation may lead to wrong localization!



(a) LSS descriptor



(b) DASC descriptor

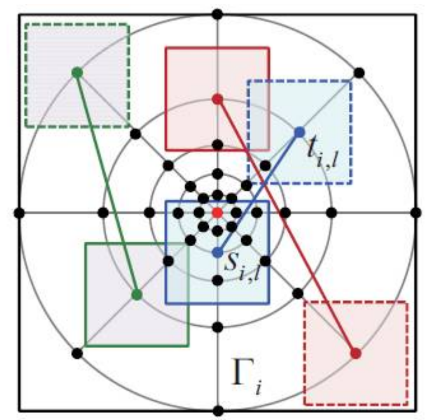
Randomized Receptive Field Pooling

- Using **all** sampling patterns does **NOT** always produce the best results

➡ Let's select a <u>subset</u> of sampling patterns **randomly**

➡ What about **learning** this sampling patters?

Ex) $\mathbf{41}$ points
→ # of possible sampling pattern: $\mathbf{41 \times 40/2}$
→ Let's just select $\mathbf{3}$ sampling patterns randomly.



Randomized Receptive Field Pooling

- Let $\Gamma_i = \{j | j \in \mathcal{R}_i, |i-j| = \rho_r, \angle(i-j) = \theta_a\}$.
- Our DASC descriptor $\mathcal{D}_i = \bigcup_l d_{i,l}$ for $l = 1, \ldots, L$ is encoded with a set of patch similarity between two patches based on sampling patterns that are randomly selected from Γ_i:

$$d_{i,l} = \mathcal{C}(s_{i,l}, t_{i,l}), \quad s_{i,l}, t_{i,l} \in \Gamma_i,$$

where s_l and t_l are l^{th} randomly selected sampling patterns.

Randomized Receptive Field Pooling

- Sampling Pattern Learning
  - **Key idea**: Learn the sampling pattern using training pairs

From a large number of randomly generated pairs from Γ_i, our goal is to select the best sampling patterns.

- First, the feature $r_m = \bigcup_l r_{m,l}$ that describes two support window pairs $\mathcal{R}_m^1$ and $\mathcal{R}_m^2$ is defined

$$r_{m,l} = \exp\left(-(d_{m,l}^1 - d_{m,l}^2)^2 / 2\sigma_r^2\right).$$

- The decision function to classify the training data set $\mathcal{P}$ as

$$\rho(r_m) = v^T r_m + b,$$

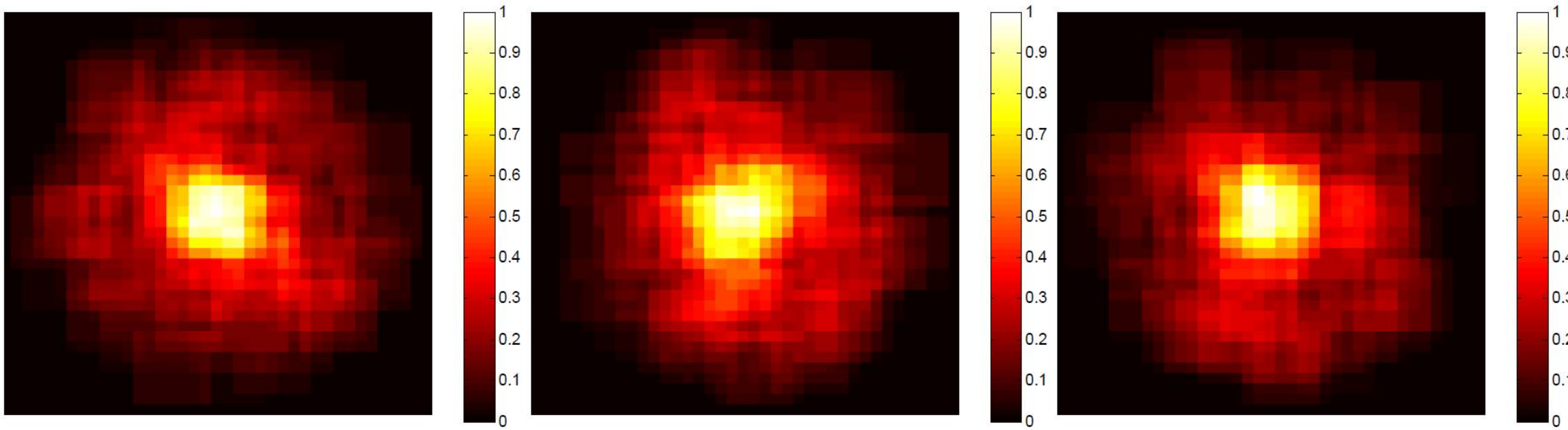
An amount of contribution of each candidate sampling pattern

where weight v indicates an a weight of sampling pattern.

- We use LIBSVM[2] to learn the weight function.

Randomized Receptive Field Pooling

- **Sampling Pattern Learning**
  - The training data-set was built from images taken under varying illumination conditions and/or imaging devices



(a) (b) (c)

Figure: Visualization of patch-wise receptive fields of the DASC descriptor which are learned from (a) Middlebury benchmark, (b) multi-spectral and multi-modal benchmark, and (c) MPI SINTEL benchmark.

The DASC Descriptor Formulation

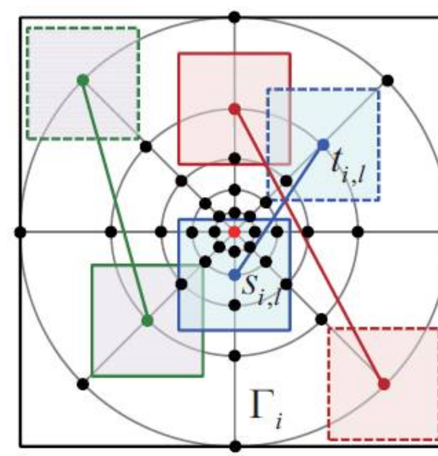
- With the sampling patterns learned, our next job is to compute the self-similarity between two patches

- Adaptive Self-Correlation (ASC) Measure
  - For given two patches $\mathcal{F}_s$ and $\mathcal{F}_t$, the patch-wise similarity is measured using a truncated robust function

$$\mathcal{C}(s,t) = \max(\exp(-(1-|\Psi(s,t)|)/\sigma),\tau)$$

  - For $(s,t) \in \cup_i^L$, we measure the Adaptive Self-Correlation (**ASC**)

$$\Psi(s,t) = \frac{\sum_{s',t'} \omega_{s,s'}\omega_{t,t'}(f_{s'}-\mathcal{G}_s)(f_{t'}-\mathcal{G}_t)}{\sqrt{\sum_{s'}\{\omega_{s,s'}(f_{s'}-\mathcal{G}_s)\}^2}\sqrt{\sum_{t'}\{\omega_{t,t'}(f_{t'}-\mathcal{G}_t)\}^2}}$$



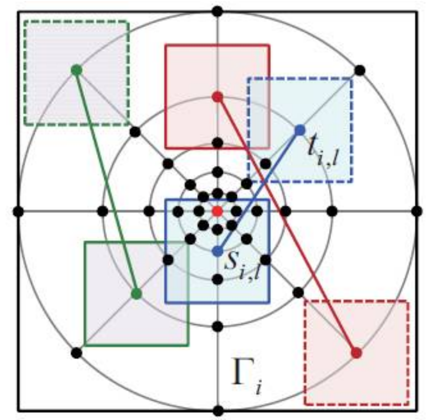
We wish to compute the descriptor densely!

- **Straightforward computation** of the ASC for the selected sampling patterns of all pixels is extremely time-consuming.

$$O(INL)$$

I: Image size, N: Patch size
L: the number of sampling patterns

$$\Psi(s,t)=\frac{\sum_{s',t'}\omega_{s,s'}\omega_{t,t'}(f_{s'}-\mathcal{G}_s)(f_{t'}-\mathcal{G}_t)}{\sqrt{\sum_{s'}\{\omega_{s,s'}(f_{s'}-\mathcal{G}_s)\}^2}\sqrt{\sum_{t'}\{\omega_{t,t'}(f_{t'}-\mathcal{G}_t)\}^2}}$$



Observation: There are **computational redundancies** in the equation above when executing this for all pixels.

Our Solution: Let's employ the **constant-time edge-aware filter (EAF)** to reduce the redundancies

Efficient Computation of DASC

One problem is the **symmetric** weight $\omega_{s,s'}\omega_{t,t'}$ varies for each l, and it is 6-D vector, which increases a computational burden needed for employing **constant-time EAFs**.

- In order to make using EAF computationally feasible, we approximate the ASC with an asymmetric weight

$$\Psi(s,t) = \frac{\sum_{s',t'} \omega_{s,s'}\omega_{t,t'}(f_{s'} - \mathcal{G}_s)(f_{t'} - \mathcal{G}_t)}{\sqrt{\sum_{s'} \{\omega_{s,s'}(f_{s'} - \mathcal{G}_s)\}^2}\sqrt{\sum_{t'} \{\omega_{t,t'}(f_{t'} - \mathcal{G}_t)\}^2}}$$

$$\Rightarrow \quad \boxed{\tilde{\Psi}(i,j) = \frac{\sum_{i',j'} \omega_{i,i'}(f_{i'} - \mathcal{G}_i)(f_{j'} - \mathcal{G}_{i,j})}{\sqrt{\sum_{i'} \omega_{i,i'}(f_{i'} - \mathcal{G}_i)^2}\sqrt{\sum_{i',j'} \omega_{i,i'}(f_{j'} - \mathcal{G}_{i,j})^2}}}$$

- The similarity measure above can be computed in O(1) time using e.g., the Guided Filter. But, Other kinds of EAFs can be used as well.

Efficient Computation of Dense Descriptor

I: Image size, N: Patch size
L: the number of sampling patterns

Straightforward computation of ASC for the selected sampling patterns of all pixels

$$\Psi(s,t) = \frac{\sum_{s',t'} \omega_{s,s'}\omega_{t,t'}(f_{s'} - \mathcal{G}_s)(f_{t'} - \mathcal{G}_t)}{\sqrt{\sum_{s'} \{\omega_{s,s'}(f_{s'} - \mathcal{G}_s)\}^2} \sqrt{\sum_{t'} \{\omega_{t,t'}(f_{t'} - \mathcal{G}_t)\}^2}}$$

$$O(INL)$$

Efficient computation of approximated ASC for the selected sampling patterns of all pixels using **EAF**

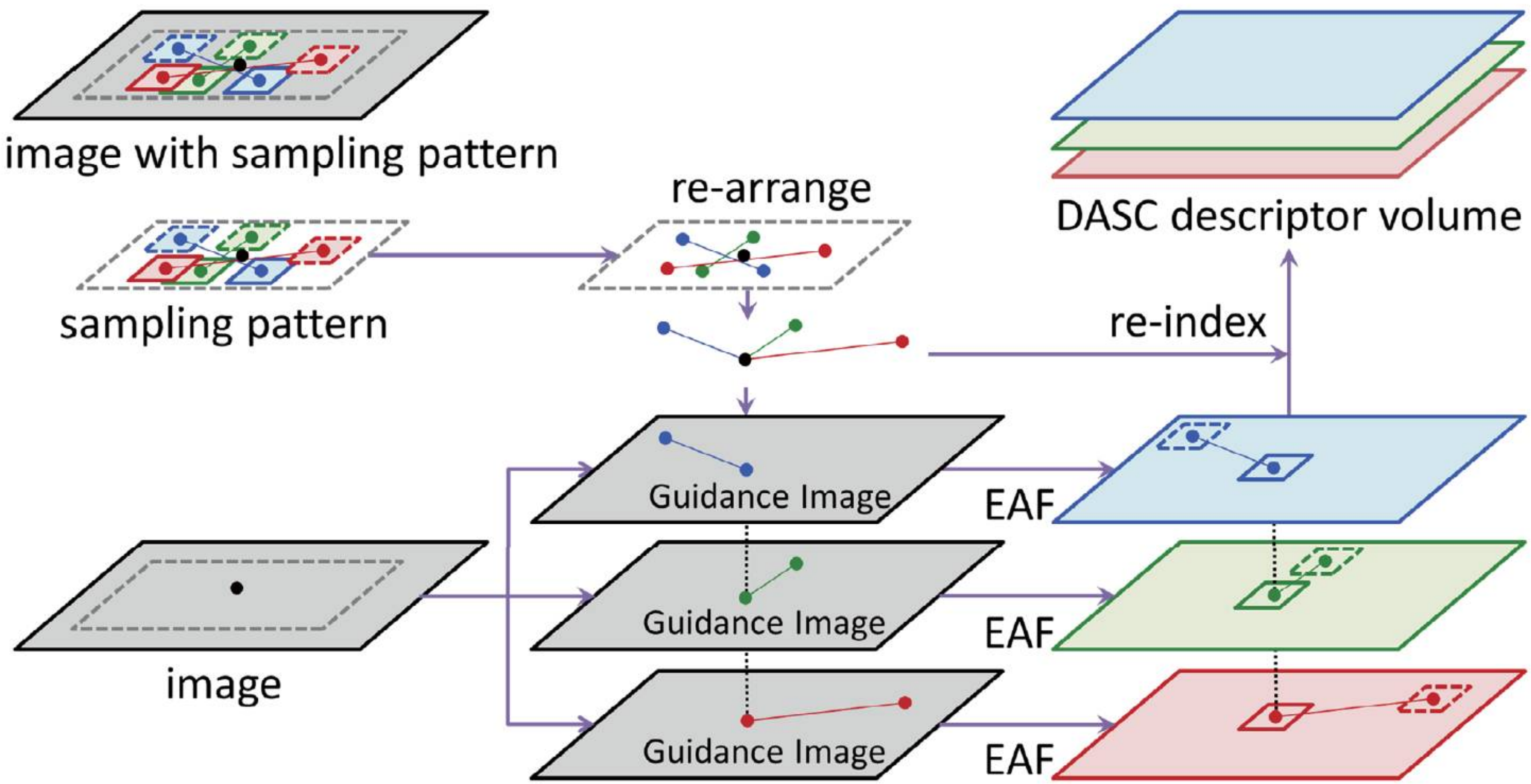
$$\boxed{\tilde{\Psi}(i,j) = \frac{\sum_{i',j'} \omega_{i,i'}(f_{i'} - \mathcal{G}_i)(f_{j'} - \mathcal{G}_{i,j})}{\sqrt{\sum_{i'} \omega_{i,i'}(f_{i'} - \mathcal{G}_i)^2} \sqrt{\sum_{i',j'} \omega_{i,i'}(f_{j'} - \mathcal{G}_{i,j})^2}}}$$

$$O(IL)$$

No dependency on the patch size!

Overall Process of DASC Descriptor

EAF: Edge-Aware Filtering



Note that all pixels share the same sampling pattern!

Computational Complexity Analysis

- Let I, N, and L represent an image size, a patch size, and the number of sampling patterns, respectively.
- A straightforward computation is extremely time-consuming, in specific, the computational complexity becomes $O(INL)$.
- Our approach removes the complexity dependency on the patch size N, *i.e.*, $O(IL)$. Furthermore, since there exist repeated offsets, the complexity is reduced as $O(I\tilde{L})$ for $\tilde{L} < L$.

image size	DAISY[6]	LSS	DASC*	DASC†
463×370	$2.5s$	$31s$	$128s$	$5s$

Table: Evaluation of computational complexity. The brute-force and efficient implementation of DASC is denoted as * and †, respectively.

[6] E. Tola, V. Lepetit, and P. Fua, Daisy: An efficient dense descriptor applied to wide-baseline stereo, IEEE TPAMI, 2010.

Experimental Environments

- We implemented the DASC descriptor in C++ on Intel Core i7-3770 CPU at 3.40 GHz, and measured the runtime on a single CPU core without further code optimizations and parallel implementation using multi-core CPUs/GPU.
- The DASC descriptor was evaluated with other state-of-the-art descriptors, *e.g.*, SIFT[7], DAISY, BRIEF[8], and LSS, and other area-based approaches, *e.g.*, ANCC[9] and RSNCC[10].

---

[7] D. Lowe. Distinctive image features from scale-invariant keypoints, IJCV, 60(2):91-110, 2004.

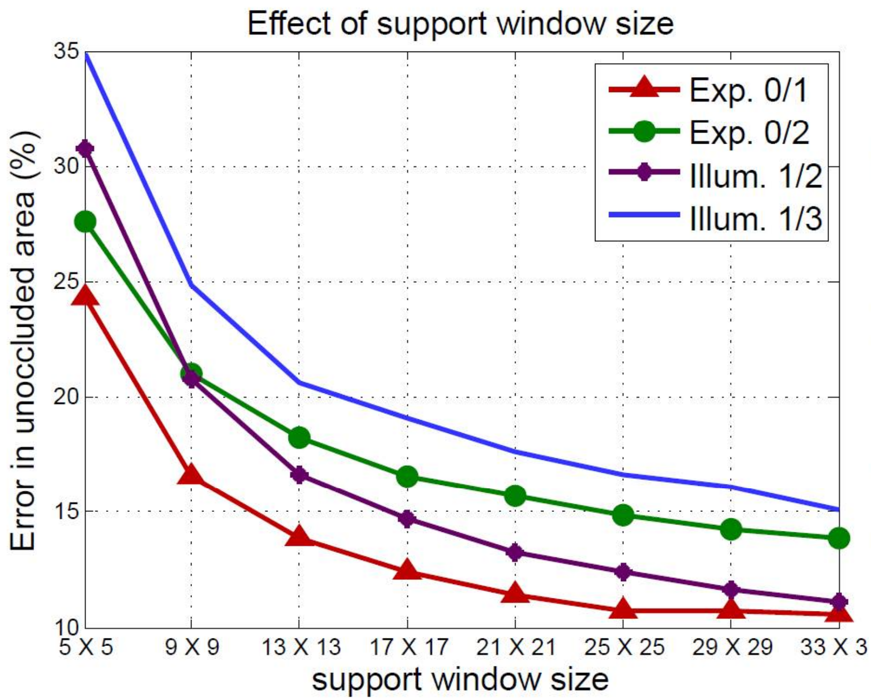
[8] M. Calonder. Brief: Computing a local binary descriptor very fast, IEEE TPAMI, 34(7):1281-1298, 2011.

[9] Y. Heo, K. Lee, and S. Lee. Joint depth map and color consistency estimation for stereo images with different illuminations and cameras, IEEE TPAMI, 35(5):1094-1106

[10] X. Shen, L. Xu, Q. Zhang, and J. Jia. Multi-modal and multi-spectral registration for natural images, ECCV, 2014.

Parameter Setting

Our DASC descriptor is constructed with the following same parameter settings for all datasets:
$\{\sigma, \tau, N, M, L\} = \{0.5, 0.03, 5 \times 5, 31 \times 31, 128\}$.



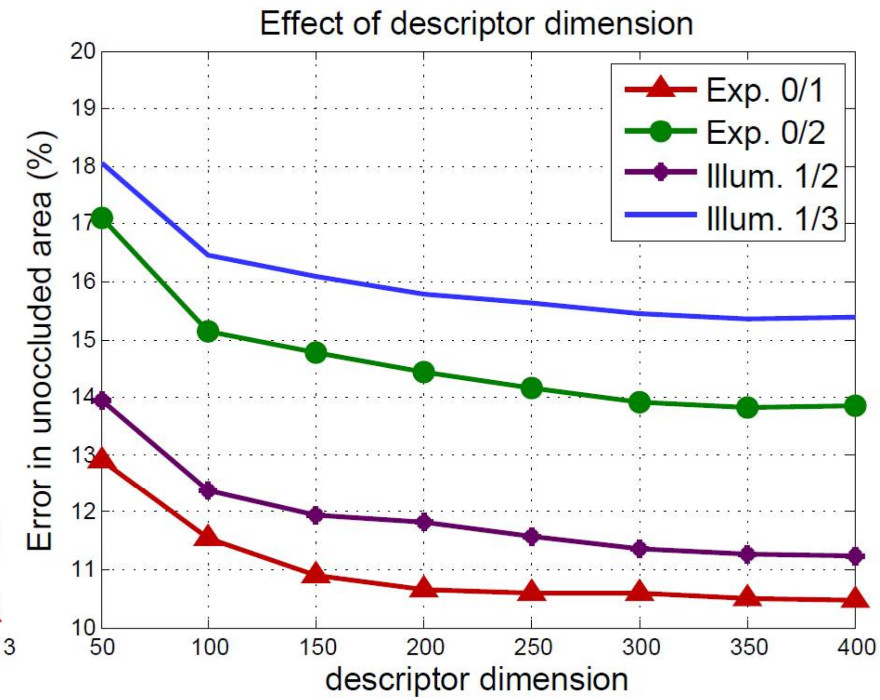


Figure: Average bad-pixel error rate on Middlebury benchmark of DASC+LRP descriptor with WTA optimization as varying support window size and descriptor dimension.

Middlebury Stereo Benchmark

We first evaluated our DASC+LRP descriptor in Middlebury stereo benchmark containing illumination and exposure variations.

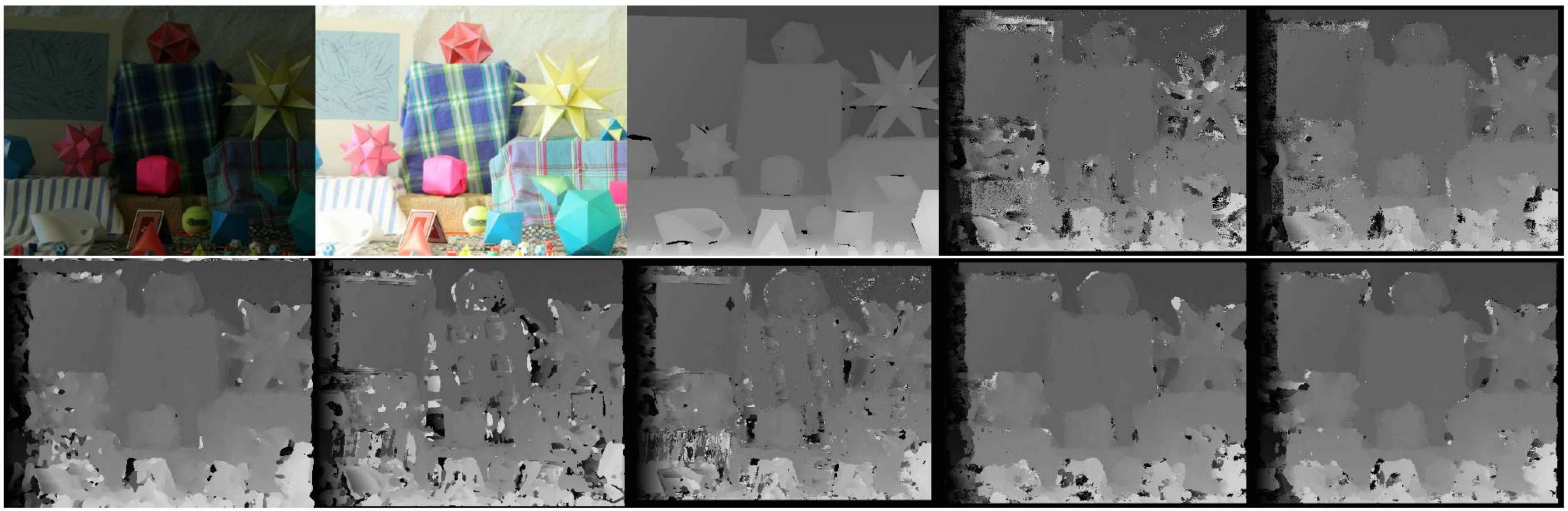


Figure: Comparison of disparity estimation for *Moebius* image pairs taken under illumination combination '0/2'. (from left to right, top and bottom) Left color image, right color image, and disparity maps for the ground truth, ANCC, BRIEF, DAISY, SIFT, LSS, DASC+RP, and DASC+LRP.

Middlebury Stereo Benchmark

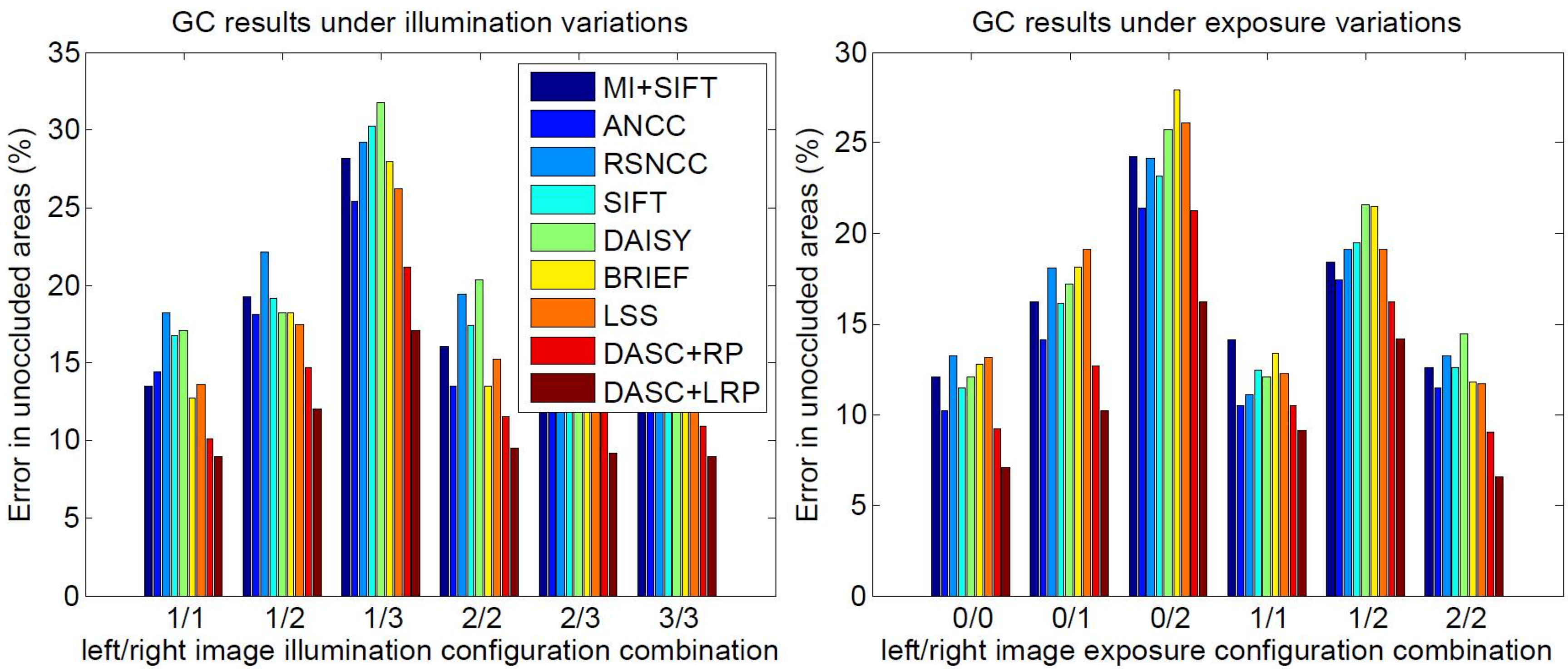
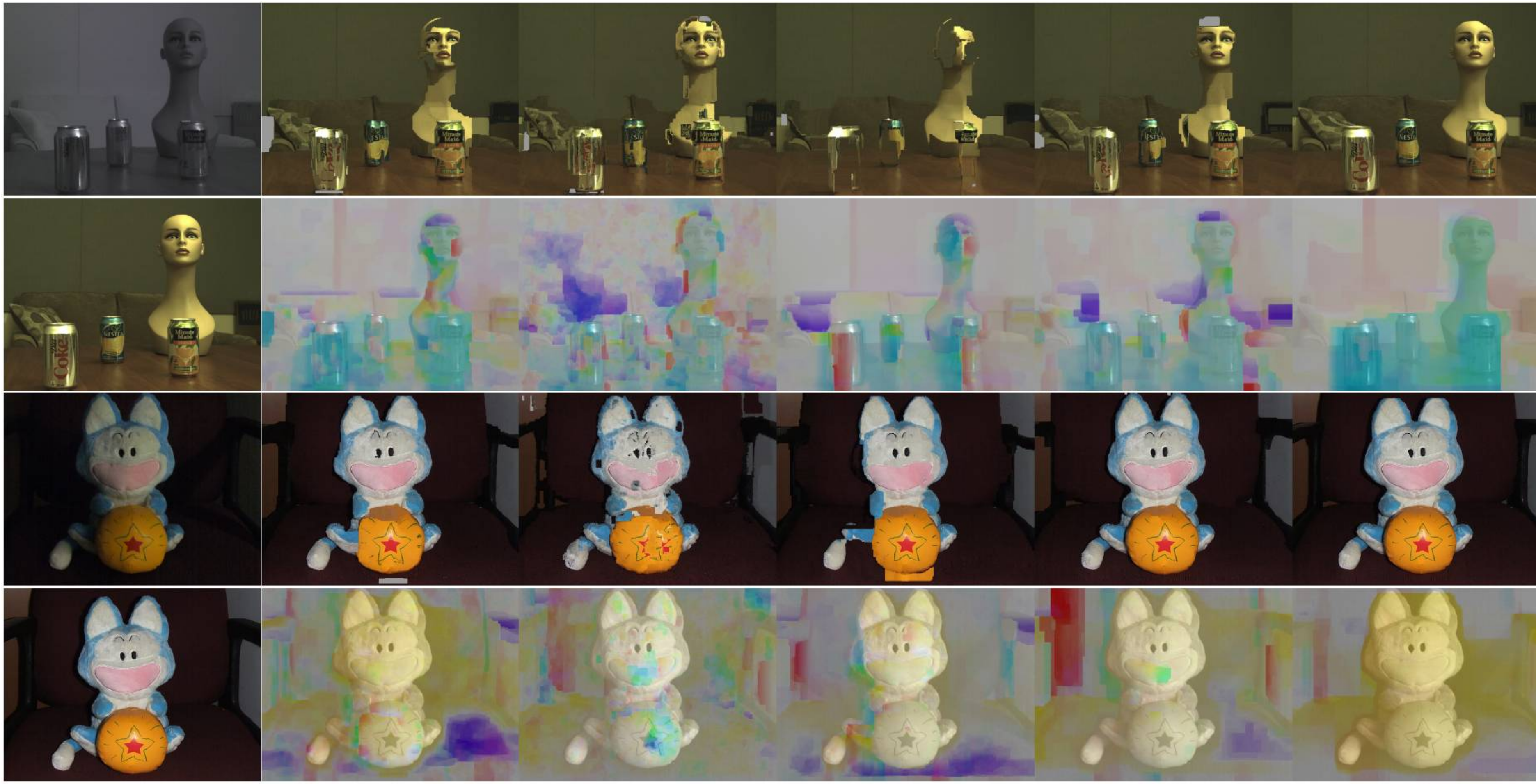


Figure: Average bad-pixel error rate on Middlebury benchmark with illumination variations and exposure variations. The GC was used for optimization. Our DASC+LRP shows the best performance.

Multi-modal and Multi-spectral Image Pairs



(a) (b) (c) (d) (e) (f)

Figure: Comparison of dense correspondence for RGB-NIR images and flash-noflash images for (a) input image pairs, (b) RSNCC, (c) BRIEF, (d) DAISY, (e) LSS, (f) DASC. The results consist of warped color images and 2-D flow fields.

Multi-modal and Multi-spectral Image Pairs

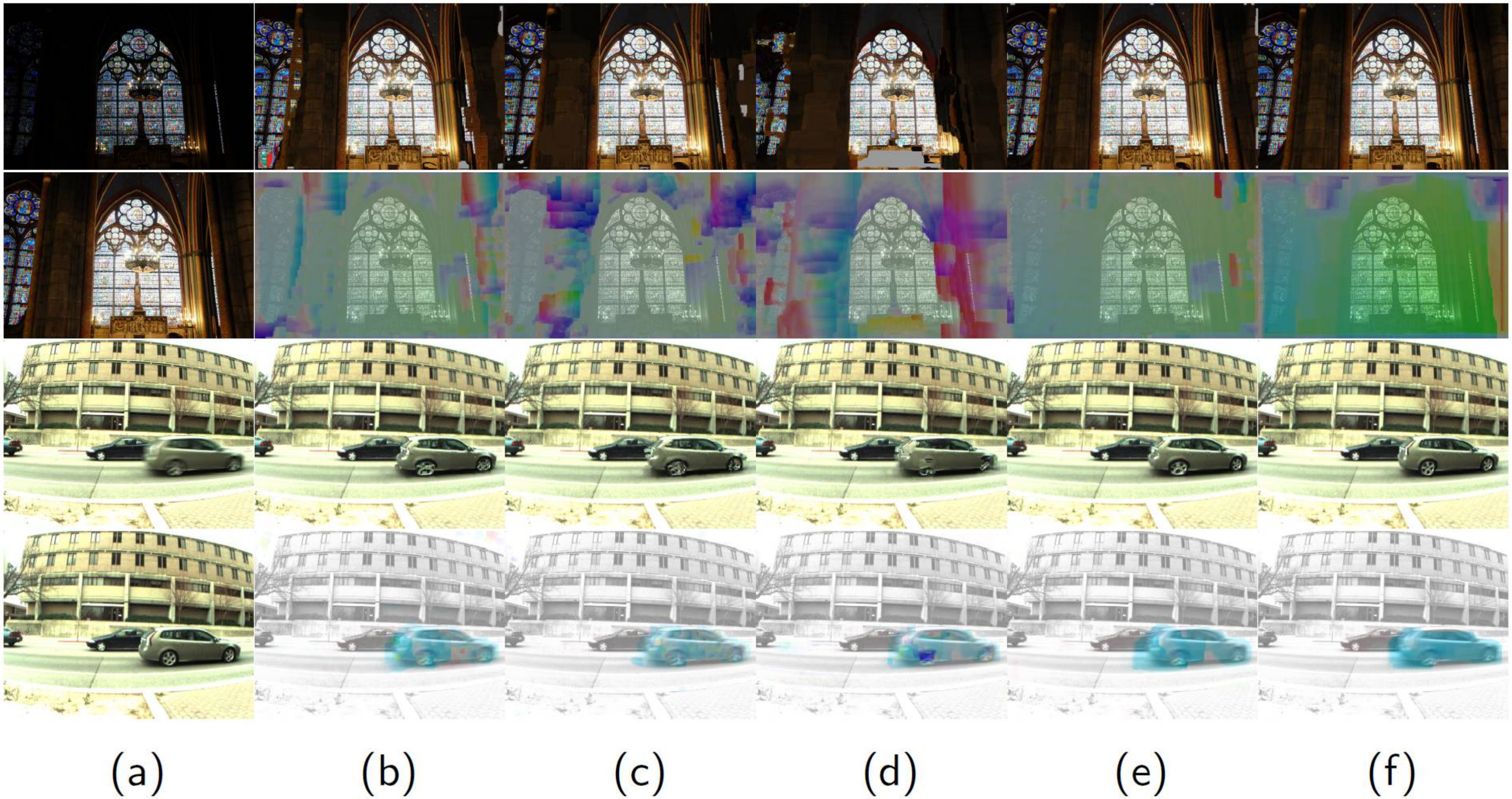


Figure: Comparison of dense correspondence for different exposure images and blurred-sharpen images for (a) input image pairs, (b) RSNCC, (c) BRIEF, (d) DAISY, (e) LSS, (f) DASC. The results consist of warped color images and 2-D flow fields.

Multi-modal and Multi-spectral Image Pairs

	RGB-NIR	Flash-noflash	Diff. Exp.	Blur-Sharp	Ave.
NRDC[11]	54.27	48.92	51.34	59.72	53.56
ANCC	18.45	14.14	11.96	19.24	15.94
RSNCC	13.41	15.87	9.15	18.21	14.16
SIFT	18.51	11.06	14.87	20.78	16.35
DAISY	20.42	10.84	12.71	22.91	16.72
BRIEF	17.54	9.21	9.54	19.72	14.05
LSS	16.14	11.88	9.11	18.51	13.91
DASC+RP	11.71	7.51	7.32	12.21	9.68
DASC+LRP	**8.10**	**5.41**	**6.24**	**10.81**	**7.64**

Table: Comparison of quantitative evaluation on multi-spectral and multi-modal images: hierarchical BP optimization was used.

[11] Y. HaCohen, E. Shechtman, D. B. Goldman, and D. Lischinski. Non-rigid dense correspondence with applications for image enhancement, ToG, 2011.

Concluding Remarks

- The robust novel local descriptor called the DASC has been proposed for dense multi-modal and multi-spectral matching.
  – Adaptive self-correlation measure and patch-wise receptive field pooling.

- **Secret Source**
  – **Speed**: With the fast edge-aware filters (**EAF**), our DASC descriptor can compute the dense descriptor very efficiently.
  – **Robustness and Accuracy**: 1) **Randomness** + 2) **Non-center biased** sampling + 3) Adaptive Self-Correlation (**ASC**)