A fast local descriptor for dense matching

CVPR 2008
Accepted by IEEE Trans. PAMI

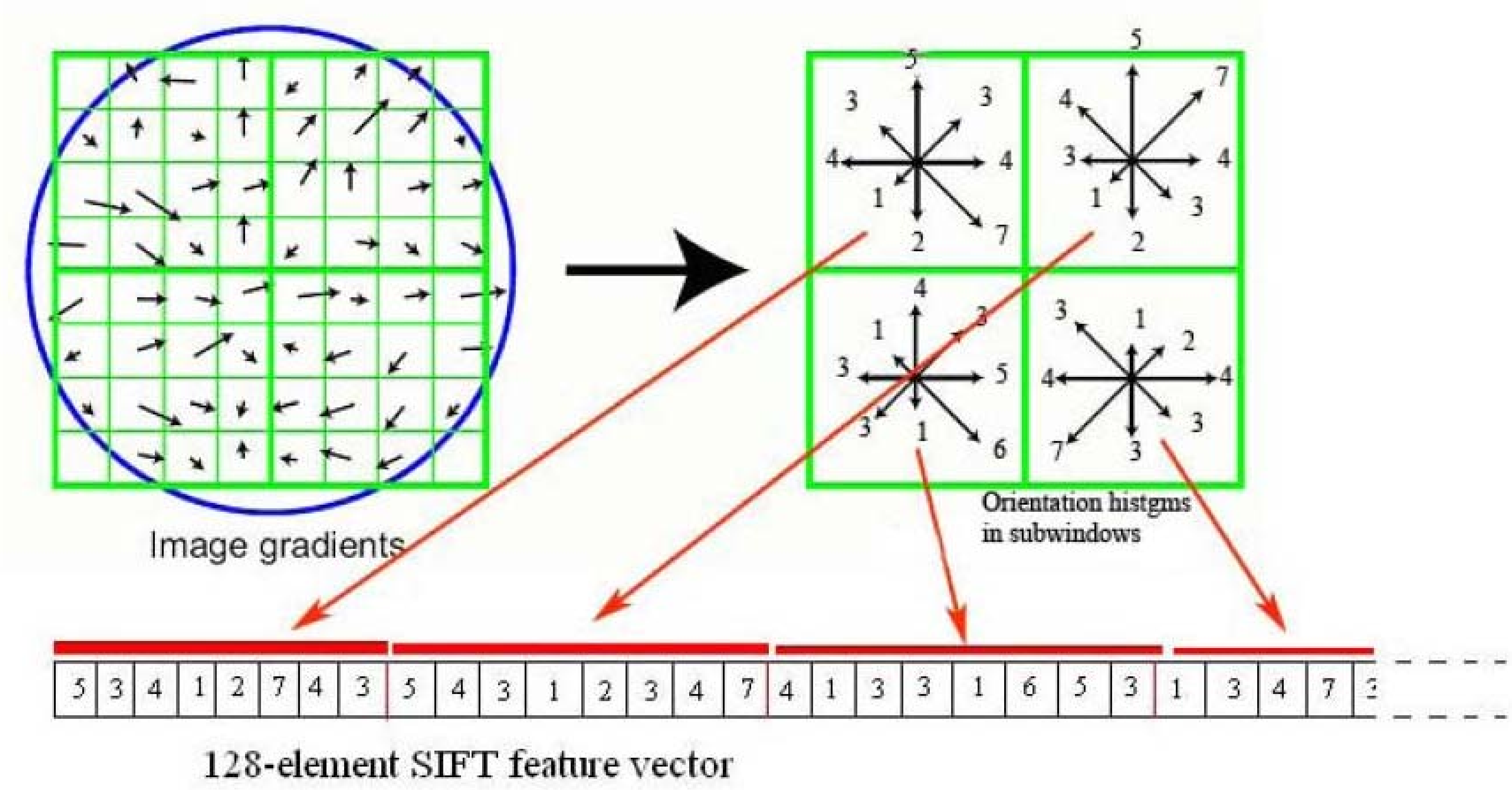
Engin Tola, Vincent Lepetit, Pascal Fua
Ecole Polytechnique Federale de Lausanne, Switzerland

Paper novelty

- introduces DAISY local image descriptor
  - much faster to compute than SIFT for dense point matching
  - works on the par or better than SIFT
- DAISY descriptors are fed into expectation-maximization (EM) algorithm which uses graph cuts to estimate the scene's depth

SIFT local image descriptor

- Each bin contains a weighted sum of the norms of the image gradients around its center, where the weights roughly depend on the distance to the bin center



DAISY local image descriptor

- Gaussian convolved orientation maps are calculated for every direction

$$\mathbf{G}_o^{\Sigma} = G_{\Sigma} * \left(\frac{\partial \mathbf{I}}{\partial o}\right)^{+}$$

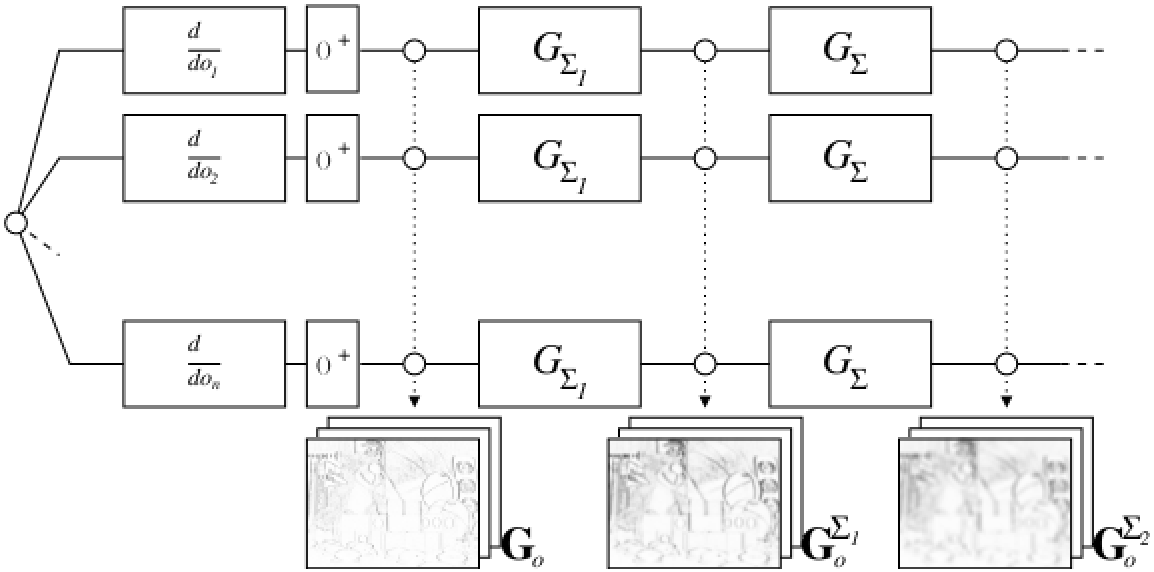
G_{Σ} : Gaussian convolution filter with variance Σ

$\frac{\partial \mathbf{I}}{\partial o}$: image gradient in direction o

$(.)^{+}$: operator $(a)^{+} = \max(a, 0)$

- It is observed that every location $\mathbf{G}_o^{\Sigma}$ contains a value very similar to what a bin in SIFT contains: a weighted sum computed over an area of gradient norms

DAISY local image descriptor



DAISY local image descriptor

I. $\mathbf{h}_\Sigma(u,v)$ at every pixel location are computed

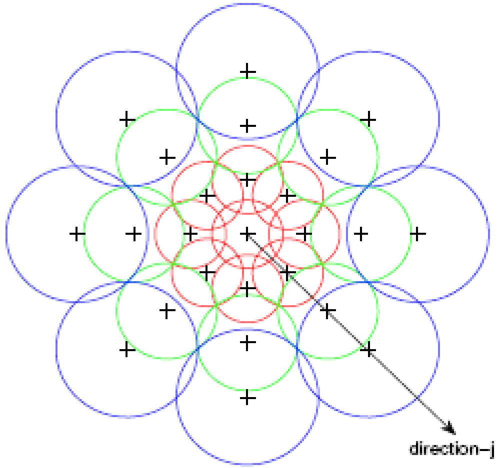
$$\mathbf{h}_\Sigma(u,v) = \left[\mathbf{G}_1^\Sigma(u,v), \ldots, \mathbf{G}_8^\Sigma(u,v)\right]^\top ,$$

$\mathbf{G}_1^\Sigma$: Gaussian convolved orientation map

II. Vectors h are normalized to unit norm

III. Local image descriptor is computed as

$$\begin{aligned}
&\mathcal{D}(u_0, v_0) = \\
\Big[\;& \widetilde{\mathbf{h}}_{\Sigma_1}^\top(u_0, v_0), \\
& \widetilde{\mathbf{h}}_{\Sigma_1}^\top(\mathbf{l}_1(u_0, v_0, R_1)), \cdots, \widetilde{\mathbf{h}}_{\Sigma_1}^\top(\mathbf{l}_N(u_0, v_0, R_1)), \\
& \widetilde{\mathbf{h}}_{\Sigma_2}^\top(\mathbf{l}_1(u_0, v_0, R_2)), \cdots, \widetilde{\mathbf{h}}_{\Sigma_2}^\top(\mathbf{l}_N(u_0, v_0, R_2)), \\
& \widetilde{\mathbf{h}}_{\Sigma_3}^\top(\mathbf{l}_1(u_0, v_0, R_3)), \cdots, \widetilde{\mathbf{h}}_{\Sigma_3}^\top(\mathbf{l}_N(u_0, v_0, R_3)) \;\Big]^\top
\end{aligned}$$



参数设置

$R_1=2.5$ $\quad\quad$ $\Sigma_1 = 2.55$

$R_2=3R_1$ $\quad\quad$ $\Sigma_2 = 3\Sigma_1$

$R_3=6R_1$ $\quad\quad$ $\Sigma_3 = 5\Sigma_1$

描述子维数=（3*8+1）*8=200

（25个位置，每个位置8个方向）

DAISY vs SIFT: computational complexity

- Convolution is time-efficient for separable kernels like Gaussian
- Convolution maps with larger Gaussian kernel can be built upon convolution maps with smaller Gaussian kernel:

$$\mathbf{G}_o^{\Sigma_2} = G_{\Sigma_2} * \left(\frac{\partial \mathbf{I}}{\partial o}\right)^+ = G_{\Sigma} * G_{\Sigma_1} * \left(\frac{\partial \mathbf{I}}{\partial o}\right)^+ = G_{\Sigma} * \mathbf{G}_o^{\Sigma_1},$$

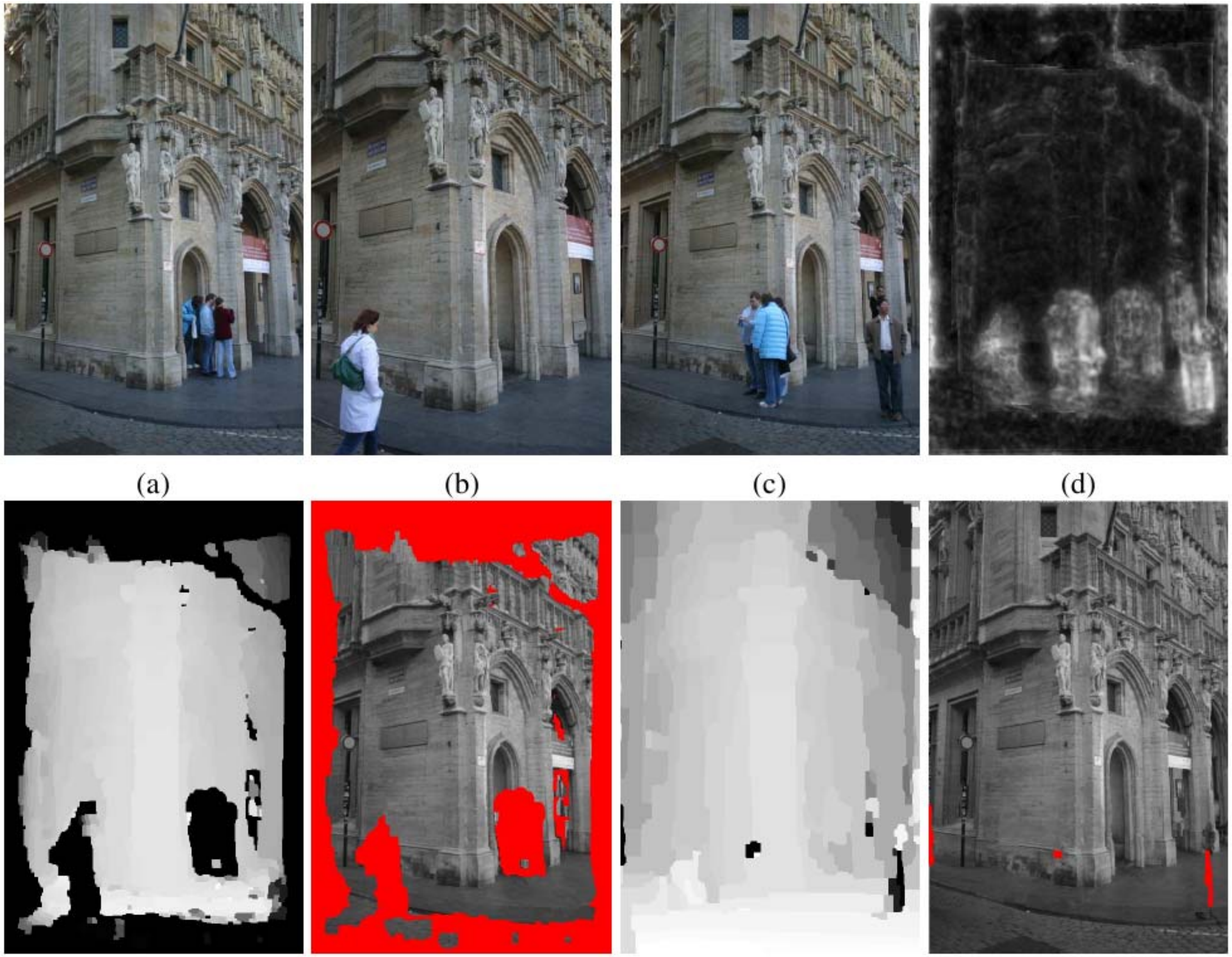
with $\Sigma = \sqrt{\Sigma_2^2 - \Sigma_1^2}$.

Image Size	DAISY	SIFT
800x600	5	252
1024x768	10	432
1290x960	13	651

Table 1. Computation Time Comparison (in seconds)

Results

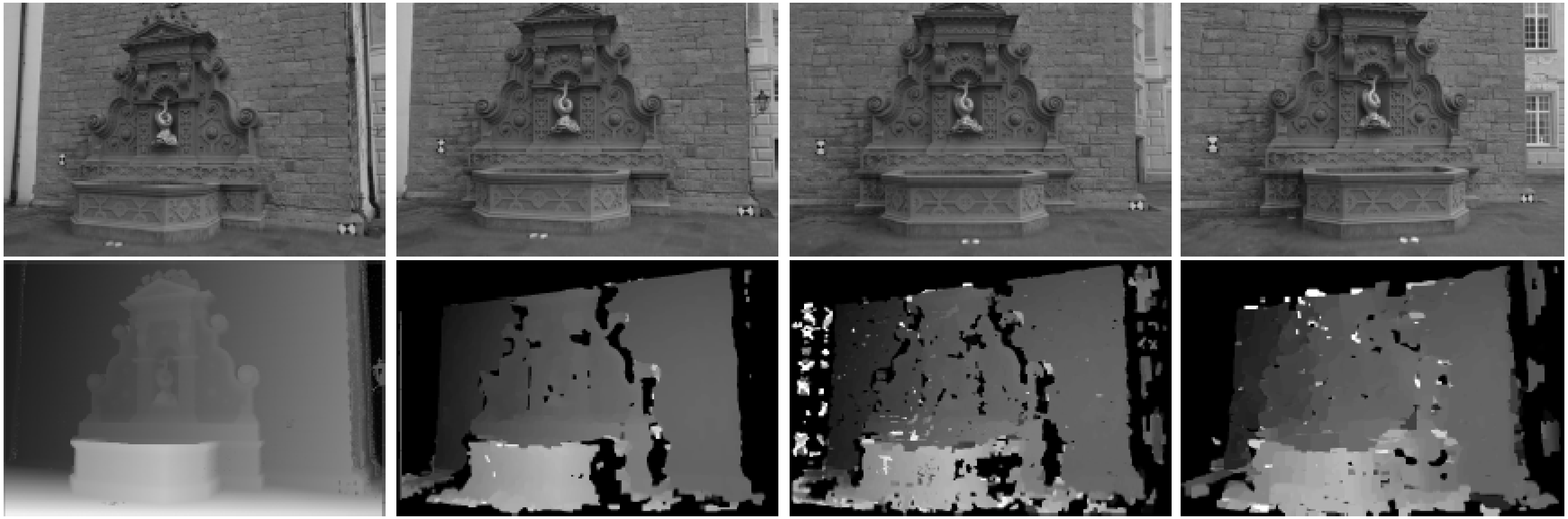
Results



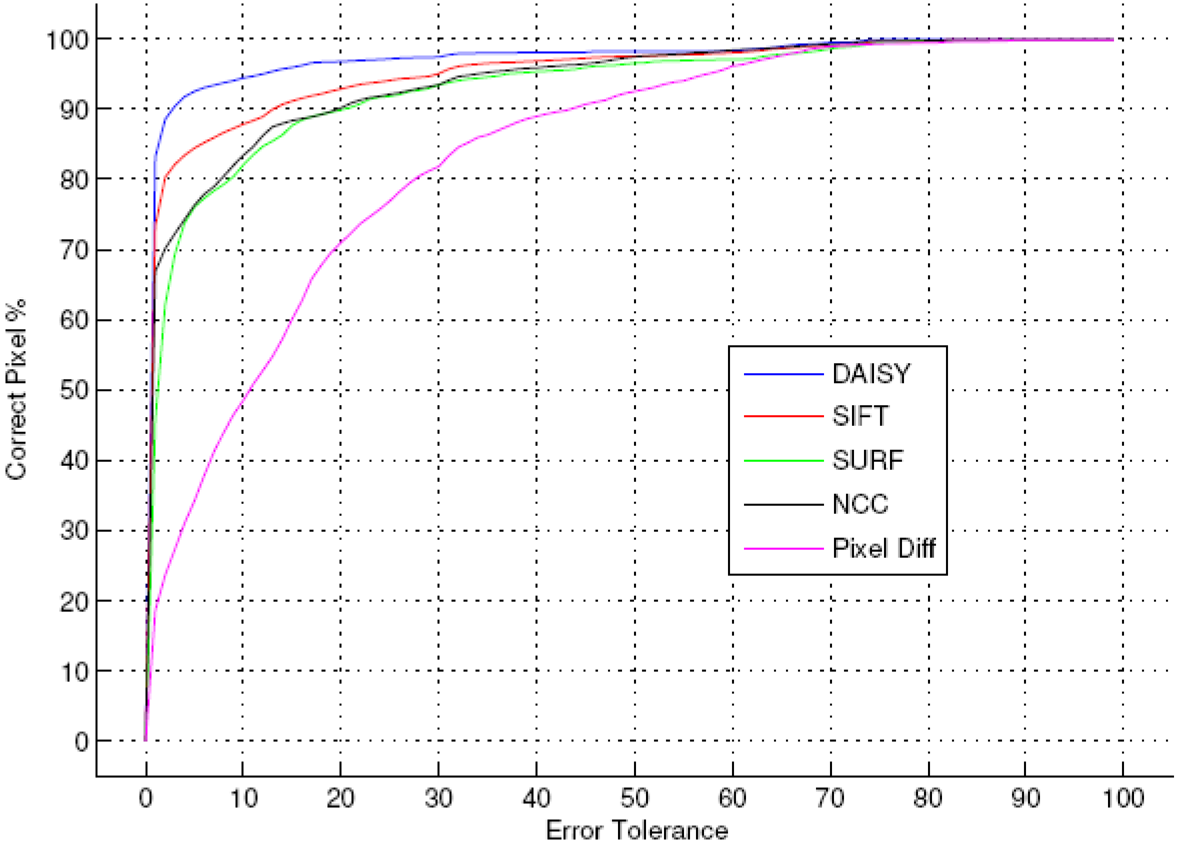
(a) (b) (c) (d)

(e) (f) (g) (h)

Results



Results



100
90
80
70
60
50
40
30
20
10
0
Correct Pixel %
0
10
20
30
40
50
60
70
80
90
100
Error Tolerance
DAISY
SIFT
SURF
NCC
Pixel Diff