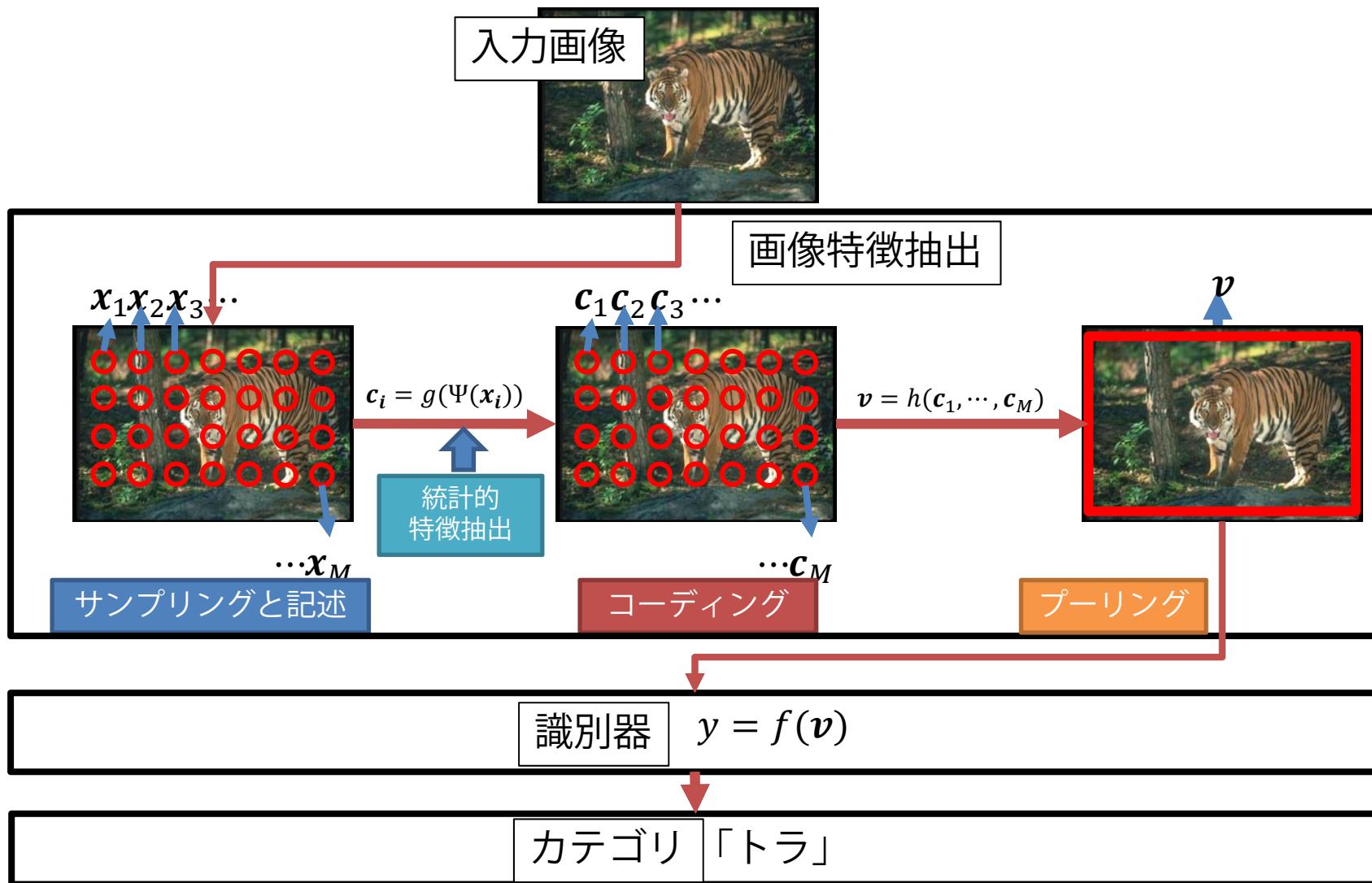


知能情報論 局所特徴・検出器・記述子

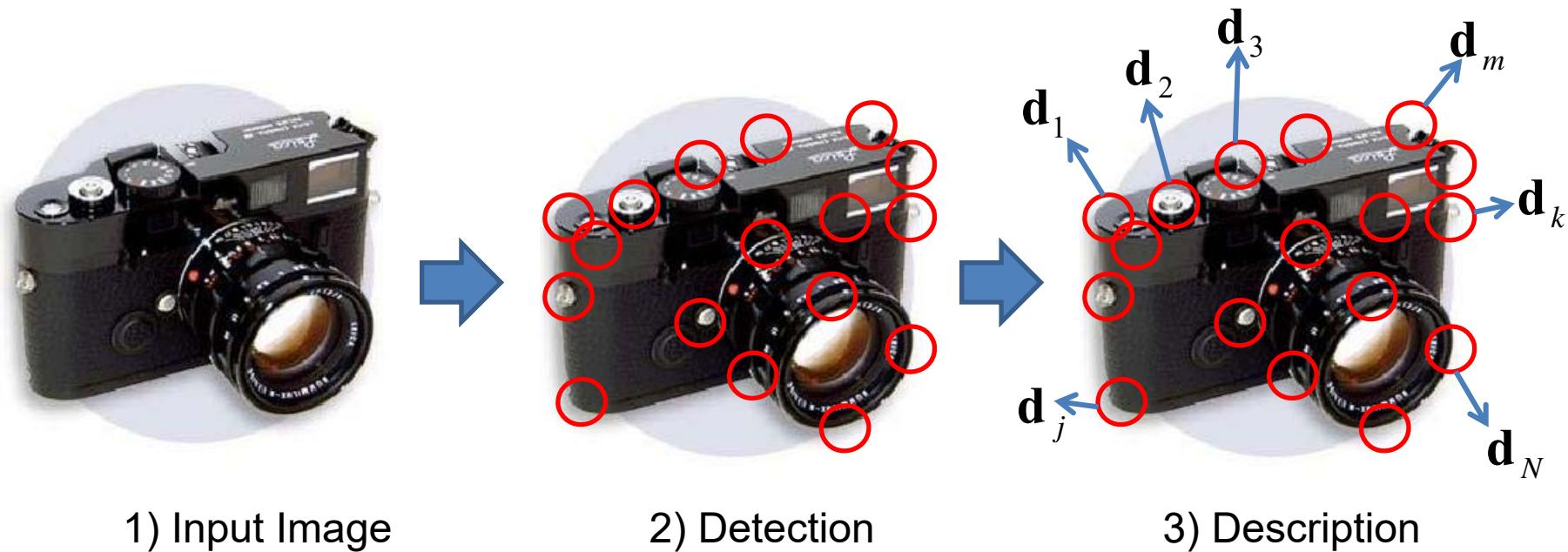
2016年5月11日

東京大学 大学院情報理工学系研究科
原田達也

カテゴリ認識の流れ



局所特徴量の抽出手順

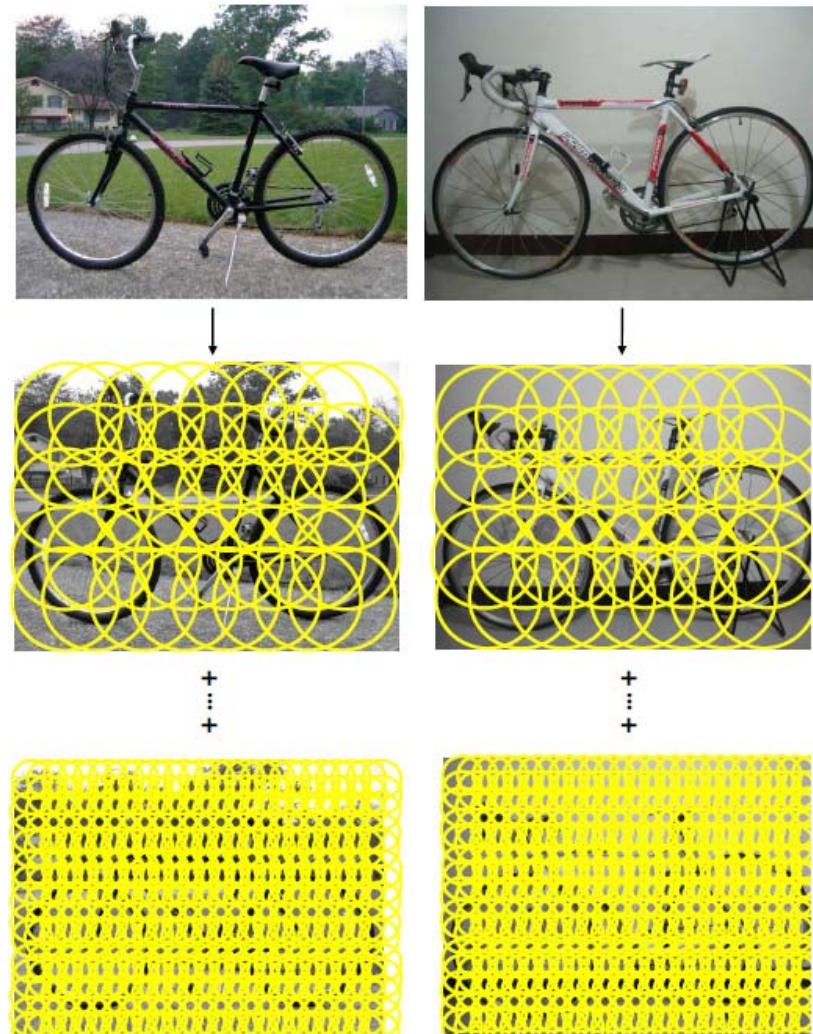


- Detection
 - 画像の中から特徴的な点を検出すること.
 - 画像内の位置(x,y)を検出.
- Description
 - 特徴点周りの画像の様子を表現すること.
 - 一般的に多次元ベクトル.
 - RGB値, 輝度勾配など.

Detection: sampling

K. Grauman and B. Leibe

Dense sampling
グリッド上など



特定物体には検出器は（一般的に）有効
一般物体には密なサンプリングが（一般的に）有効 5

ICCV2009

- Sameer Agarwal, Noah Snavely, Ian Simon, Steven M. Seitz, and Richard Szeliski. Building Rome in a Day. In ICCV, 2009.



ローマ コロッセオ

ICCV2009

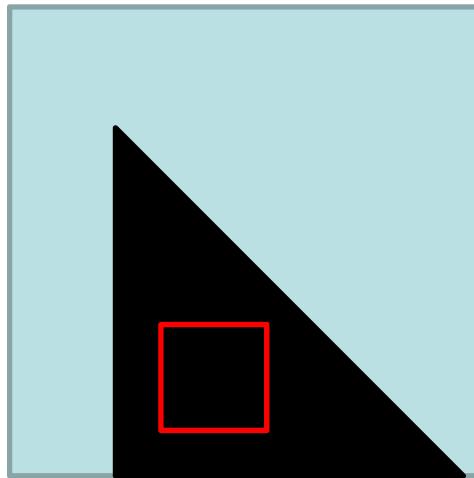
- Sameer Agarwal, Noah Snavely, Ian Simon, Steven M. Seitz, and Richard Szeliski.
Building Rome in a Day. In ICCV, 2009.



クロアチア ドゥブロヴニク

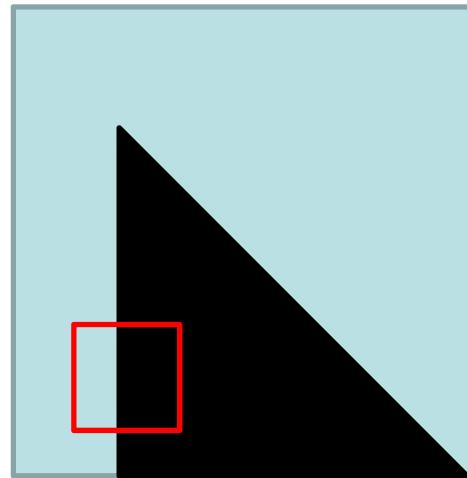
Harris Corner Detector

C. Harris, and M. Stephens. A combined corner and edge detector. 4th Alvey Conf., pp. 147-152, 1988.



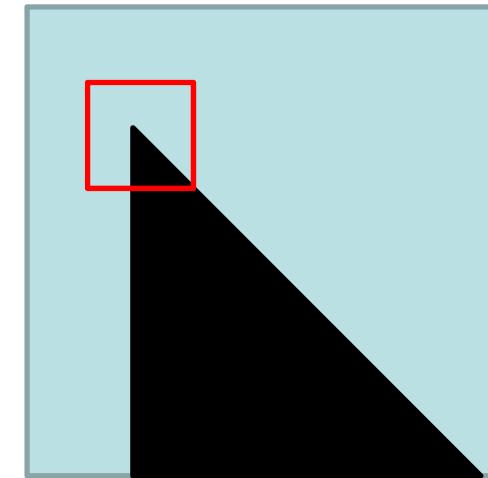
平坦領域

小領域を微小移動させても小領域内は変化しない。



エッジ

小領域をエッジに沿って微小移動させても小領域内は変化しない。エッジとは垂直方向に微小移動させると変化する。



コーナー

小領域を任意の方向に微小移動させても小領域内は変化する。

Harris Corner Detector

- 微小変動 (u, v) に対する小領域の輝度の変化を調べる

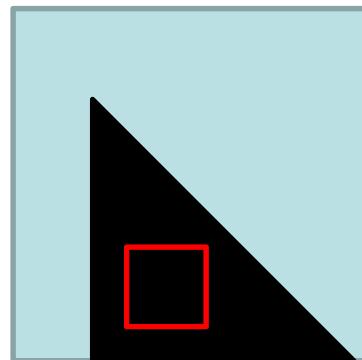
小領域のピクセルに
対して足しあわせる

微小変位させた
位置での輝度

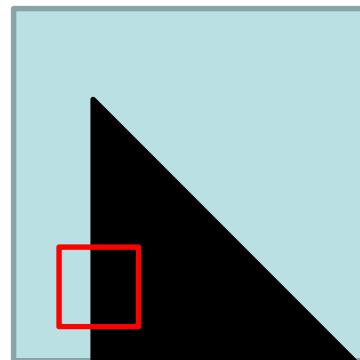
輝度

$$E(u, v) = \sum_{x, y} w(x, y)[I(x + u, y + v) - I(x, y)]^2$$

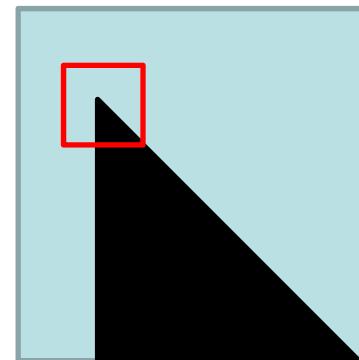
窓関数 Uniform or Gaussian



平坦領域
小領域を微小移動させ
ても小領域内は変化し
ない。



エッジ
小領域をエッジに沿つ
て微小移動させても小
領域内は変化しない。
エッジとは垂直方向に
微小移動させると変化
する。



コーナー
小領域を任意の方向に
微小移動させても小領
域内は変化する。

Harris Corner Detector

$$E(u, v) = \sum_{x, y} w(x, y) [I(x+u, y+v) - I(x, y)]^2$$

テーラー展開して整理



$$E(u, v) = [u, v] M \begin{bmatrix} u \\ v \end{bmatrix}$$
$$M = \sum_{x, y} w(x, y) \begin{bmatrix} I_x^2 & I_x I_y \\ I_x I_y & I_y^2 \end{bmatrix}$$

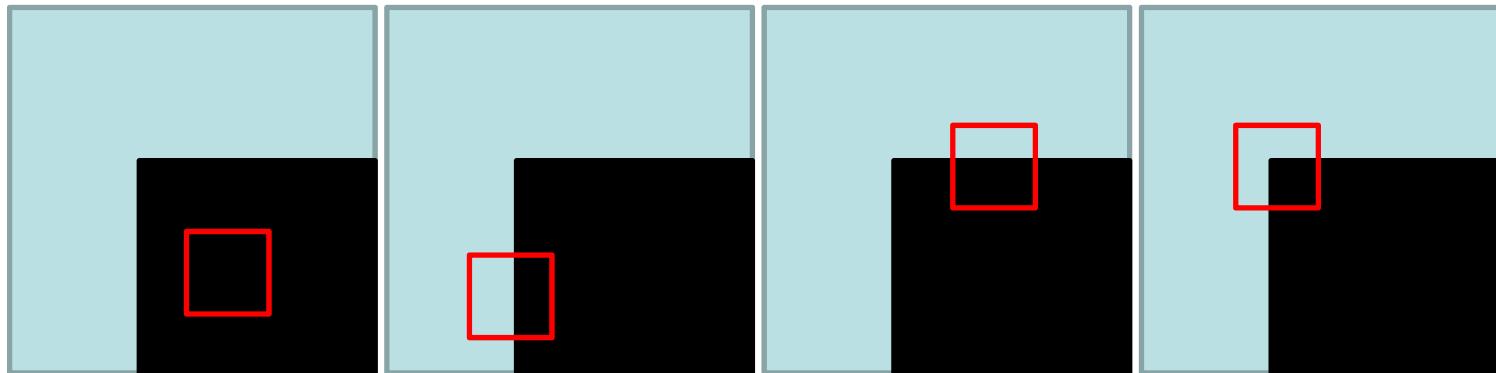
x方向の勾配とy方向の勾配の積
y方向の勾配の二乗

Harris Corner Detector

$$E(u, v) = [u, v] M \begin{bmatrix} u \\ v \end{bmatrix}$$

$$M = \sum_{x,y} w(x, y) \begin{bmatrix} I_x^2 & I_x I_y \\ I_x I_y & I_y^2 \end{bmatrix} \quad \leftarrow ?$$

最も単純なケースで考える



$$M = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}$$

$$M = \begin{bmatrix} \lambda_1 & 0 \\ 0 & 0 \end{bmatrix}$$

$$M = \begin{bmatrix} 0 & 0 \\ 0 & \lambda_2 \end{bmatrix}$$

$$M = \begin{bmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{bmatrix}$$

つまり $M = \begin{bmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{bmatrix}$ として

$\lambda_1 \approx \lambda_2 \approx 0 \rightarrow$ 平坦

$\lambda_1 \gg \lambda_2 \approx 0$ or $\lambda_2 \gg \lambda_1 \approx 0 \rightarrow$ エッジ

$\lambda_1 \approx \lambda_2 \gg 0 \rightarrow$ コーナー

Harris Corner Detector

一般的に考える

$$E(u, v) = [u, v] M \begin{bmatrix} u \\ v \end{bmatrix}$$
$$M = \sum_{x, y} w(x, y) \begin{bmatrix} I_x^2 & I_x I_y \\ I_x I_y & I_y^2 \end{bmatrix}$$

回転行列（Mの単位固有ベクトル）を利用して

$$M = U \begin{bmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{bmatrix} U^{-1} \quad U = (\mathbf{u}_1 \ \mathbf{u}_2) \quad M\mathbf{u}_1 = \lambda_1 \mathbf{u}_1$$
$$\lambda_1 \geq \lambda_2 \quad M\mathbf{u}_2 = \lambda_2 \mathbf{u}_2$$

これは前のスライドの一般形となっている。

つまり、うまく回転させれば前述のような λ の値のみを利用
してコーナーが容易に判断可能。

Harris Corner Detector

- λ の値の関係を利用してコーナーを判断

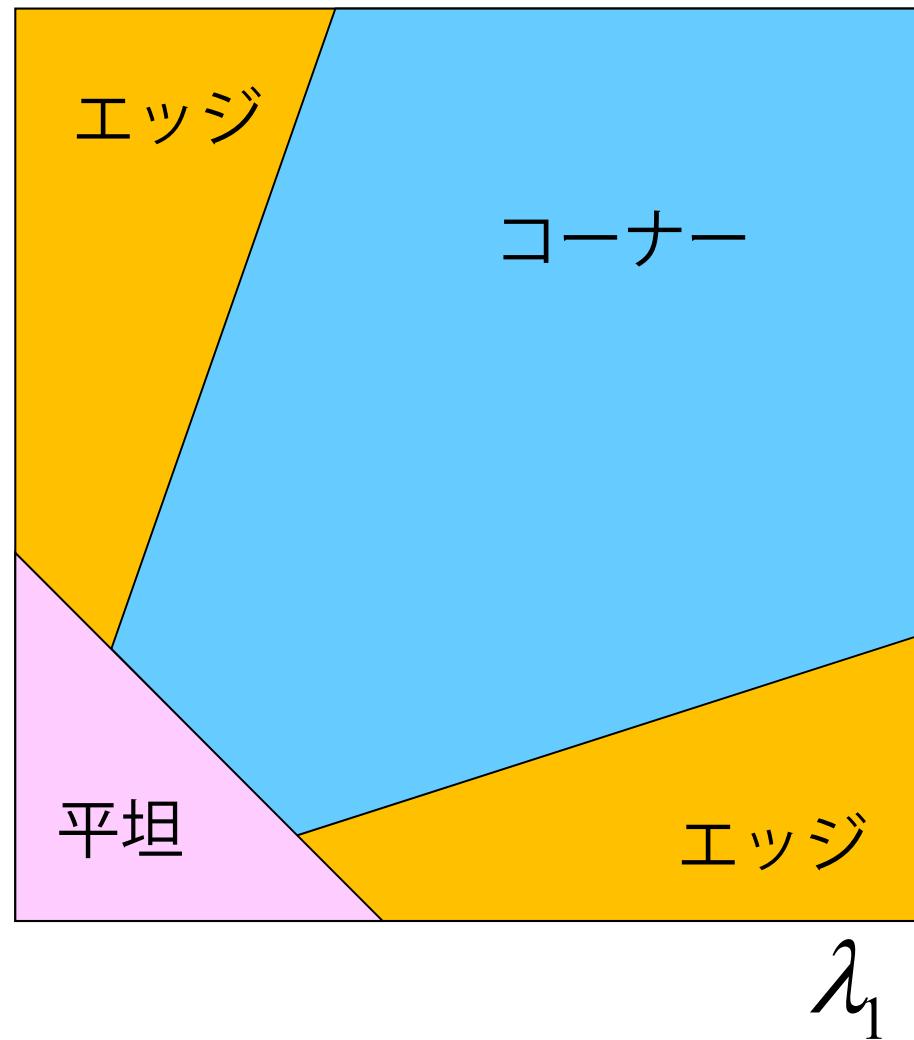
$$M = U \begin{bmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{bmatrix} U^{-1}$$

$$U = (\mathbf{u}_1 \ \mathbf{u}_2)$$

$$\lambda_1 \geq \lambda_2$$

$$M\mathbf{u}_1 = \lambda_1 \mathbf{u}_1$$

$$M\mathbf{u}_2 = \lambda_2 \mathbf{u}_2$$



Harris Corner Detector

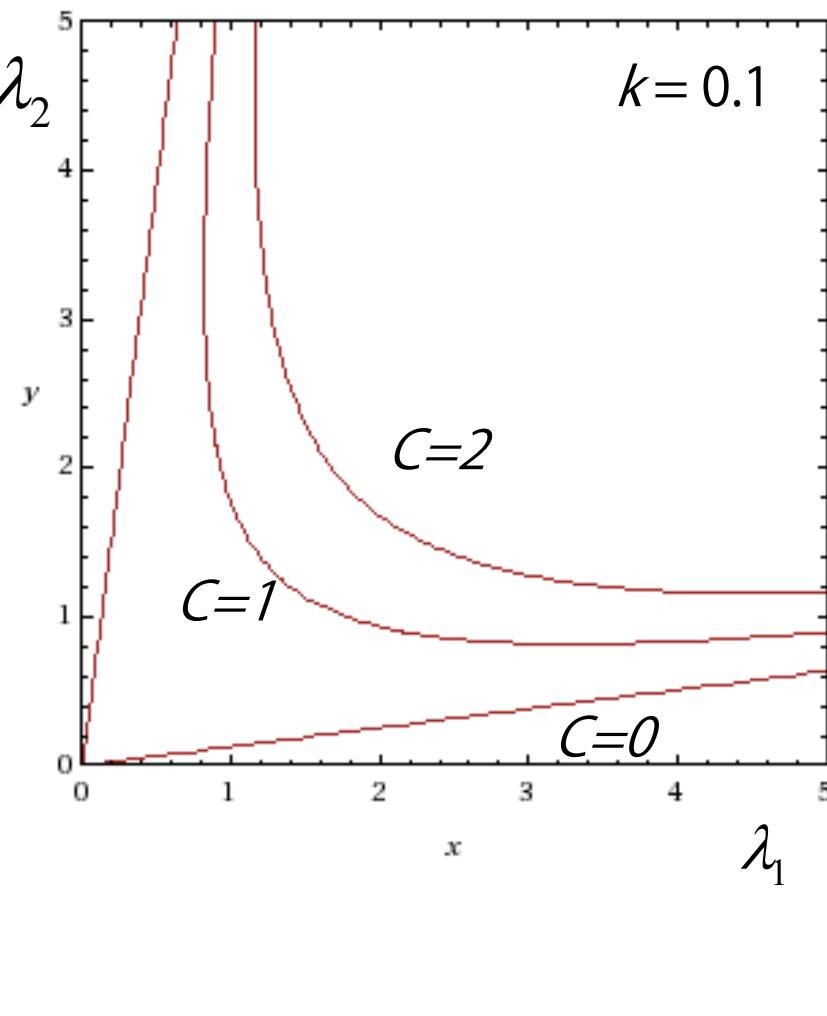
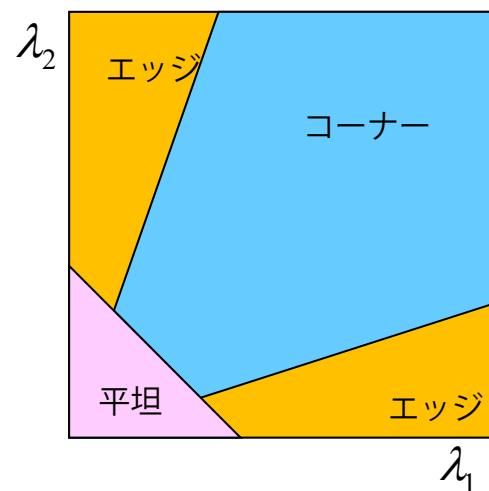
- 固有値を直接計算せずにコーナーか判断

$$C = \text{Det}(M) - k(\text{Tr}(M))^2$$

$$\text{Tr}(M) = \lambda_1 + \lambda_2$$

$$\text{Det}(M) = \lambda_1 \lambda_2$$

$$k = 0.04-0.06$$



SIFT

- Scale-invariant feature transform
 - 論文を投稿しても、画像特徴にSIFTを含めなければそれだけで認められないくらい有名な特徴
 - 回転不变、スケール不变
 - ディテクタ+局所記述子の組み合わせ
 - 表現能力は高いが計算が遅い
- 文献
 - D. Lowe. Object recognition from local scale-invariant features. In ICCV, 1999.
 - D. Lowe. Distinctive Image Features from Scale-Invariant Keypoints. International Journal of Computer Vision, Vol.60, No.2, pp. 91-110, 2004.

特徴量を記述する領域

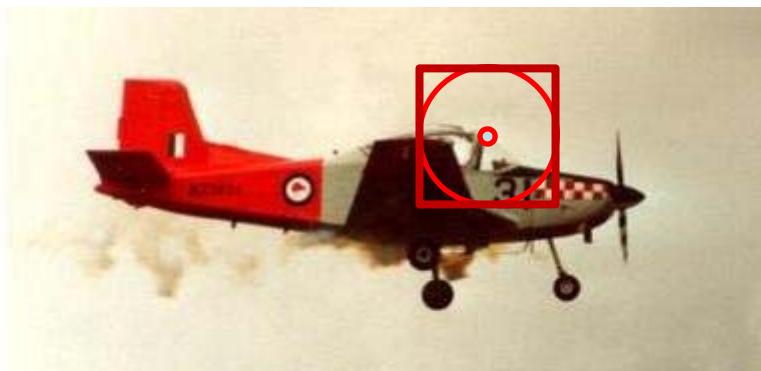
1. 着目するピクセルを選択する
 - ・ 検出器, サンプリング



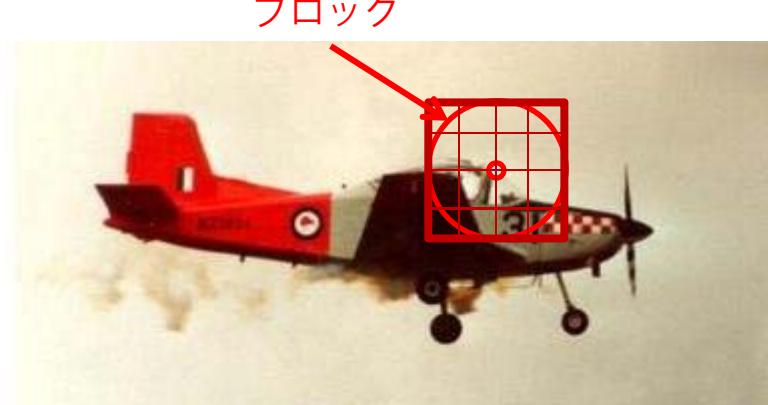
2. 着目ピクセルから特徴量を記述する円領域を決める



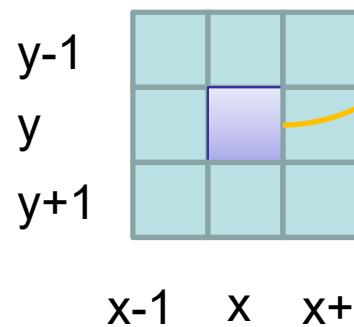
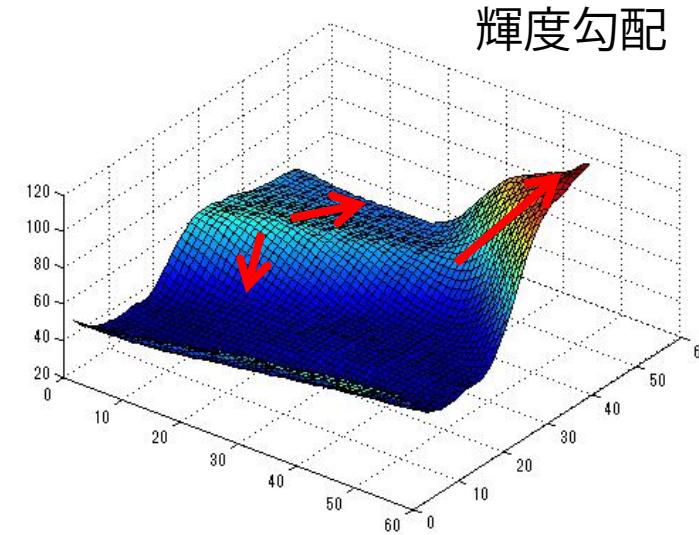
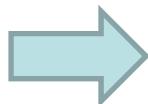
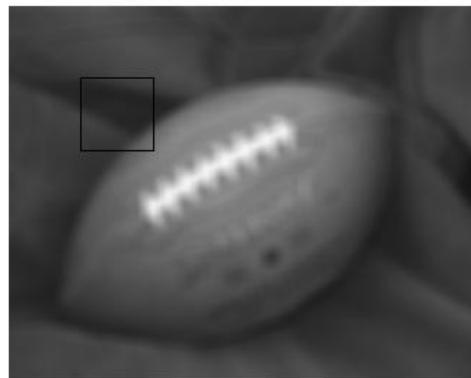
3. 円領域をちょうどカバーする正方形領域を決定する



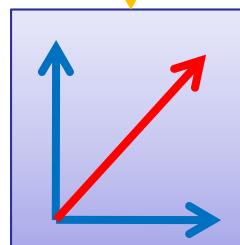
4. 正方形領域を均等に16分割する



勾配の計算



f_y



輝度勾配

局所輝度勾配の計算

$$f_x = L(x+1, y) - L(x-1, y)$$

$$f_y = L(x, y+1) - L(x, y-1)$$

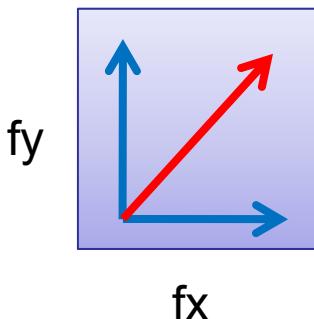
勾配の強度 $m(x, y) = \sqrt{(L(x+1, y) - L(x-1, y))^2 + (L(x, y+1) - L(x, y-1))^2}$

勾配の角度 $\theta(x, y) = \tan^{-1} \left(\frac{L(x, y+1) - L(x, y-1)}{L(x+1, y) - L(x-1, y)} \right)$

- 着目領域内の全てのピクセルで勾配を計算する。

勾配のヒストグラム化

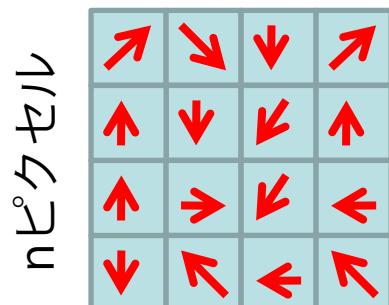
1. 注目ピクセル近傍で輝度勾配を計算



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

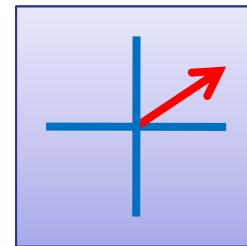
角度 $\theta(x, y) = \tan^{-1} \left(\frac{L(x, y+1) - L(x, y-1)}{L(x+1, y) - L(x-1, y)} \right)$

3. ブロック内のピクセルを
ひとまとめりとする
 n ピクセル

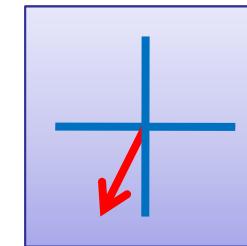


ブロック

2. 勾配方向の量子化 (45° ずつ, 8方向)

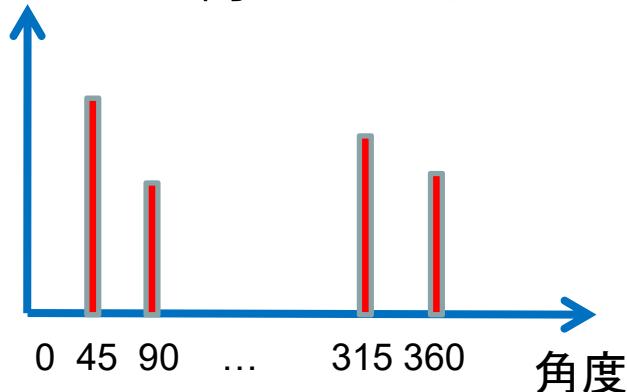


$37^\circ \rightarrow 45^\circ$

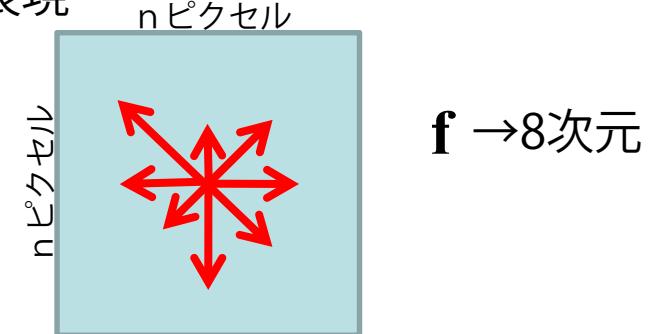


$256^\circ \rightarrow 270^\circ$

4. ブロック内のヒストグラムを作成



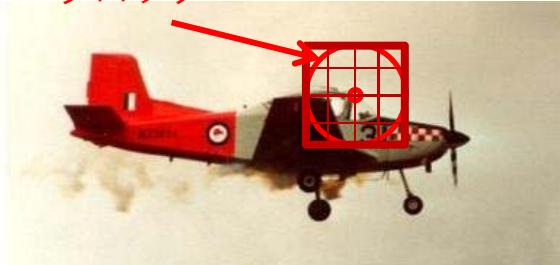
5. ブロックを表現
するベクトル



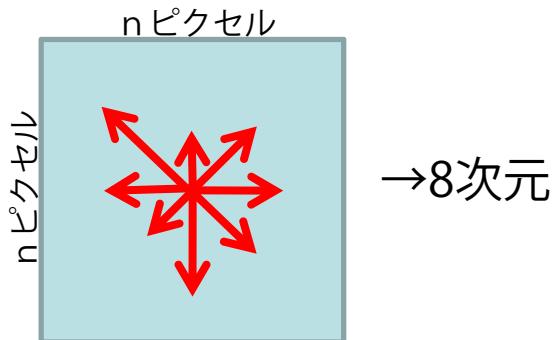
18

局所特徴量の計算

ブロック



1. ブロックを表現するベクトル



3. 16ブロックの勾配ヒストグラムをまとめて一つのベクトルとする

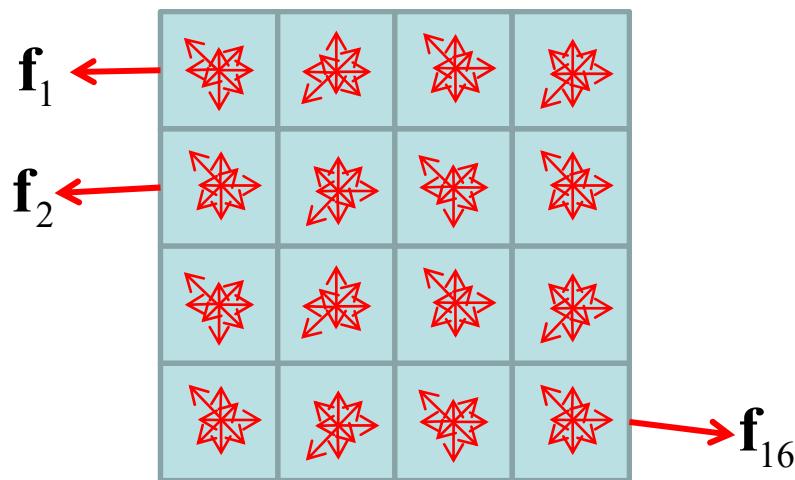
$$\mathbf{f}^T = \left(\mathbf{f}_1^T \ \mathbf{f}_2^T \ \cdots \ \mathbf{f}_{16}^T \right)$$

つまり、8次元×16ブロック=128次元のベクトルとなる

(注) ここまでプロセスをSIFT descriptorと呼ぶ。

SIFTとは特徴点検出+SIFT descriptorのことで、両者は最近では区別される

2. 着目領域内の16ブロック全てに勾配ヒストグラムを計算する



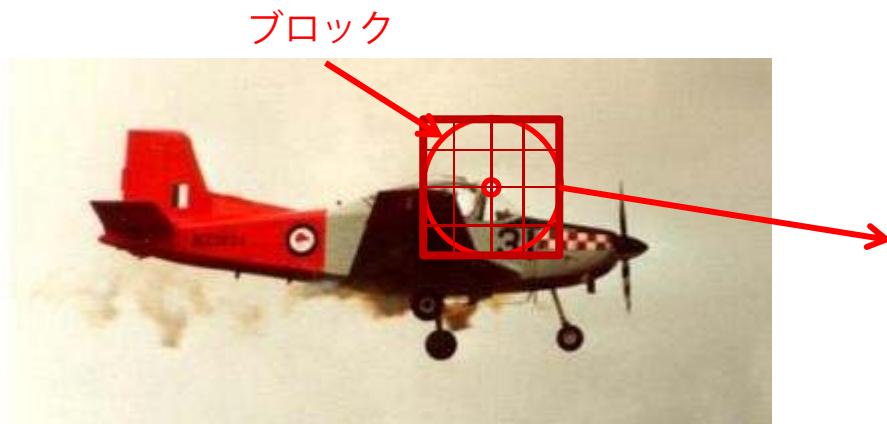
4. 得られたベクトルを正規化する

$$\mathbf{f}' = \frac{\mathbf{f}}{\|\mathbf{f}\|}$$

局所領域での正規化を行っているので
照明変化に頑健になる

勾配ヒストグラムの重み付け

- ・注目ピクセルに近いほど重要視する

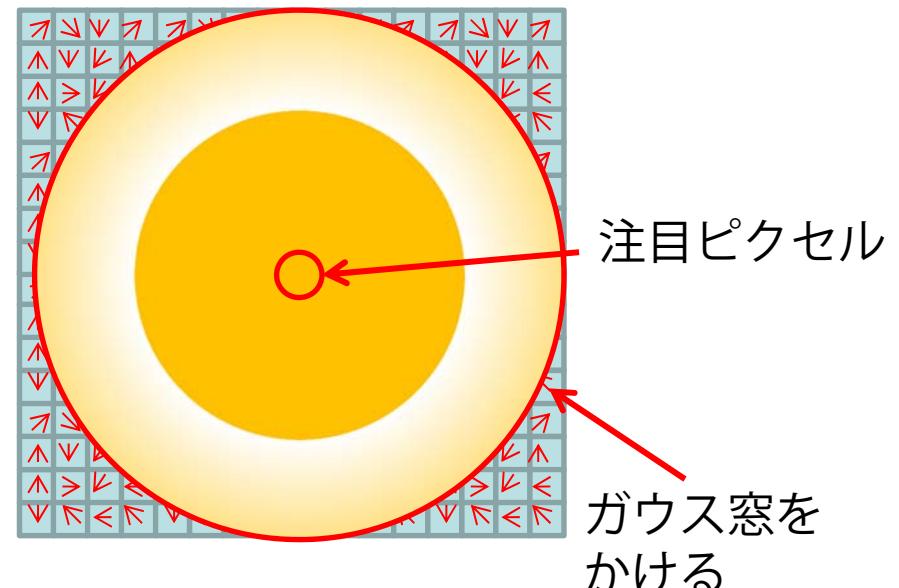


2次元ガウス分布

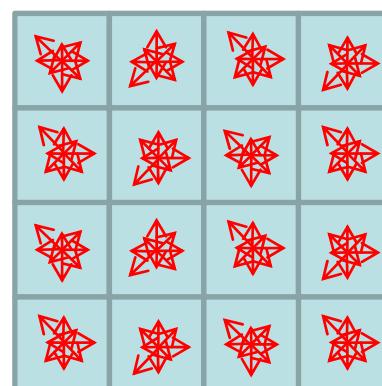
$$G(x, y, \sigma) = \frac{1}{2\pi\sigma^2} \exp\left(-\frac{x^2 + y^2}{2\sigma^2}\right)$$

ガウス分布による勾配強度の重み付け

$$m'(x, y) = G(x, y, \sigma) * m(x, y)$$



新たに得た勾配強度 m' を利用して同様の手続きでSIFT descriptorを得る

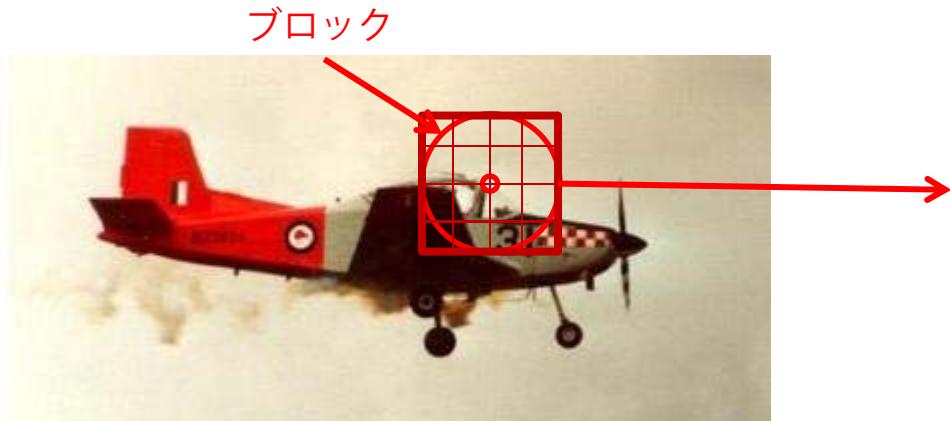


$$\mathbf{f}^T = (\mathbf{f}_1^T \ \mathbf{f}_2^T \ \cdots \ \mathbf{f}_{16}^T)$$

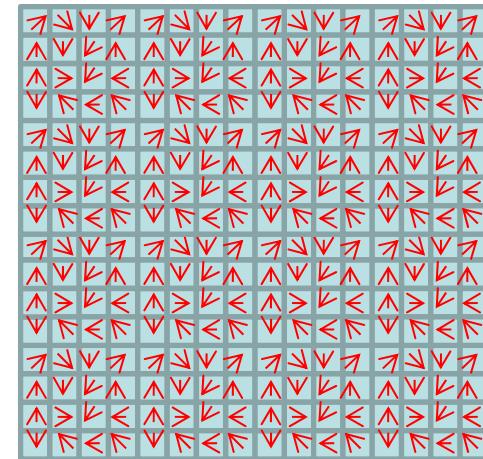
$$\mathbf{f}' = \frac{\mathbf{f}}{\|\mathbf{f}\|}$$

回転不变

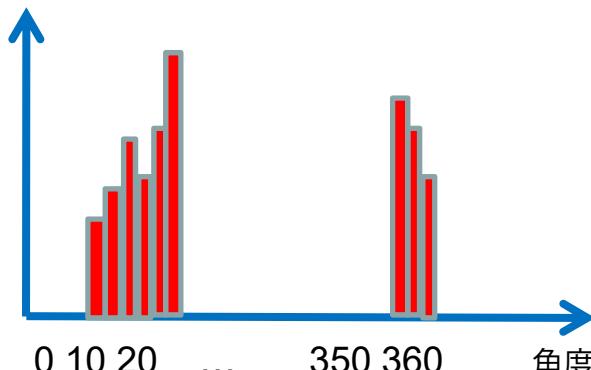
- 回転方向の正規化を行う
 - 注目領域の支配的な輝度勾配を計算



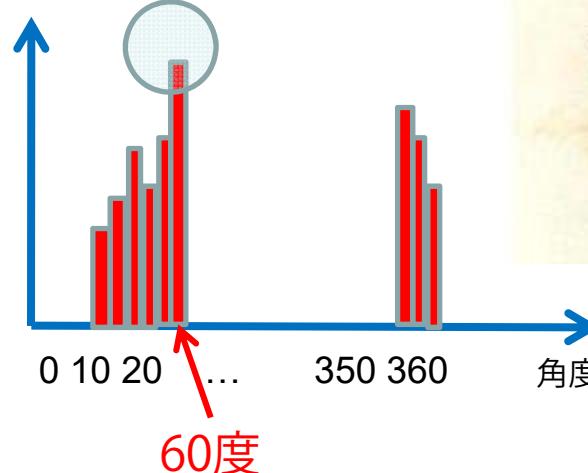
- 注目領域の輝度勾配
(は計算してある)



- 10度毎の注目領域内のヒストグラムを作成



- 支配的な角度を発見

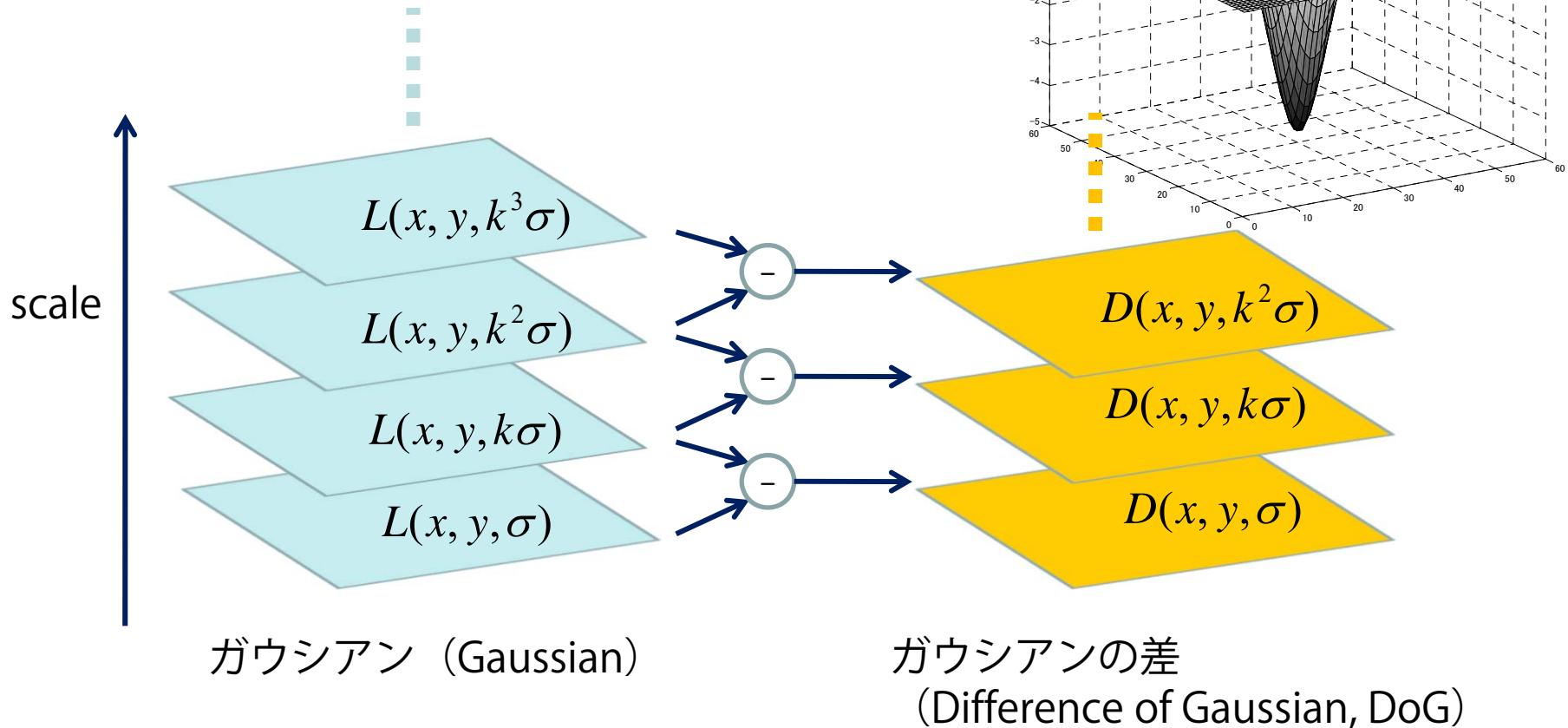


- 支配的な角度に着目領域を回転



- 回転した領域に同様の手続きでSIFT descriptorを計算

特徴点の検出 DoG



$$L(x, y, \sigma) = G(x, y, \sigma) * I(x, y)$$

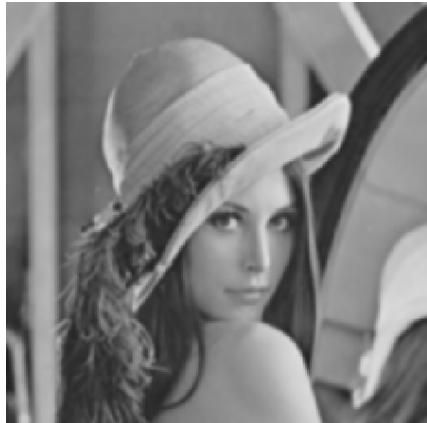
$$G(x, y, \sigma) = \frac{1}{2\pi\sigma^2} \exp\left(-\frac{x^2 + y^2}{2\sigma^2}\right)$$

$$\begin{aligned} D(x, y, \sigma) &= (G(x, y, k\sigma) - G(x, y, \sigma)) * I(x, y) \\ &= L(x, y, k\sigma) - L(x, y, \sigma) \end{aligned}$$

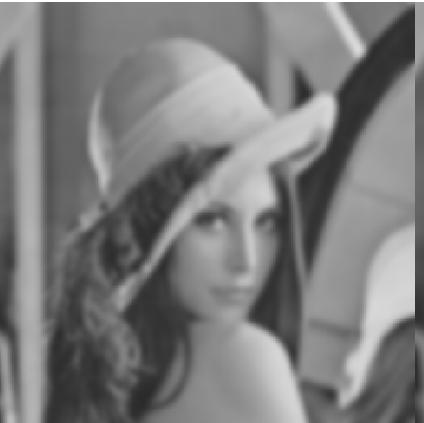
エッジやコーナーを強調する

DoG

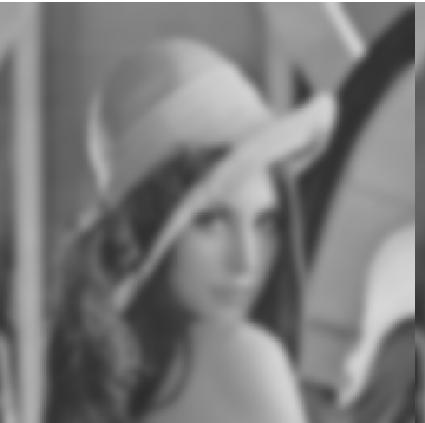
$$L(x, y, \sigma)$$



$$L(x, y, k\sigma)$$



$$L(x, y, k^2\sigma)$$



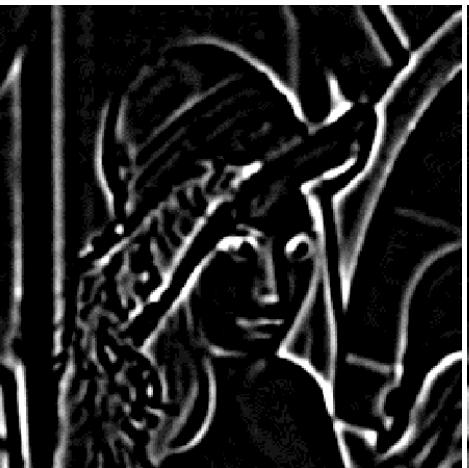
$$L(x, y, k^3\sigma)$$



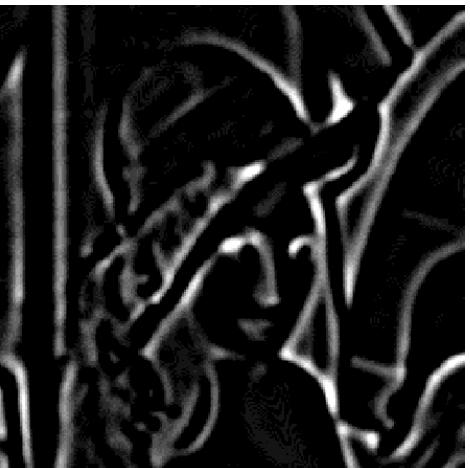
$$D(x, y, \sigma)$$



$$D(x, y, k\sigma)$$

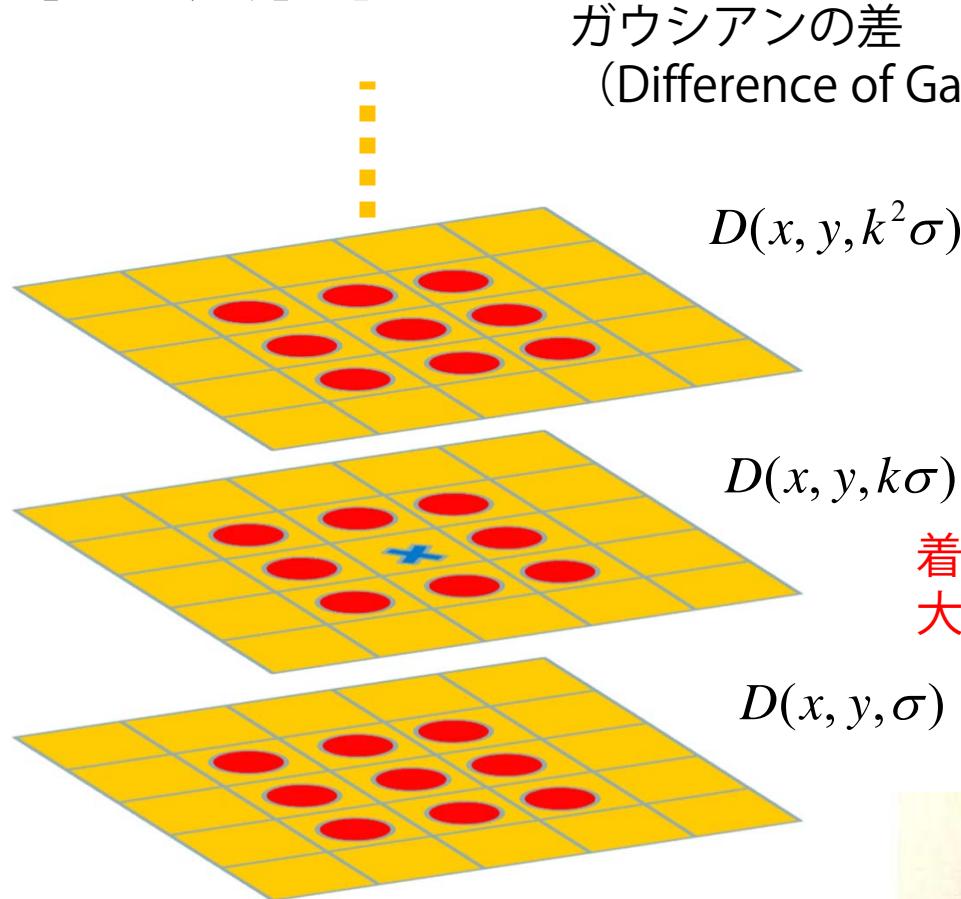


$$D(x, y, k^2\sigma)$$



局所極値検出

↑
scale



ガウシアンの差
(Difference of Gaussian, DoG)

$$D(x, y, k^2\sigma)$$

$$D(x, y, k\sigma)$$

$$D(x, y, \sigma)$$

着目点周りの記述領域の
大きさの決定に σ を利用



- $3 \times 3 \times 3$ の近傍における全ての極値を選択
- 26個の全ての近傍画素よりも値が大きい、もしくは小さい場合にXが極値として選択される → キーポイント
- 極値が発見されたスケール σ も保持する → スケール不变

正確なキーポイントの位置推定

- DoGから検出されたキーポイントの周辺ピクセルを利用してサブピクセルのキーポイント位置推定を行う
 - キーポイントのマッチングと安定性が向上することが報告されている

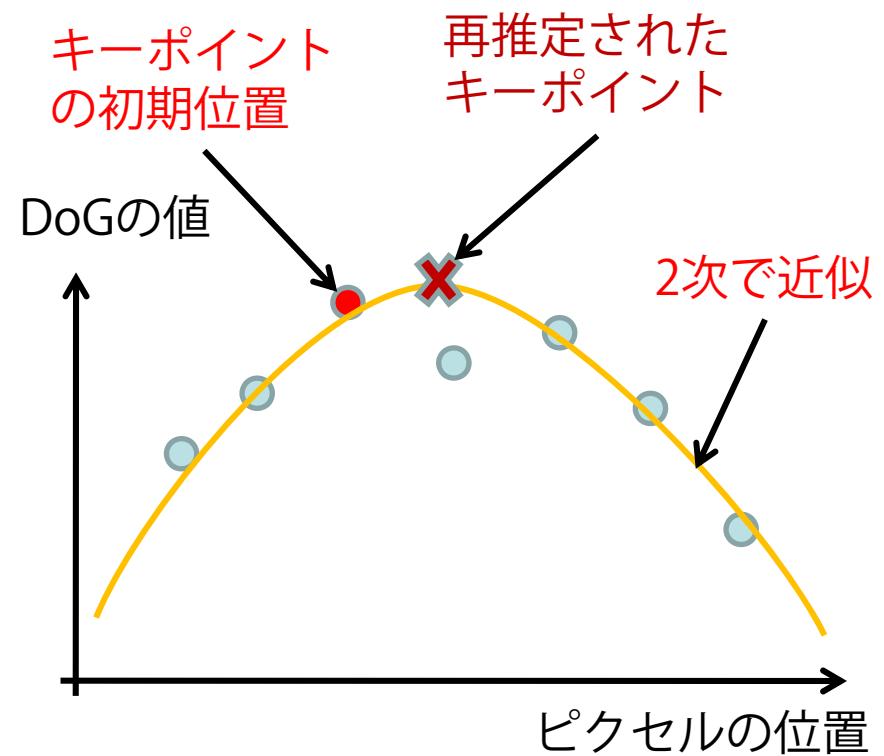
DoGの関数をテーラー展開

$$D(\mathbf{x}) = D + \frac{\partial D^T}{\partial \mathbf{x}} \mathbf{x} + \frac{1}{2} \mathbf{x}^T \frac{\partial^2 D}{\partial \mathbf{x}^2} \mathbf{x}$$
$$\mathbf{x} = (x, y, \sigma)^T$$

停留点を求める

$$\frac{\partial D(\mathbf{x})}{\partial \mathbf{x}} = 0 \Rightarrow \hat{\mathbf{x}} = -\frac{\partial^2 D}{\partial \mathbf{x}^2}^{-1} \frac{\partial D}{\partial \mathbf{x}}$$

近接するサンプル点の
差分で求める



キーポイントのフィルタリング

- キーポイント検出によりピクセル数よりも少数の点に絞り込まれた
- しかしながら、まだ点の数が多い
- 信頼性の低いキーポイントを削除する
 - コントラストが低い点
 - エッジ上の点

低コントラストの点の削除

再推定されたキーポイント

$$\hat{\mathbf{x}} = -\frac{\partial^2 D}{\partial \mathbf{x}^2}^{-1} \frac{\partial D}{\partial \mathbf{x}} \quad \cdots(1)$$

DoGのテーラー展開

$$D(\mathbf{x}) = D + \frac{\partial D^T}{\partial \mathbf{x}} \mathbf{x} + \frac{1}{2} \mathbf{x}^T \frac{\partial^2 D^T}{\partial \mathbf{x}^2} \mathbf{x} \quad \cdots(2)$$

(1)を(2)に代入

$$D(\hat{\mathbf{x}}) = D + \frac{1}{2} \frac{\partial D^T}{\partial \mathbf{x}} \hat{\mathbf{x}}$$

閾値より低ければコントラストが低いと判断して削除

エッジ上の点の削除

- Harris Corner Detectorと類似した手法でエッジ上の点を削除

DoGのHessian Matrix

$$H = \begin{bmatrix} D_{xx} & D_{xy} \\ D_{yx} & D_{yy} \end{bmatrix}$$

Hessian Matrixの対角化

$$H = U \begin{pmatrix} \alpha & 0 \\ 0 & \beta \end{pmatrix} U^{-1} \quad U = (\mathbf{u}_\alpha \ \mathbf{u}_\beta) \quad H\mathbf{u}_\alpha = \alpha \mathbf{u}_\alpha \quad \alpha \geq \beta \quad H\mathbf{u}_\beta = \beta \mathbf{u}_\beta$$

aとβの差が大きければエッジ！

明示的な固有値計算の回避

α と β の比率 r を導入

rが大きければエッジ！

トレースとデターミナント
と固有値との関係

$$\text{Tr}(H) = D_{xx} + D_{yy} = \alpha + \beta$$

$$\text{Det}(H) = D_{xx}D_{yy} - (D_{xy})^2 = \alpha\beta$$

$$\frac{\text{Tr}(H)^2}{\text{Det}(H)} = \frac{(r+1)^2}{r}$$

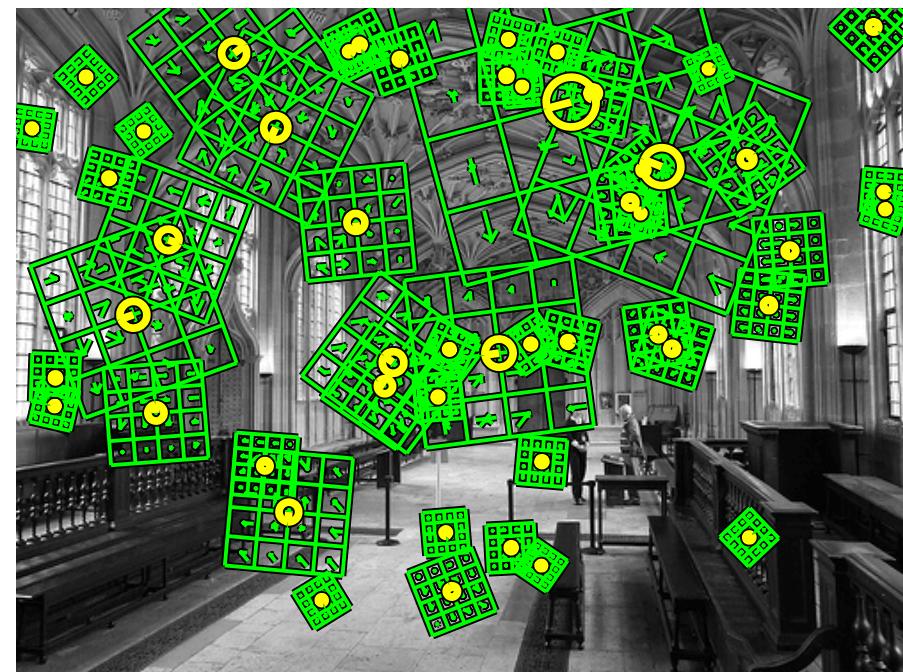
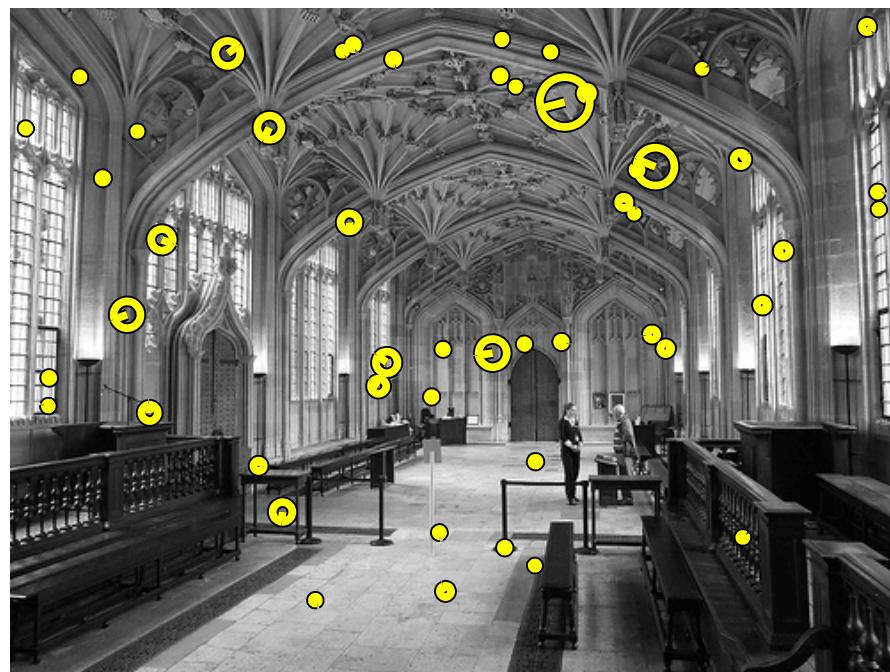
*r=1なら最小
rが大きくなるにつれ大きくなる*

コーナーの検出

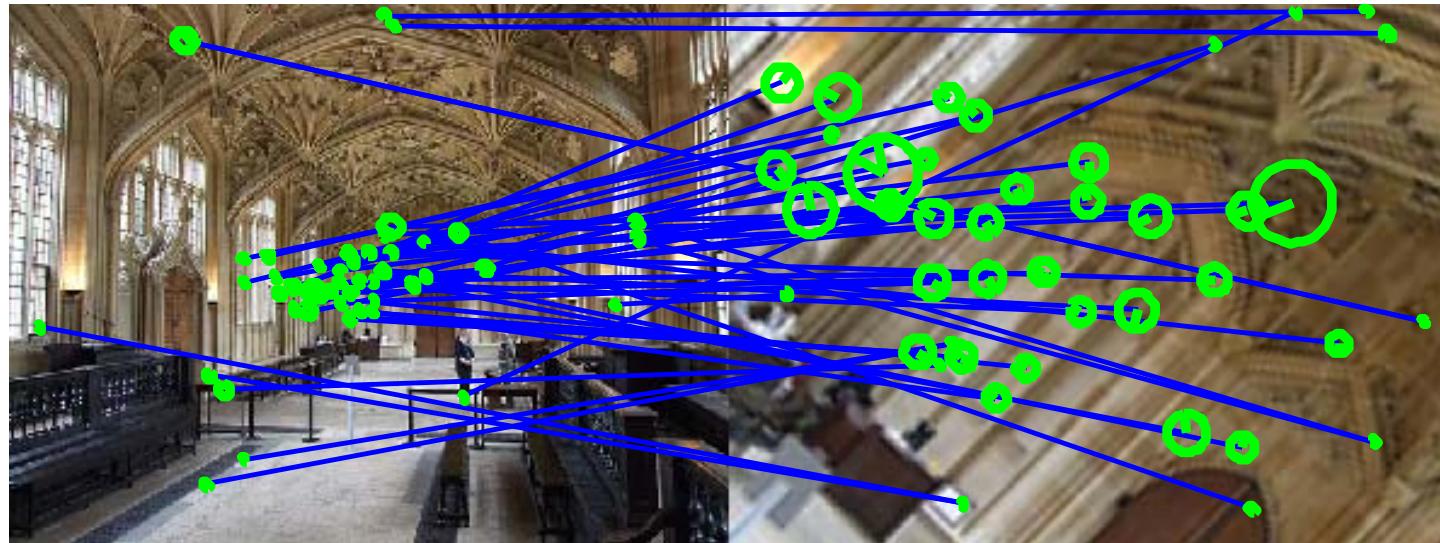
$$\frac{\text{Tr}(H)^2}{\text{Det}(H)} < \frac{(r_{th}+1)^2}{r_{th}}$$

rに閾値をもうけ、この値より小さければコーナーとして採用

SIFTの例



SIFTによる物体検出の例



SURF

- Herbert Bay, Tinne Tuytelaars, and Luc Van Gool.
SURF: Speeded Up Robust Features. In ECCV, 2006.
- SIFTと似ているが高速化が図られている
 - 積分画像の利用！
- OpenCV2.0に実装済み
- SURFの構成
 - Detector: Hessian matrix measure
 - Descriptor: Histograms of oriented gradient

SURF

1. 着目するピクセルを選択する
 - Hessianの近似



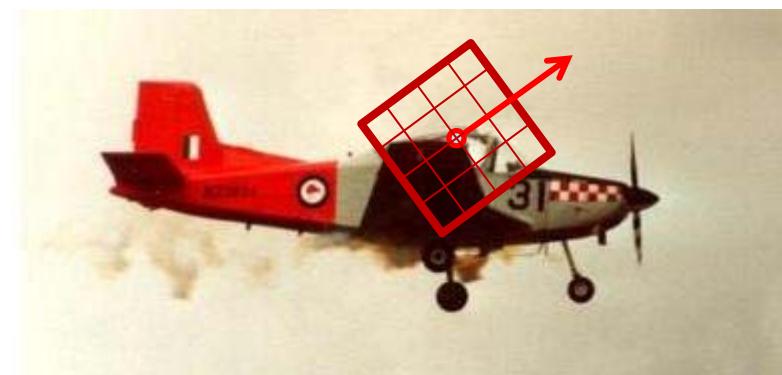
3. 正方形領域内の勾配を計算し支
配的な勾配を求める



2. 正方形領域を決定する
スケール不変



4. 領域を回転させ、正方形領域を
均等に16分割し、勾配ヒストグ
ラムを計算



特徴点検出

Hessian Matrix Based Interest Points

- ガウシアンの二次微分の畳み込みによるHessian Matrixの計算

$$\frac{\partial^2}{\partial x^2} g(\sigma) * I(x)$$
$$\frac{\partial^2}{\partial x \partial y} g(\sigma) * I(x)$$
$$\frac{\partial^2}{\partial y^2} g(\sigma) * I(x)$$
$$H(x, \sigma) = \begin{pmatrix} L_{xx}(x, \sigma) & L_{xy}(x, \sigma) \\ L_{yx}(x, \sigma) & L_{yy}(x, \sigma) \end{pmatrix}$$

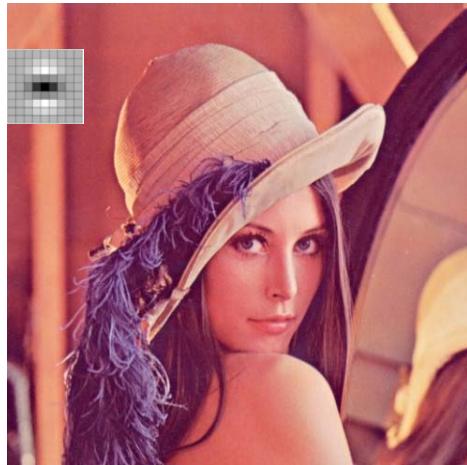
- Hessianの行列式が極大点をとる場所を特徴点と考える。

$$\det(H(x, \sigma))$$

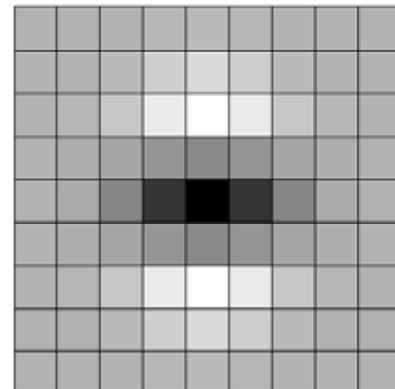


ガウシアンの二次微分の近似

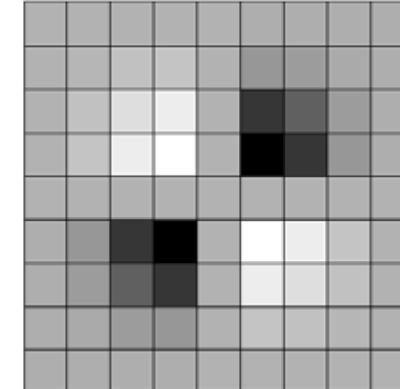
ガウシアンの二次微分のマスク (9x9) $\sigma = 1.2$



$$L_{yy}$$



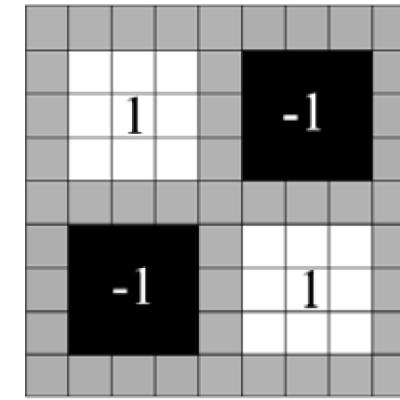
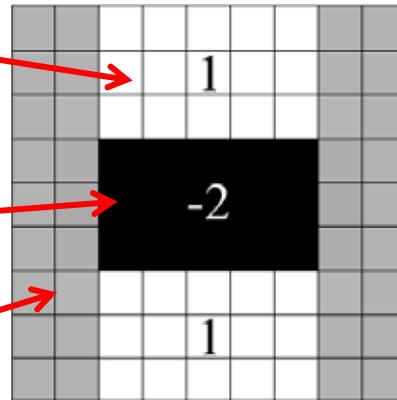
$$L_{xy}$$



Box filterによる近似 (9x9)



白い領域は+1をかけて足す
黒い領域は-2をかけて足す
灰色領域は何もしない

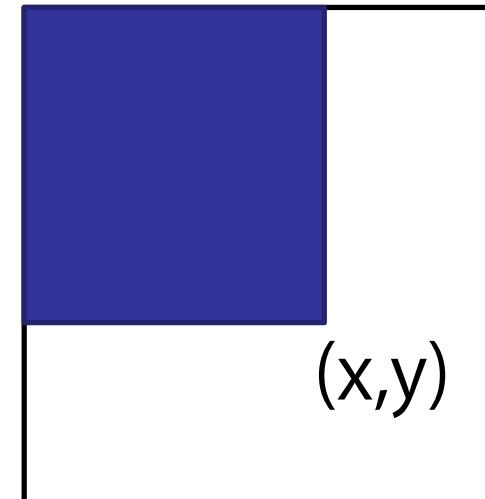


任意のスケールのマスクも積分画像により高速に計算可能！

積分画像

- 積分画像は原点(0,0)と(x,y)を対角の頂点とする長方形で囲まれる領域内のピクセル値の和
- 一度のパスで高速計算可能
- 行のピクセル値の和
 - $s(x, y) = s(x-1, y) + i(x, y)$
- 積分画像
 - $ii(x, y) = ii(x, y-1) + s(x, y)$

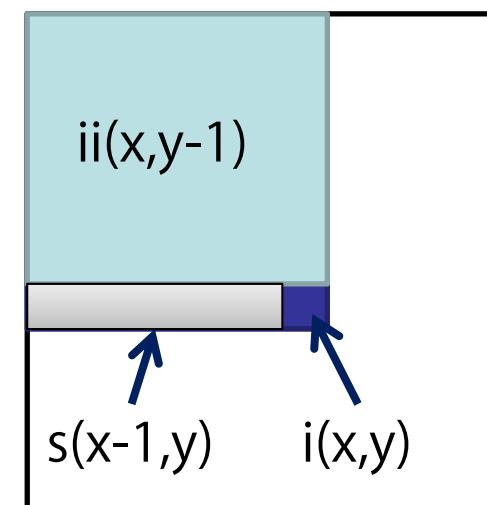
(0,0)



$ii(x, y-1)$

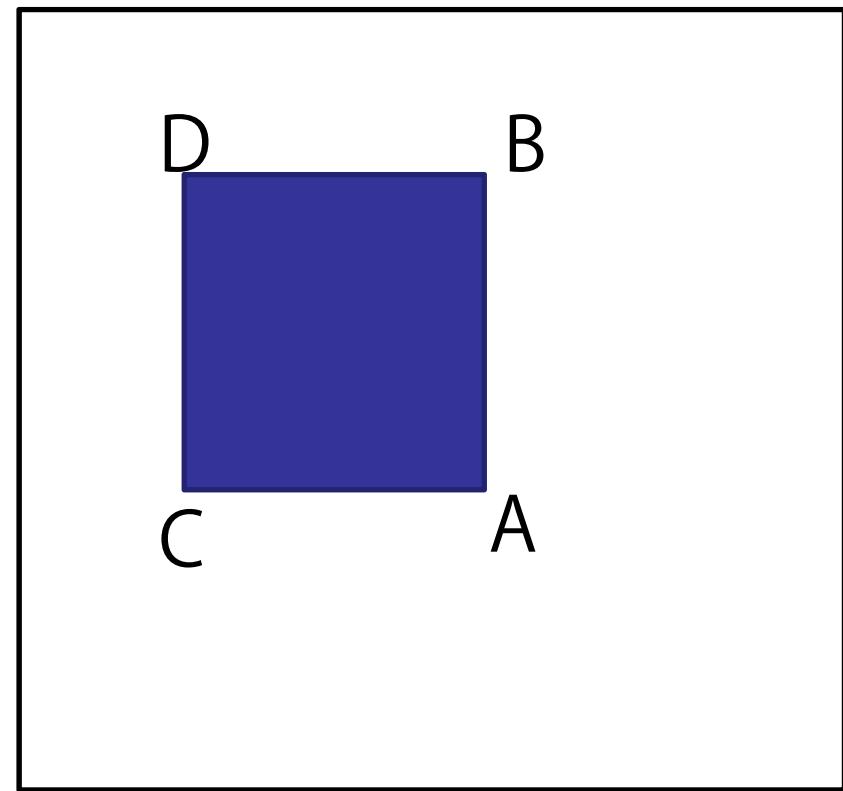
$s(x-1, y)$

$i(x, y)$



長方形領域内の和の計算

- A, B, C, Dを積分画像の値とする
- 長方形領域内の画素値の合計
 - $A-B-C+D$
- 3加算だけで任意の長方形領域内の画素値の和を計算可能



Hessianの行列式の近似

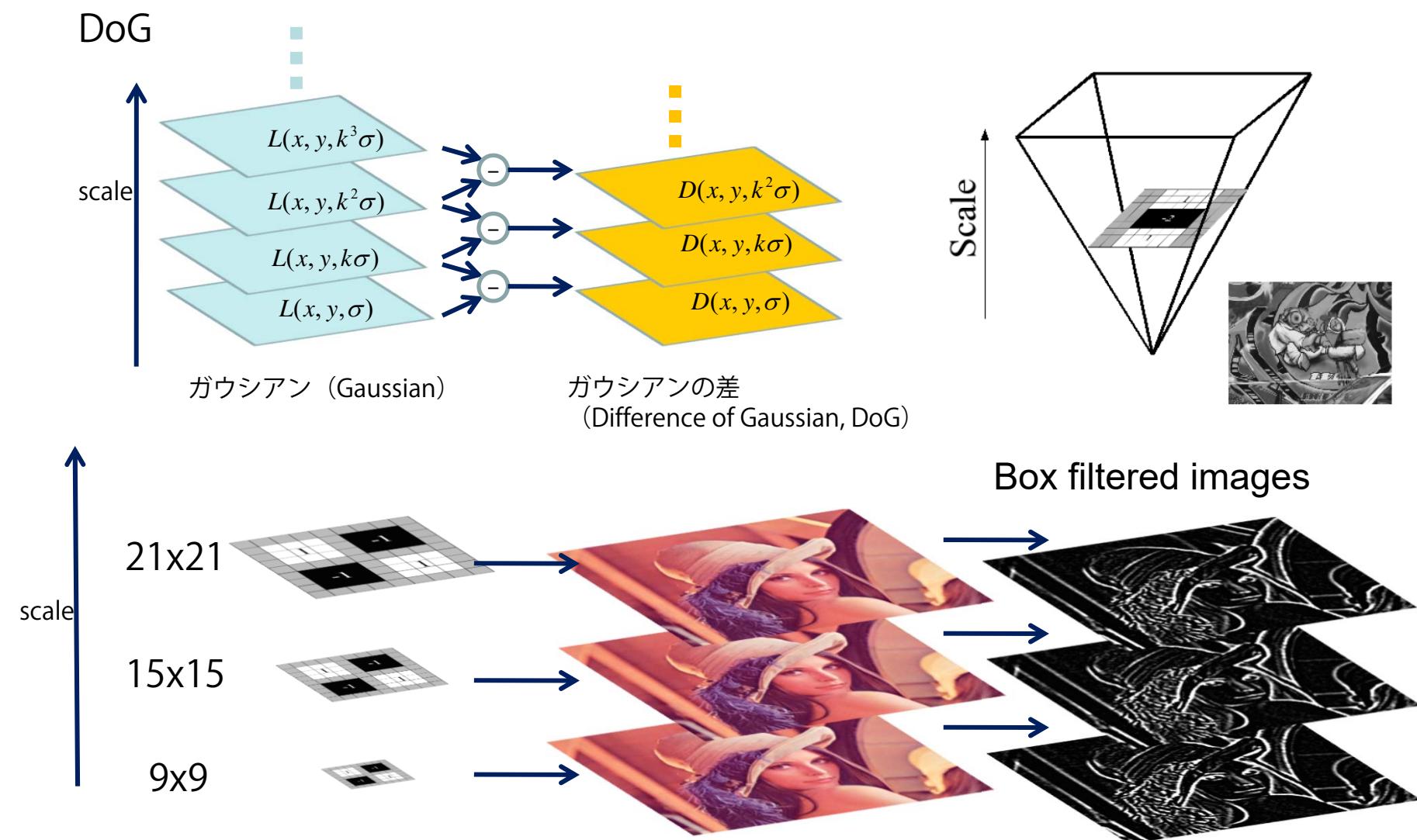
- ガウシアンの二次微分とBox filterとの誤差の補正

スケール依存だが、固定値を利用

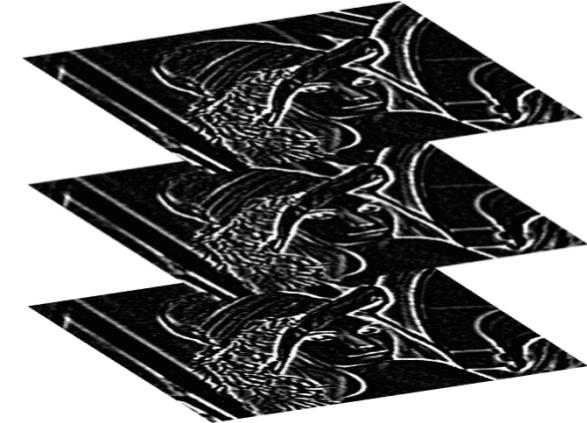
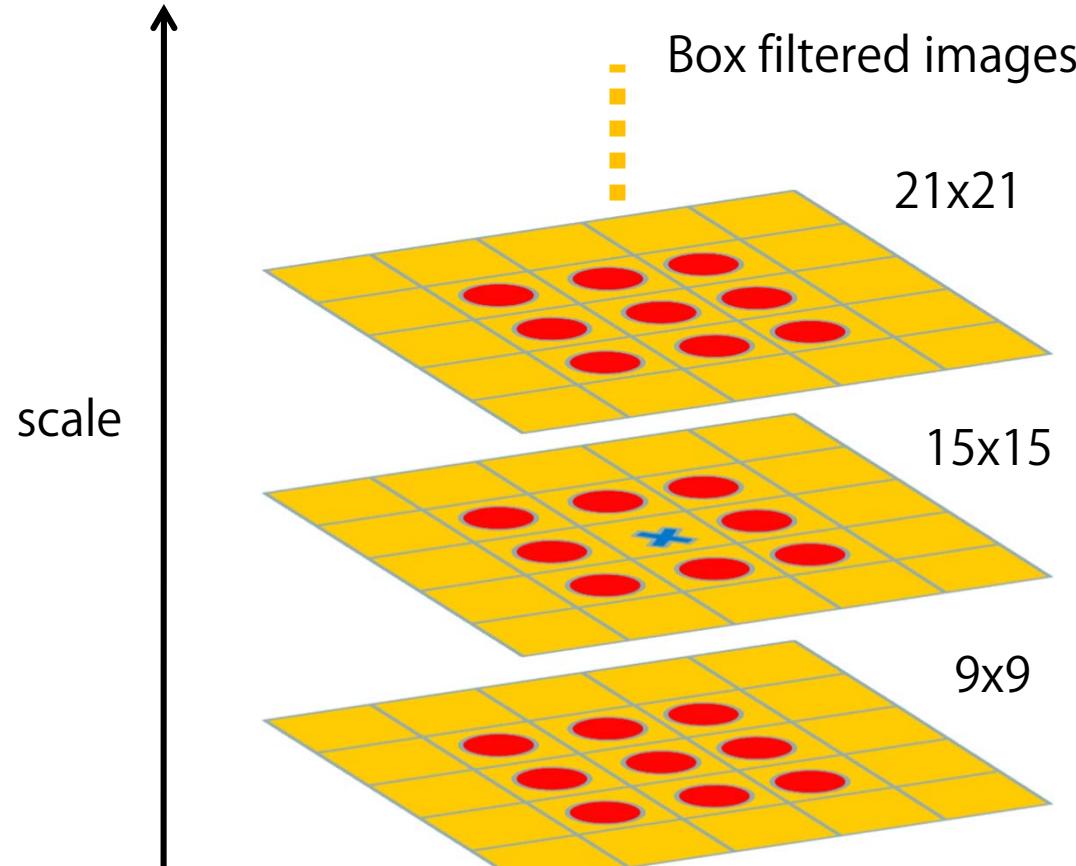
$$\det(H_{\text{approx}}) = D_{xx}D_{yy} - \overset{\leftarrow}{(wD_{xy})^2}$$

行列式が極大点をとる場所を特徴点と考える

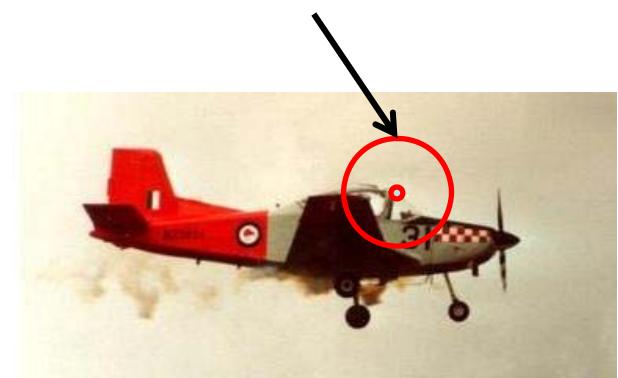
スケールスペース表現



局所極値検出



着目点周りの記述領域の大きさの決定にスケールを利用



- $3 \times 3 \times 3$ の近傍における全ての極値を選択
- 26個の全ての近傍画素よりも値が大きい、もしくは小さい場合にXが極値として選択される
- 極値が発見されたスケール s も保持する → スケール不变

特徴点の検出結果

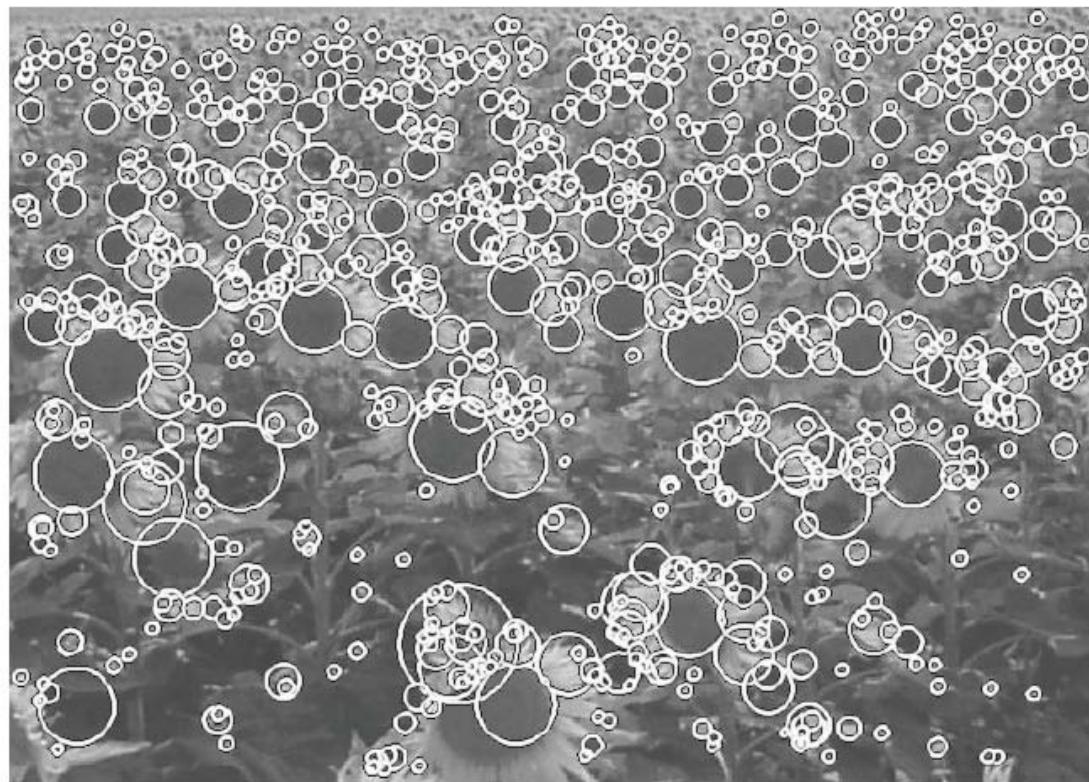
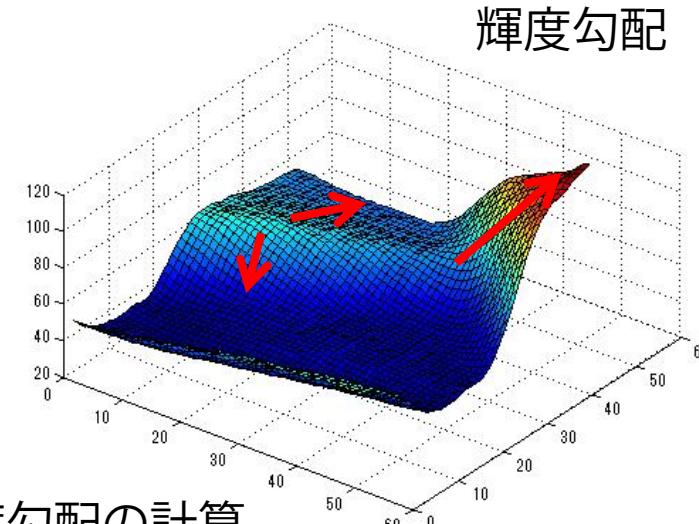
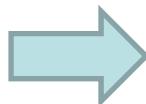
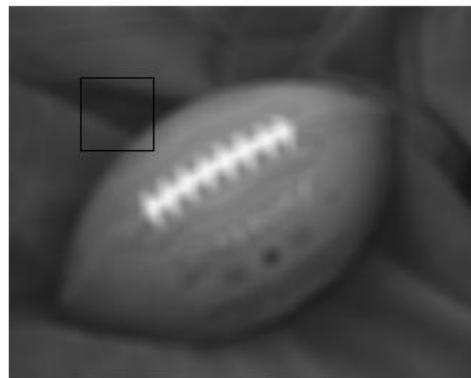
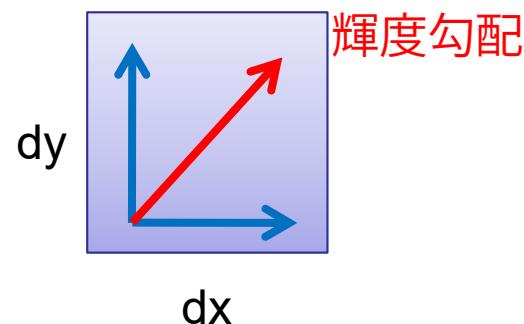


Fig. 8. Detected interest points for a Sunflower field. This kind of scenes shows the nature of the features obtained using Hessian-based detectors.

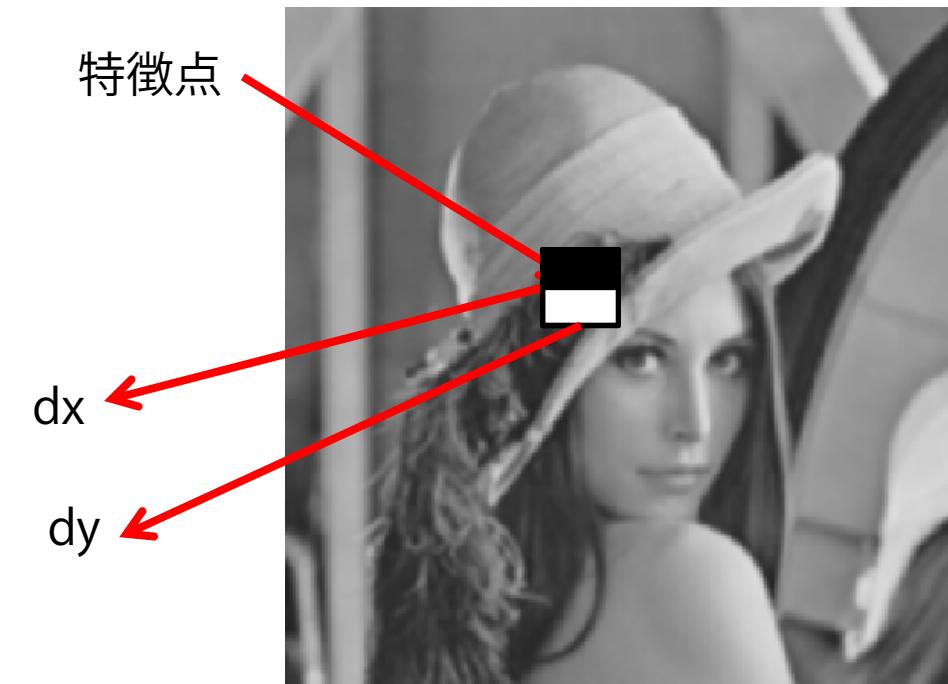
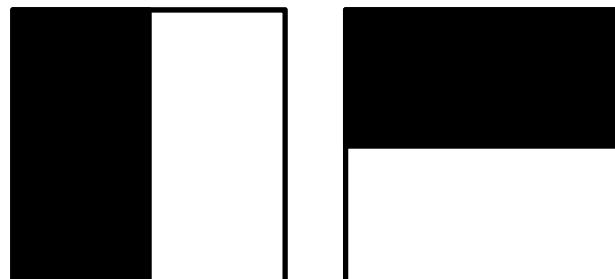
勾配の計算



局所輝度勾配の計算



Haar Wavelet Filter

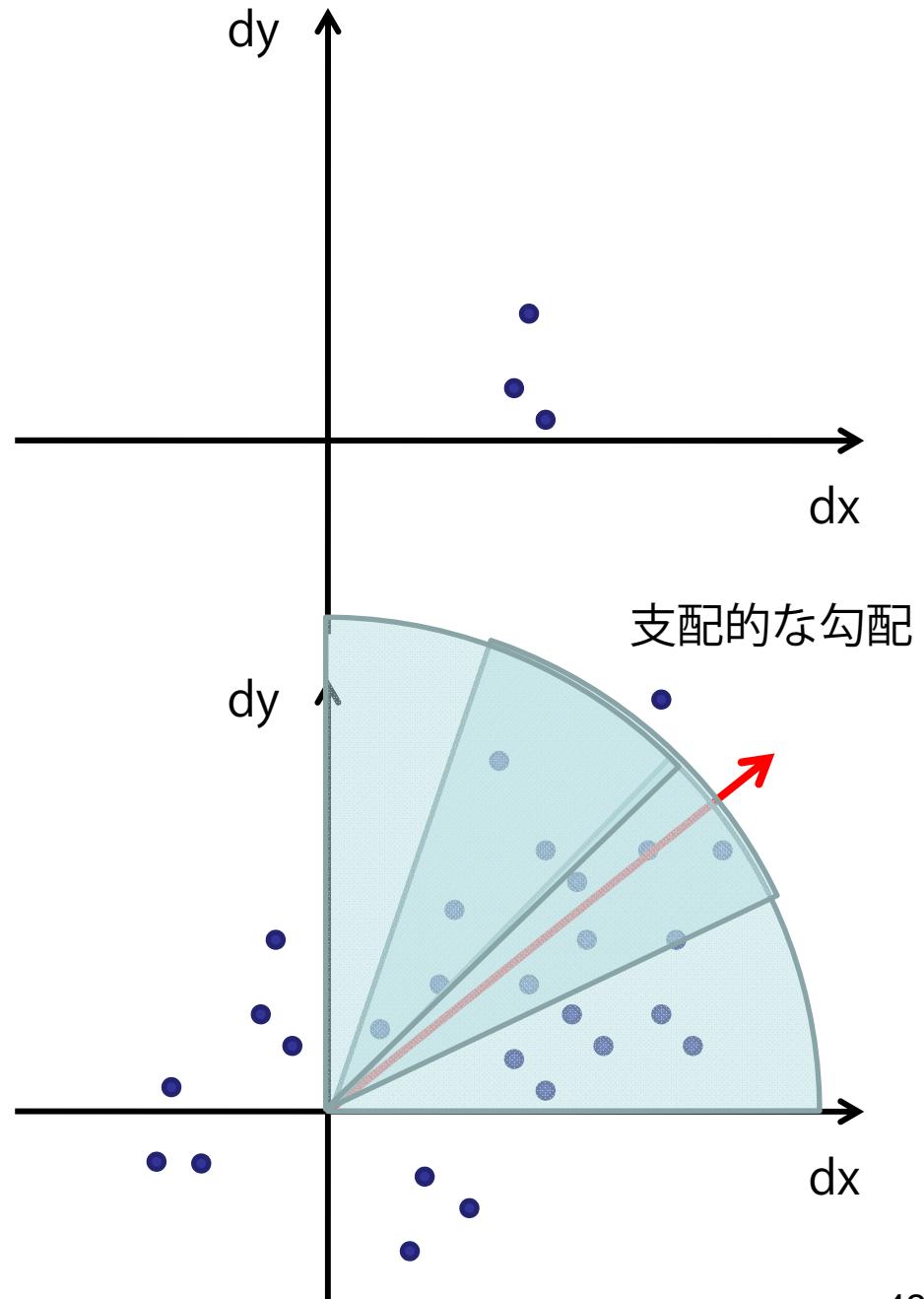


- ・ 着目領域内の全てのピクセルで勾配を計算
- ・ 積分画像により計算

支配的な勾配



特徴点



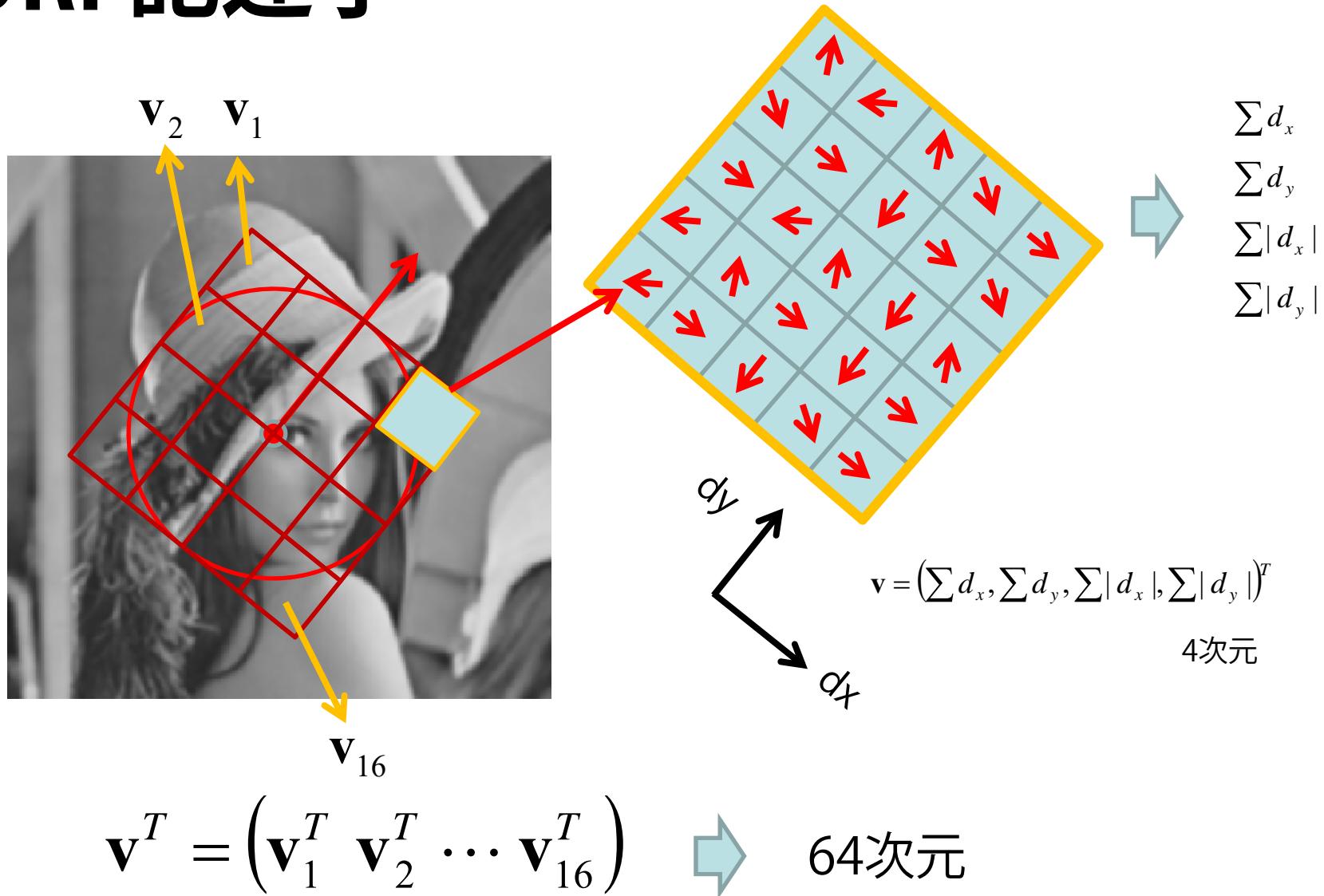
- Wavelet responseは特徴点を中心としたガウシアン ($\sigma=2s$) で重み付け
- 円弧の領域 $\pi/3$ をずらしながらその領域に含まれる点を加算しベクトルを計算
- 最も大きなノルムをもつベクトルをその領域の支配的な勾配とする

得られたスケールと支配的な勾配



Fig. 11. Detail of the Graffiti scene showing the size of the oriented descriptor window at different scales.

SURF記述子



SURF記述子の特性

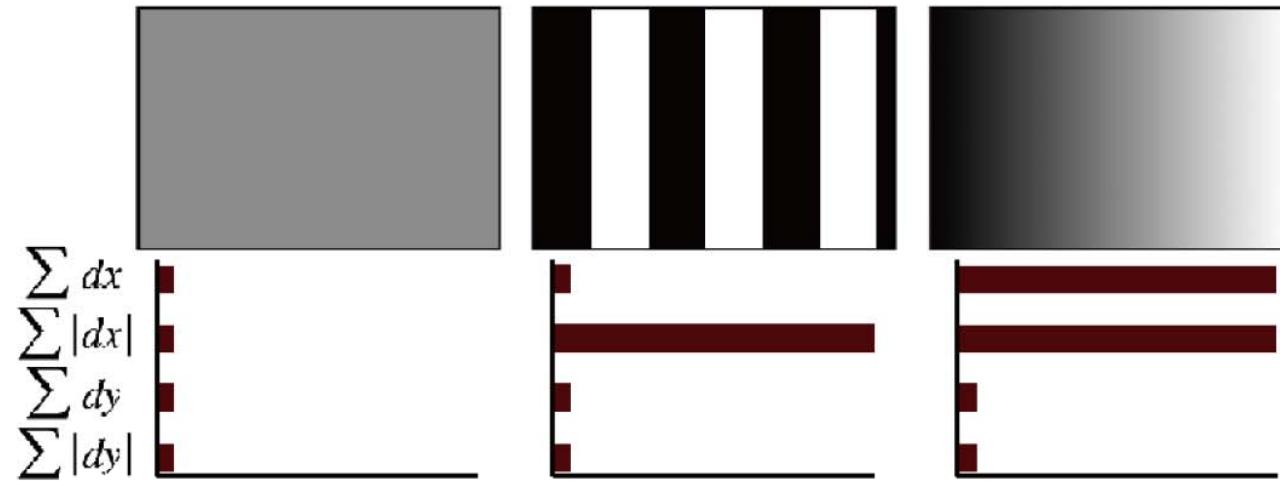


Fig. 13. The descriptor entries of a sub-region represent the nature of the underlying intensity pattern. Left: In case of a homogeneous region, all values are relatively low. Middle: In presence of frequencies in x direction, the value of $\sum |d_x|$ is high, but all others remain low. If the intensity is gradually increasing in x direction, both values $\sum d_x$ and $\sum |d_x|$ are high.

SURF記述子の特性

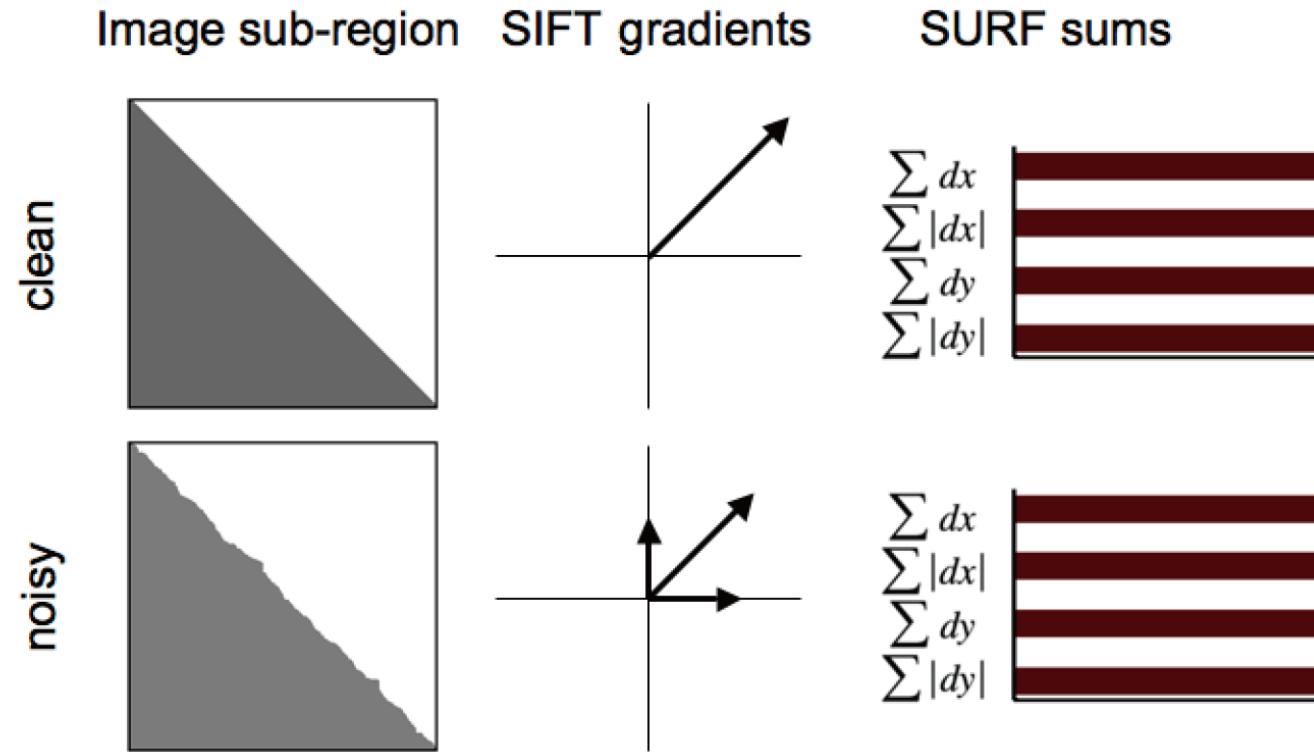


Fig. 14. Due to the global integration of SURF's descriptor, it stays more robust to various image perturbations than the more locally operating SIFT descriptor.

SURFの性能

detector	threshold	nb of points	comp. time (ms)
FH-15	60000	1813	160
FH-9	50000	1411	70
Hessian-Laplace	1000	1979	700
Harris-Laplace	2500	1664	2100
DoG	default	1520	400

Table 1

Thresholds, number of detected points and calculation time for the detectors in our comparison. (First image of Graffiti scene, 800×640)

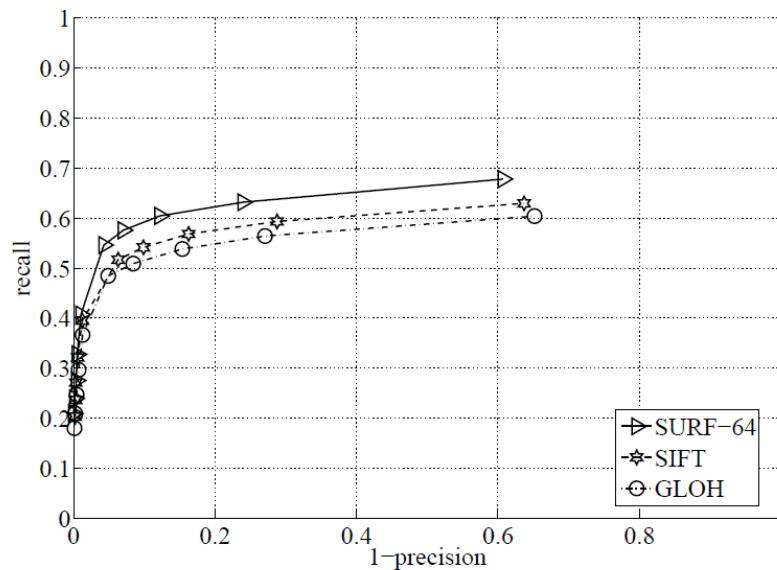


Fig. 19. Recall-precision for nearest neighbor ratio matching for different description schemes, evaluated on SURF keypoints. Figures averaged over 8 image pairs of Mikolajczyk's database.

FAST Corner Detector

- 高速, 高性能なコーナー検出器
- Edward Rosten and Tom Drummond.
Machine Learning for High-speed Corner
Detection. In ECCV, 2006.
- PTAMなどでも利用されている.

Parallel Tracking and Mapping for Small AR Workspaces

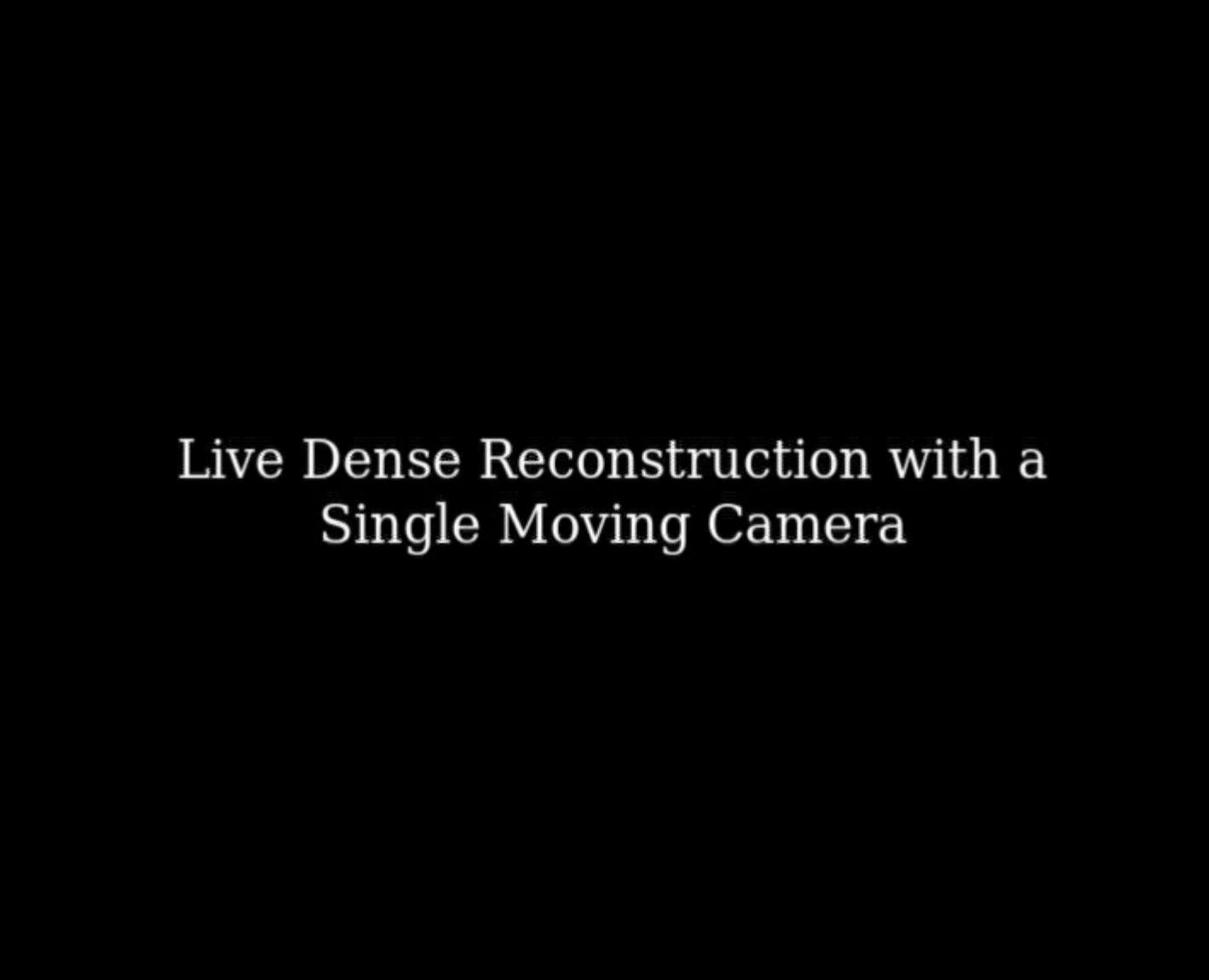
Parallel Tracking and Mapping
for Small AR Workspaces

Extra video results made for
ISMAR 2007 conference

Georg Klein and David Murray
Active Vision Laboratory
University of Oxford

Mono SLAM

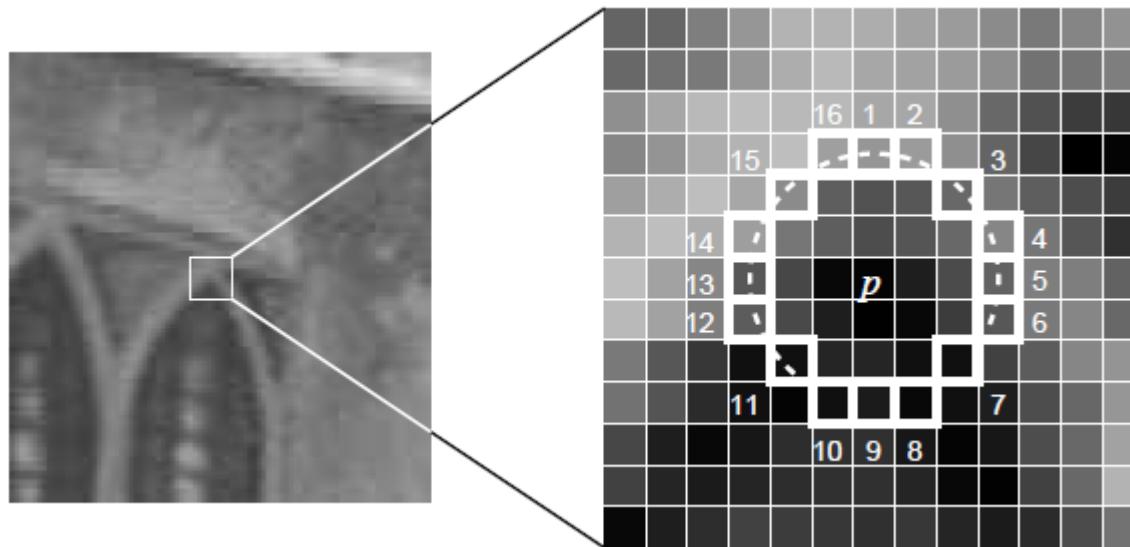
- Richard A. Newcombe and Andrew J. Davison. Live Dense Reconstruction with a Single Moving Camera. In CVPR, 2010.



Live Dense Reconstruction with a
Single Moving Camera

The Segment Test Algorithm

- ピクセル p を中心として半径 r 上にあるN個以上の隣接するピクセルが、中心のピクセル p の輝度よりも閾値 t 以上明るい、もしくは t 以下で暗い場合、 p は特徴点である。



Rostenら, ECCV2006ポスターより

$r=3$, $N=12$. ピクセル11-16, 1-6は全て p よりも明るい

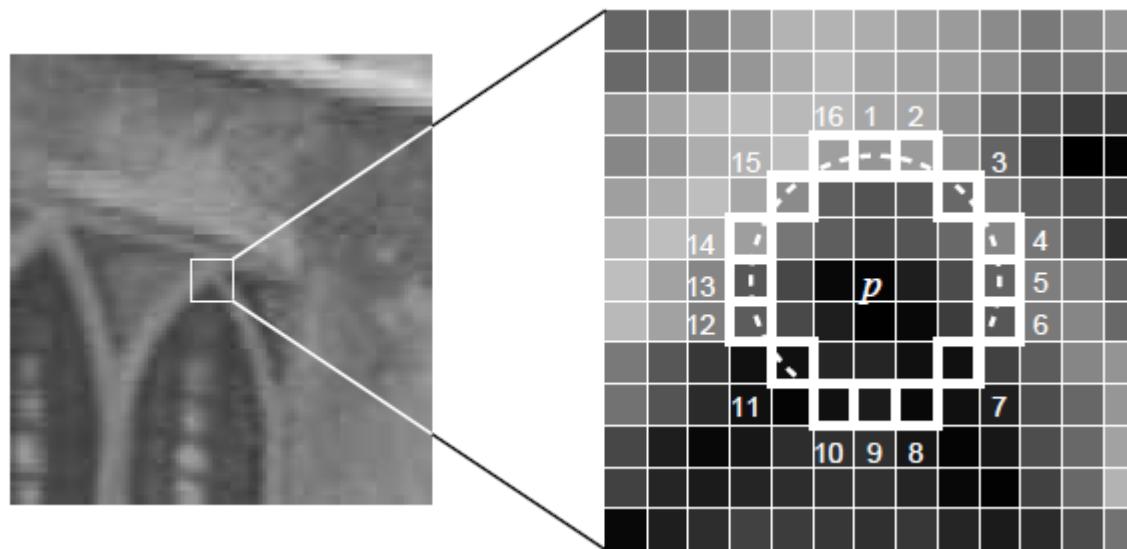


p は特徴点

FAST: Features from Accelerated Segment Test

- $N=12$, $r=3$ の場合を考える
- ピクセル1とピクセル9をテストする
 - ピクセル p と同じような輝度の場合, p は特徴点ではない
- ピクセル5とピクセル13をテストする
 - ピクセル p と同じような輝度の場合, p は特徴点ではない

Rostenら, ECCV2006ポスターより



2回テストするだけで p が特徴点でないことが分かる！

問題点 1 : $N=12$ は最良の選択であるか不明

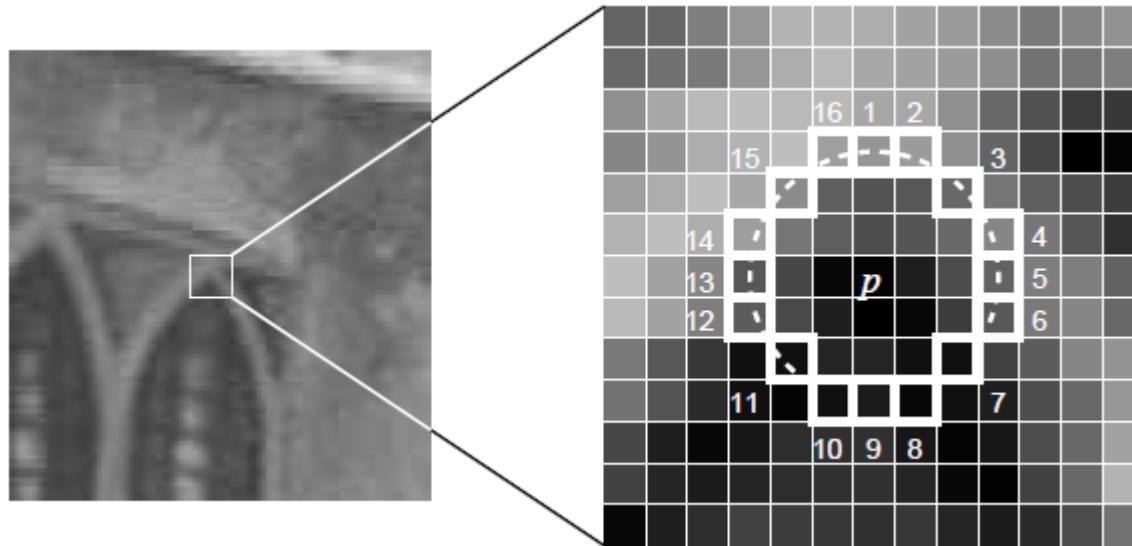
問題点 2 : テストの順番が最適であるか不明

問題点 3 : 複数特徴点が近隣の点で検出される

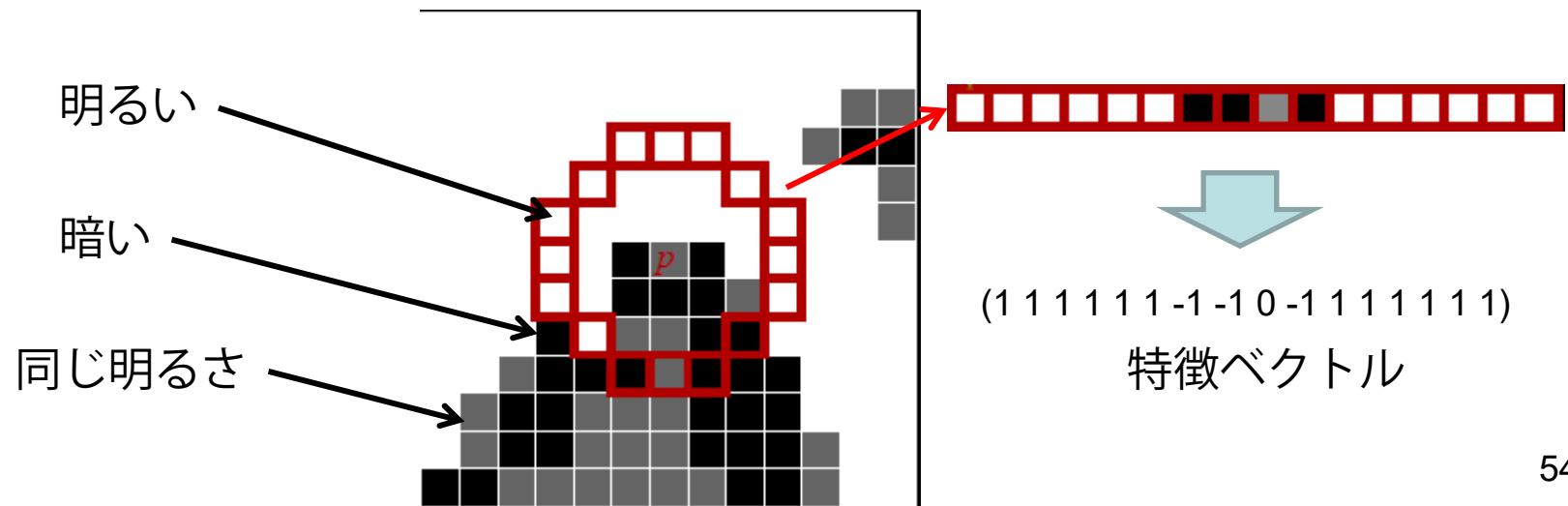
決定木によるFASTの高速化

- ・ 全てのピクセルを「特徴点」もしくは「非特徴点」に分類する識別問題と見なせる
- ・ 半径 $r=3$ の時、全てのピクセルは16個の属性を持つ ($I(x)$: x における輝度, t : 閾値)
 - +1 $I(x) > I(p)+t \rightarrow$ 中心よりも明るい
 - -1 $I(x) < I(p)-t \rightarrow$ 中心よりも暗い
 - 0 上記以外 \rightarrow 中心と同じ明るさ
- ・ ピクセルの特性が与えられると、決定木はそのピクセルを「特徴点」もしくは「非特徴点」に分類可能

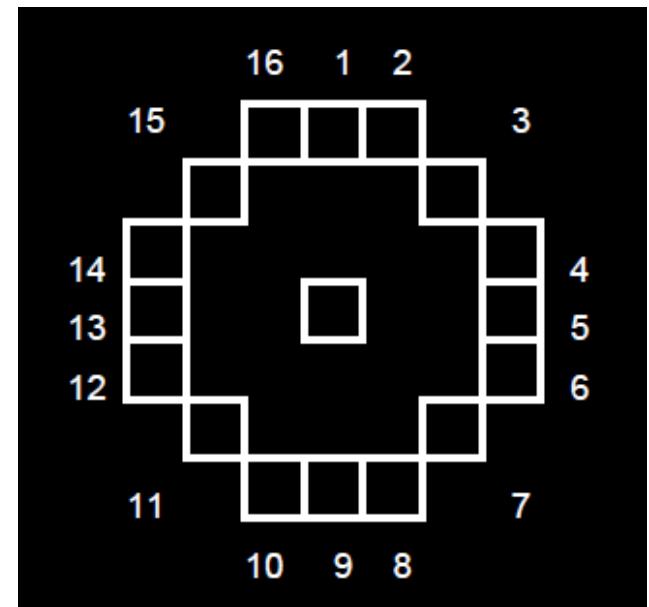
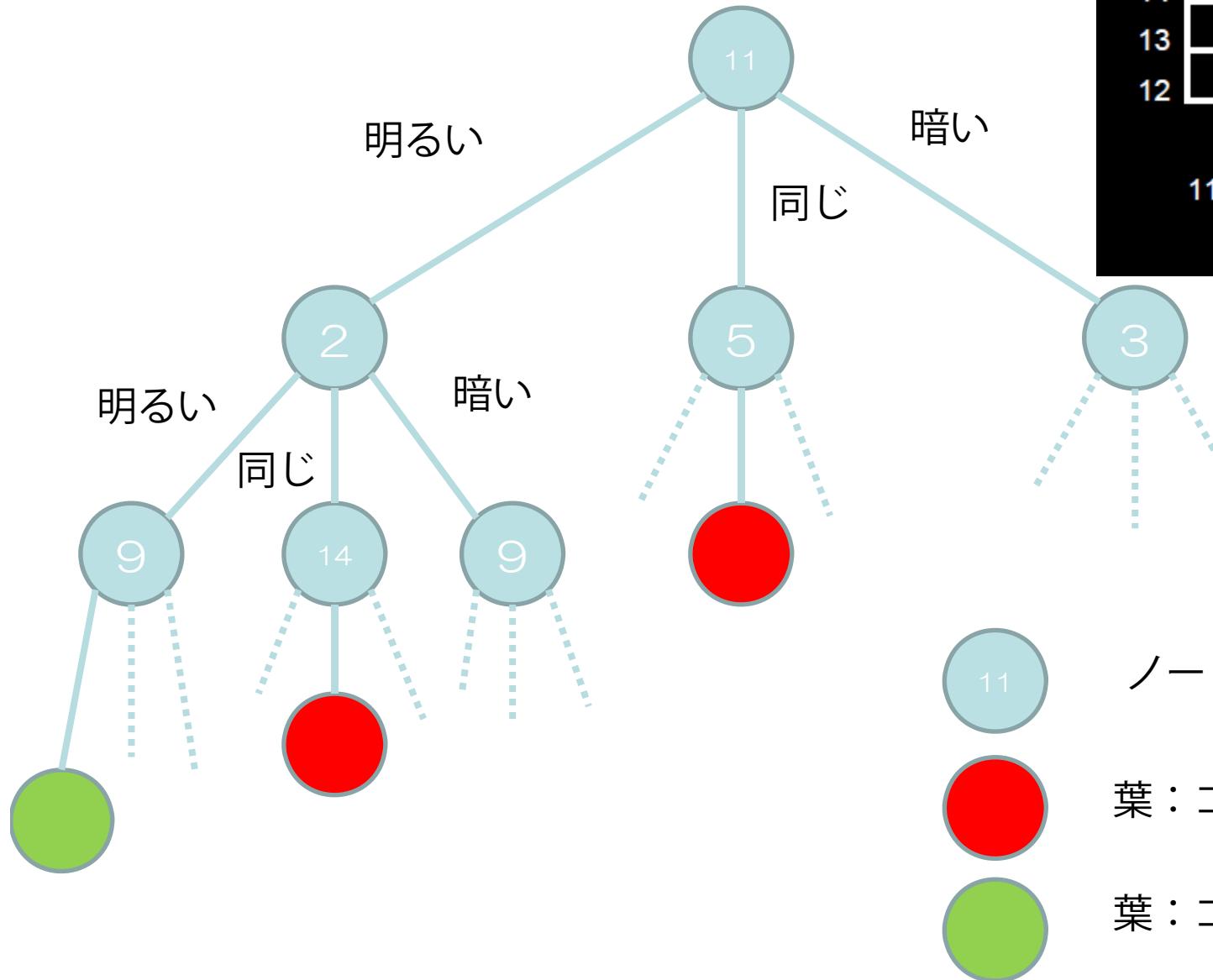
各ピクセルにおける属性表現



+1: $I(x) > I(p)+t \rightarrow$ 中心よりも明るい
-1: $I(x) < I(p)-t \rightarrow$ 中心よりも暗い
0: 上記以外 \rightarrow 中心と同じ明るさ



決定木の例



決定木 (ID3) の学習

学習データの全てのピクセルに対して、特徴ベクトルを計算し、「特徴点」、「非特徴点」とラベルを与える。このデータ集合をCとする。ルートノードを作成して、データ集合Cを割り付ける。

1. データ集合Cを入力とし、データ集合のラベルが全て「特徴点」もしくは「非特徴点」なら、ノードにそのラベルを付与して終了。

2. データ集合Cのエントロピーを計算

$$H(C) = -\frac{c}{c + \bar{c}} \log \frac{c}{c + \bar{c}} - \frac{\bar{c}}{c + \bar{c}} \log \frac{\bar{c}}{c + \bar{c}}$$

3. 変数を一つ選択する

- a. データ集合Cを分割しエントロピーを計算

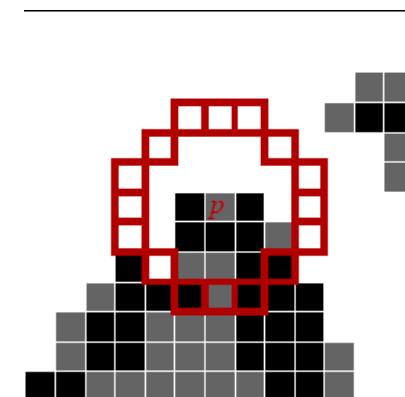
- ・ ピクセルiが明るい場合 → C_b
- ・ ピクセルiは同じくらいの場合 → C_s
- ・ ピクセルiは暗い場合 → C_d

- b. インフォメーションゲインを計算 $H_i = H(C) - \frac{|C_b|}{|C|} H(C_b) - \frac{|C_s|}{|C|} H(C_s) - \frac{|C_d|}{|C|} H(C_d)$

4. インフォメーションゲインが大きい変数をノードが着目する変数とする

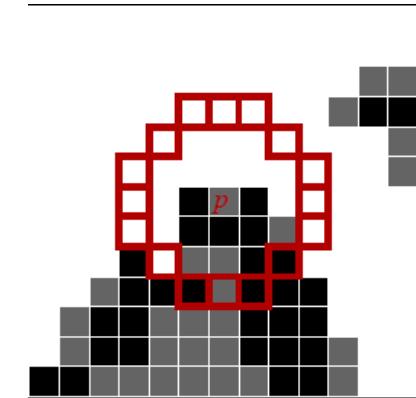
5. 子ノードを作成し、それぞれに C_b, C_s, C_d を所属させる

6. 上記1~5の処理を子ノード毎に再帰的に行う。



Non-maximal suppression

- Segment testでは真のコーナーに隣接するピクセルもコーナーとして検出される
- スコアファンクションVを定義し、隣接するコーナーでVが高いものがあれば、削除する
- 局所最大値の検出
 - Non-maximal suppression

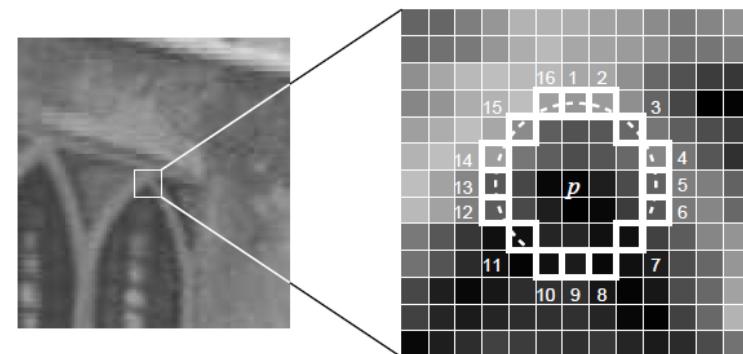


$$V = \max \left(\sum_{x \in S_{bright}} |I_{p \rightarrow x} - I_p| - t, \sum_{x \in S_{dark}} |I_p - I_{p \rightarrow x}| - t \right)$$

明るいと判断された
ピクセルがどれだけ明るいか？

暗いと判断された
ピクセルがどれだけ暗いか？

$$S_{bright} = \{x \mid I_{p \rightarrow x} \geq I_p + t\}$$
$$S_{dark} = \{x \mid I_{p \rightarrow x} \leq I_p - t\}$$



FASTの性能

Detector	Opteron 2.6GHz		Pentium III 850MHz	
	ms	%	ms	%
Fast $n = 9$ (non-max suppression)	1.33	6.65	5.29	26.5
Fast $n = 9$ (raw)	1.08	5.40	4.34	21.7
Fast $n = 12$ (non-max suppression)	1.34	6.70	4.60	23.0
Fast $n = 12$ (raw)	1.17	5.85	4.31	21.5
Original FAST $n = 12$ (non-max suppression)	1.59	7.95	9.60	48.0
Original FAST $n = 12$ (raw)	1.49	7.45	9.25	48.5
Harris	24.0	120	166	830
DoG	60.1	301	345	1280
SUSAN	7.58	37.9	27.5	137.5

Table 1. Timing results for a selection of feature detectors run on fields (768×288) of a PAL video sequence in milliseconds, and as a percentage of the processing budget per frame. Note that since PAL and NTSC, DV and 30Hz VGA (common for webcams) have approximately the same pixel rate, the percentages are widely applicable. Approximately 500 features per field are detected.

Binary Feature

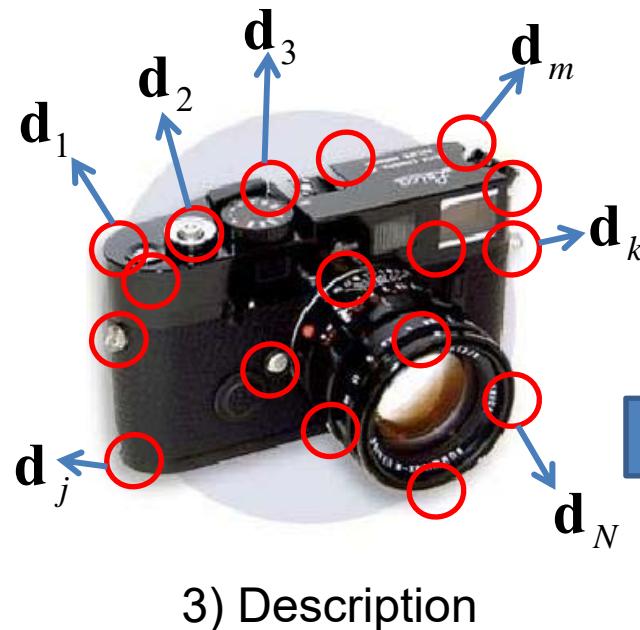
BRIEF

- M. Calonder, V. Lepetit, C. Strecha, and P. Fua. BRIEF: Binary Robust Independent Elementary Features. ECCV 2010.
- 画像から直接バイナリ特徴を抽出する手法
- 高速でありながらSURFと同等の表現能力を持つ.

効率的な局所記述子の必要性

- 大量の局所特徴
 - 一枚の画像から数万点程度の局所特徴が得られ、対象となる画像は永遠に増え続ける可能性がある。
 - 局所記述子はできる限りコンパクトである必要がある。
- 画像同士のマッチング
 - 入力画像が一枚でも、対象となる画像は大量であることが一般的。
 - 局所記述子間の類似度が高速であることが望ましい。
- モバイルデバイスでの利用
 - 計算リソースが限られる。
 - 高速かつ省メモリが望ましい。

コンパクトな記述子の作り方



例) SURF

4 bytes x 64 dim = 256 bytes

$$\mathbf{d}_j = \underbrace{(0.33 \ 0.52 \ 0.92 \cdots 0.27)}_{64 \text{ dim}}$$

次元削減

4 bytes x 16 dim = 64 bytes

$$\mathbf{d}'_j = \underbrace{(0.12 \cdots 0.37)}_{16 \text{ dim}}$$

量子化

4 bits x 64 dim = 256 bits

$$\mathbf{d}'_j = \underbrace{(1 \ 0 \ 3 \cdots 2)}_{64 \text{ dim}}$$

バイナリ化

1 bit x 64 dim = 64 bits

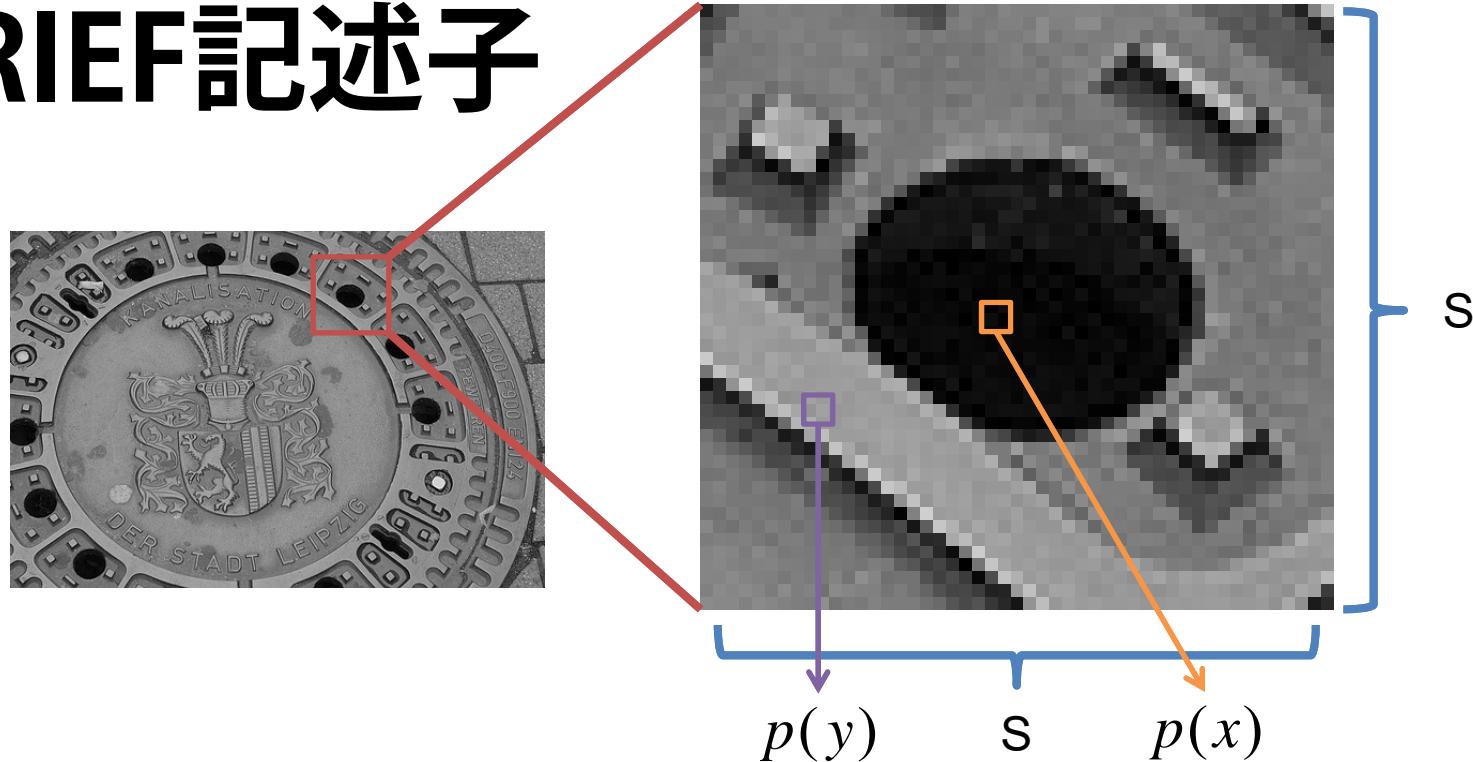
$$\mathbf{d}''_j = \underbrace{(0 \ 1 \ 1 \cdots 0)}_{64 \text{ dim}}$$

1011101
1001001

Hamming距離=2
XORで効率的に計算可能

- ・ 次元削減, 量子化, バイナリ化
- ・ 画像から「直接」バイナリ特徴を抽出できれば効率的！

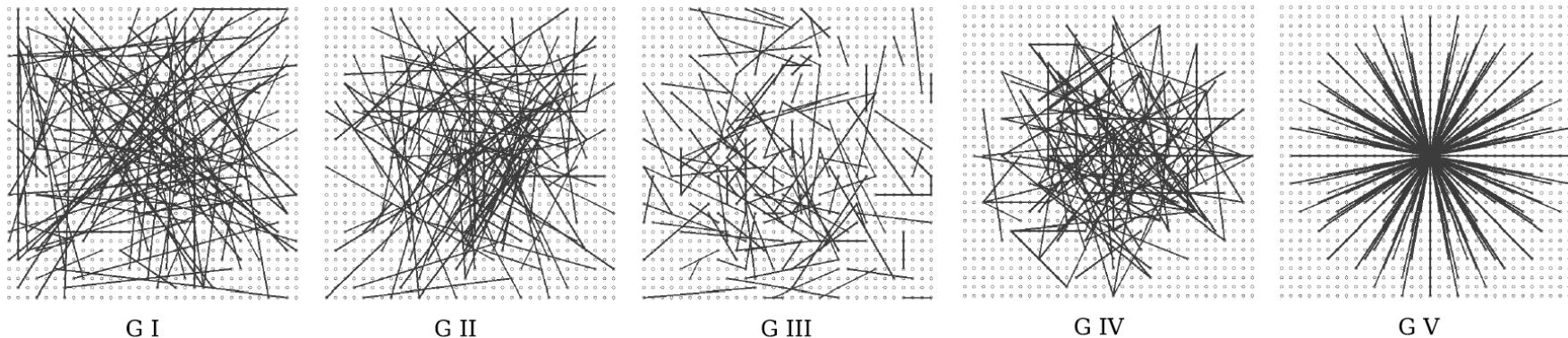
BRIEF記述子



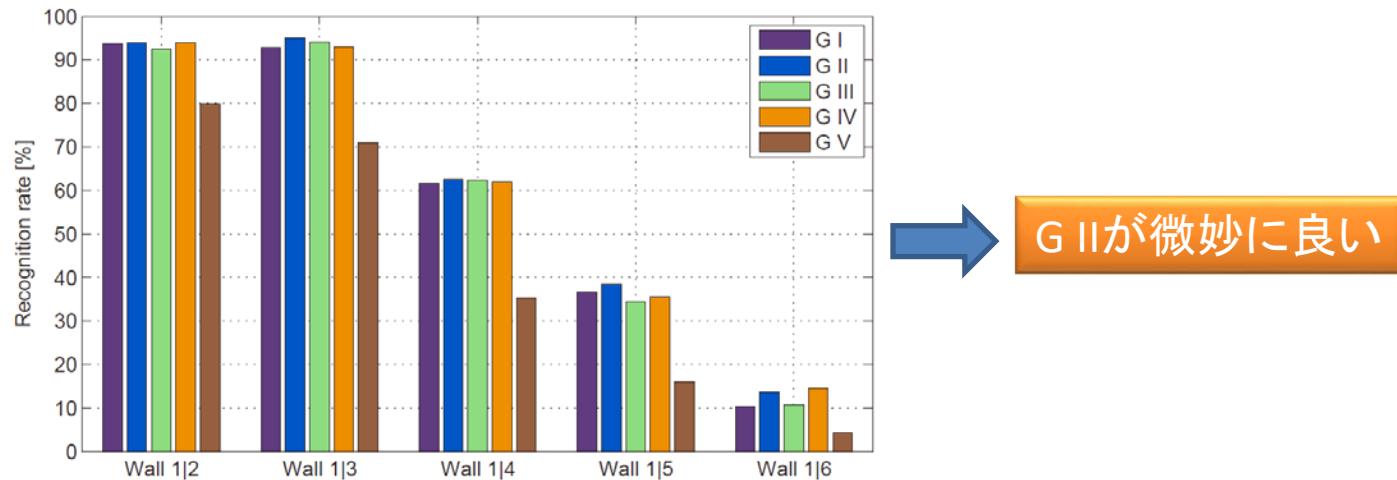
$$\tau(\mathbf{p}; \mathbf{x}, \mathbf{y}) := \begin{cases} 1 & \text{if } \mathbf{p}(\mathbf{x}) < \mathbf{p}(\mathbf{y}) \\ 0 & \text{otherwise} \end{cases} \quad f_{n_d}(\mathbf{p}) := \sum_{1 \leq i \leq n_d} 2^{i-1} \tau(\mathbf{p}; \mathbf{x}_i, \mathbf{y}_i)$$

- ・ パッチの中の適切な2点を選択
- ・ 2点の輝度の高低によって1, 0で記述
- ・ パッチ内の2点の選択方法が問題

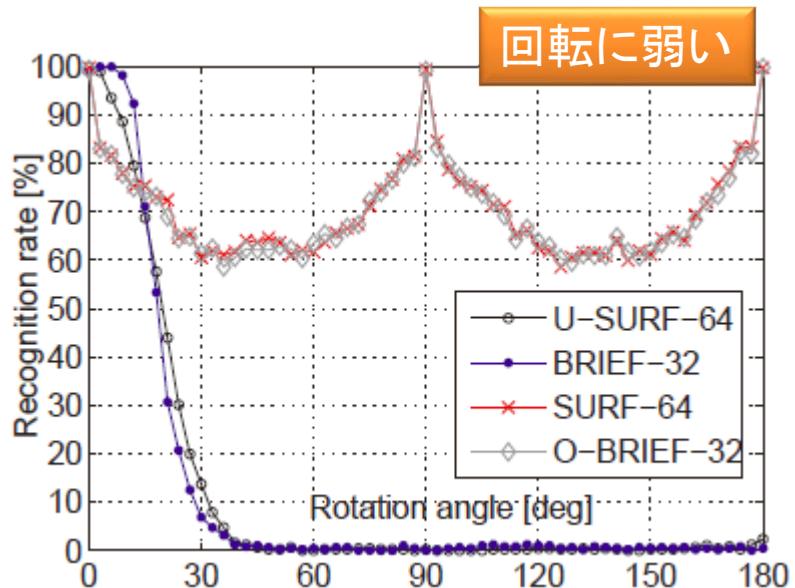
2点の選び方



- G I: $(X, Y) \sim \text{i.i.d. Uniform}(-\frac{S}{2}, \frac{S}{2})$
- G II: $(X, Y) \sim \text{i.i.d. Gaussian}(0, \frac{1}{25}S^2)$
- G III: $X \sim \text{i.i.d. Gaussian}(0, \frac{1}{25}S^2)$, $Y \sim \text{i.i.d. Gaussian}(x_i, \frac{1}{100}S^2)$
- G IV: randomly sampled from discrete locations of a coarse polar grid
- G V: $x=(0,0)$, y takes all possible values on a coarse polar grid



BRIEF記述子の性能



計算速度速い [msec]

	BRIEF-16	BRIEF-32	BRIEF-64	SURF-64
Descriptor computation	8.18	8.87	9.57	335
Matching (exact NN)	2.19	4.35	8.16	28.3

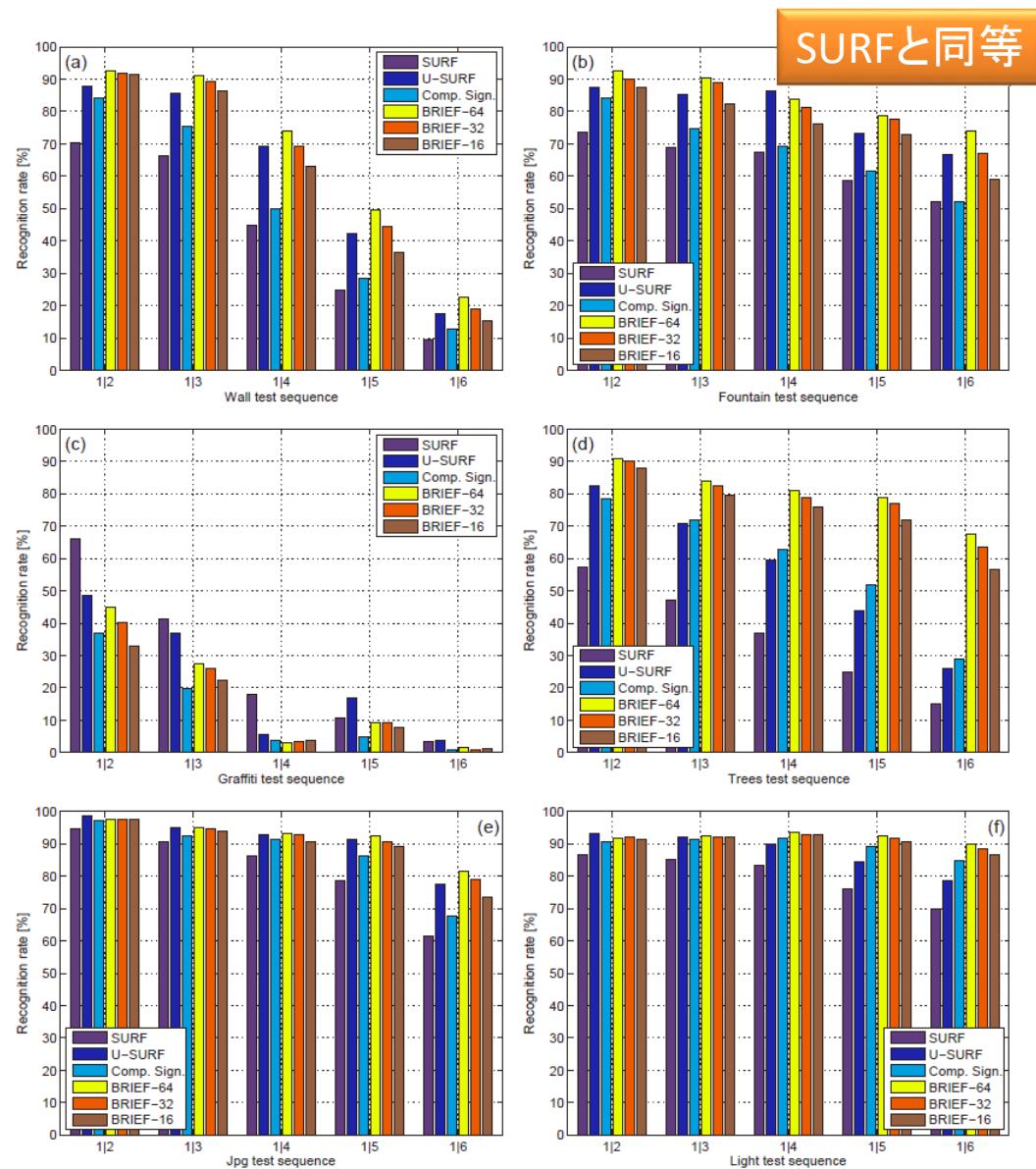


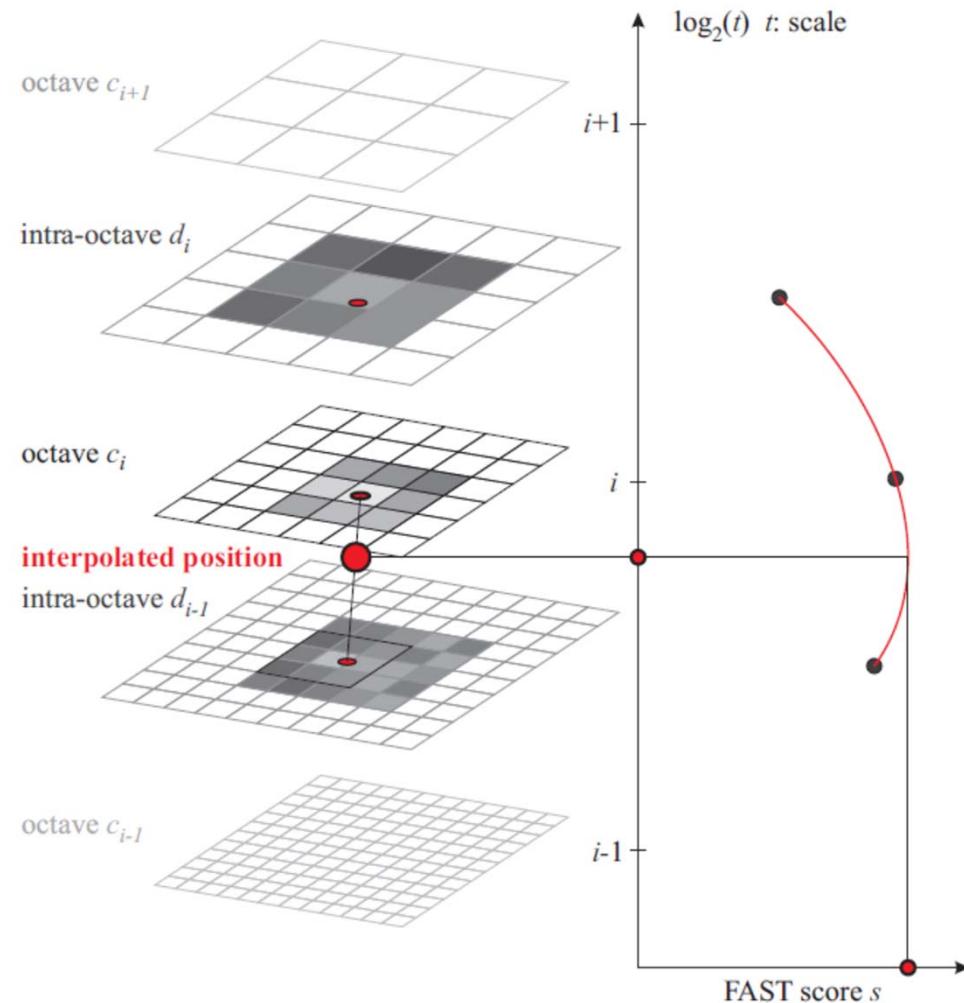
Fig. 6. Recognition rates on (a) Wall (b) Fountain. (c) Graffiti (d) Trees (e) Jpg (f) Light. The trailing 16, 32, or 64 in the descriptor's name is its length in bytes. It is much shorter than those of SURF and U-SURF, which both are 256. For completeness, we also compare to a recent approach called Compact Signatures [7] which has been shown to be very efficient. We obtained the code from OpenCV's SVN repository.

BRISK

- Sefan Leutenegger, Margarita Chli and Roland Y. Siegwart. Binary Robust Invariant Scalable Keypoints. ICCV, 2011.
- Detector
 - Scale-Space Keypoint Detection
 - 特徴点は画像内だけではなく、スケール軸に関しても検出される. →スケール不変
 - 特徴点は、サブピクセル、サブスケールで検出される.
- Descriptor
 - 特徴点まわりのピクセルの局所的勾配を計算し、特徴的な方向を検出. →回転不変.
 - BRISKのサンプリングパターンを適用し、ピクセル間の輝度の差分を計算して、バイナリ特徴を生成する.

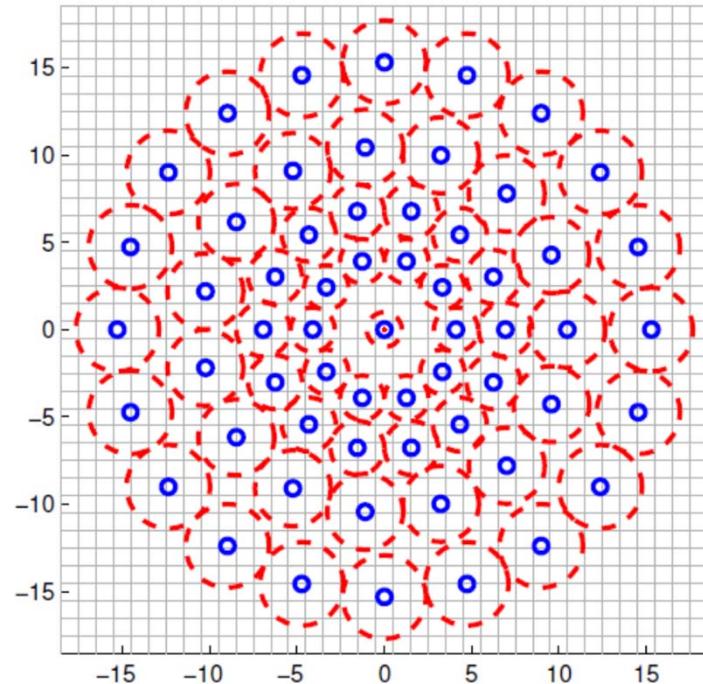
Scale-Space Keypoint Detection

- 検出手順
 - FAST detectorを各 octaveに適用し、特徴点の候補を検出
 - Non-maximal suppression
 - 各octaveで 3×3 のピクセル内に2Dの2次関数を適合してサブピクセルの特徴点を検出
 - スケール間でのFAST scoreを2次関数で近似して、最大値となるスケールを求める。そのスケールにおけるサブピクセルも求める。



Sampling Pattern and Rotation Estimation

- サンプリングパターン
 - パターンは推定されたスケールによって変える
 - 青い点がサンプリング場所, 赤い円がガウシアンを用いたスムージングの分散



- 回転方向の推定

- 局所勾配

$$\mathbf{g}(\mathbf{p}_i, \mathbf{p}_j) = (\mathbf{p}_j - \mathbf{p}_i) \cdot \frac{I(\mathbf{p}_j, \sigma_j) - I(\mathbf{p}_i, \sigma_i)}{\|\mathbf{p}_j - \mathbf{p}_i\|^2}.$$

- 回転方向の推定

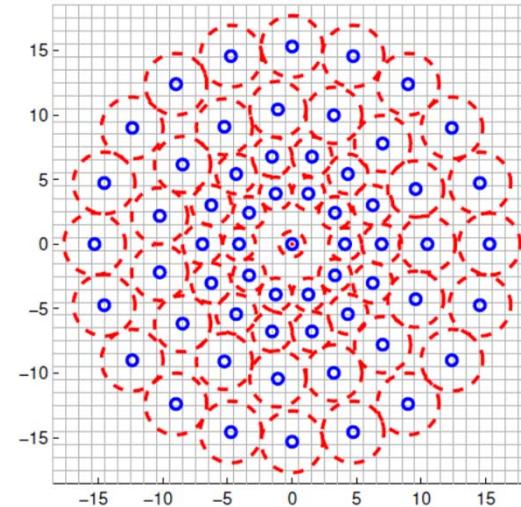
$$\mathbf{g} = \begin{pmatrix} g_x \\ g_y \end{pmatrix} = \frac{1}{L} \cdot \sum_{(\mathbf{p}_i, \mathbf{p}_j) \in \mathcal{L}} \mathbf{g}(\mathbf{p}_i, \mathbf{p}_j).$$

一定距離離れたサンプリング点の間の局所勾配の平均から回転方向を推定

Building the Descriptor

$$b = \begin{cases} 1, & I(\mathbf{p}_j^\alpha, \sigma_j) > I(\mathbf{p}_i^\alpha, \sigma_i) \\ 0, & \text{otherwise} \end{cases}$$

$$\forall (\mathbf{p}_i^\alpha, \mathbf{p}_j^\alpha) \in \mathcal{S}$$



- 推定された支配的な輝度勾配を用いて正規化（回転方向の補正）
- 距離の短いピクセル同士の輝度を比較し，その大小でバイナリ値を与える.
- N=60で512次元の特徴ベクトル

BRISK: Demo



Autonomous Systems Lab



Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

BRISK

Binary Robust Invariant Scalable Keypoints



Stefan Leutenegger, Margarita Chli and Roland Siegwart

ICCV 2011

ORB: Oriented FAST and Rotated BRIEF

- E. Rublee, V. Rabaud, K. Konolige and G. Bradski. ORB: an efficient alternative to SIFT or SURF. ICCV, 2011.
- Detector: FAST
 - ただし回転不变を追加
- Descriptor: BRIEF
 - ただし回転にも対応できるようになっている.
 - Binary testの2点の選択方法を学習により獲得

Keypoint Orientation

- Intensity centroid

- コーナーの輝度の中心は中心から少し離れており, このオフセットを利用してすることで, Orientationが得られる.

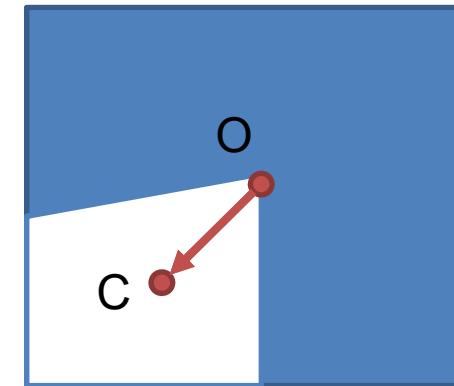
- パッチのモーメント

$$m_{pq} = \sum_{x,y} x^p y^q I(x, y),$$

- 輝度の中心

$$C = \left(\frac{m_{10}}{m_{00}}, \frac{m_{01}}{m_{00}} \right)$$

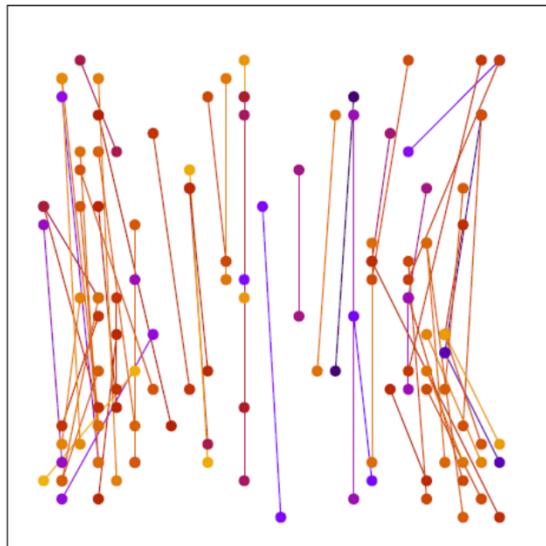
- パッチの中心から輝度の中心の向きを図ることでパッチのOrientationが計算できる.



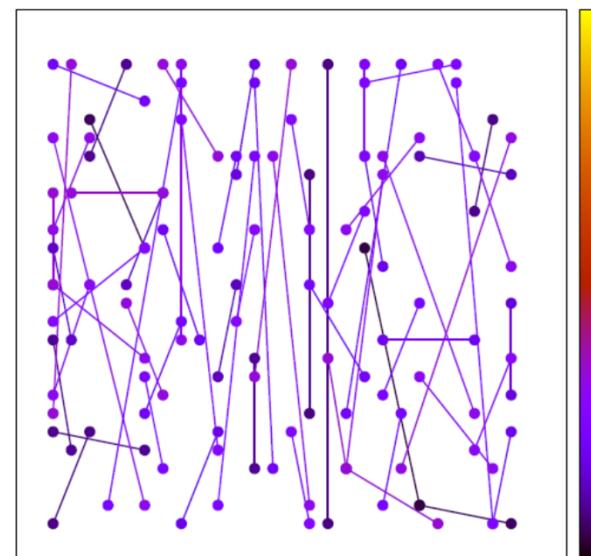
$$\theta = \text{atan2}(m_{01}, m_{10}),$$

Learning Good Binary Features

- 良い特徴とは?
 - テスト（ピクセル間の比較）を比較し、分散の大きなテストを選ぶ
 - ・いつも0ばかり、1ばかり出力するようなテストは判別的でない。
 - 互いに相関の低いテスト群を選ぶ
 - ・例えば、距離が近いテスト同士は、出力結果が似通る可能性があり、判別的ではない。つまり出力結果が似通る=相関が高い、テストは採用しない。
- これらのテスト（どのピクセル間の比較を行うか？）を大量のパッチを用いて学習



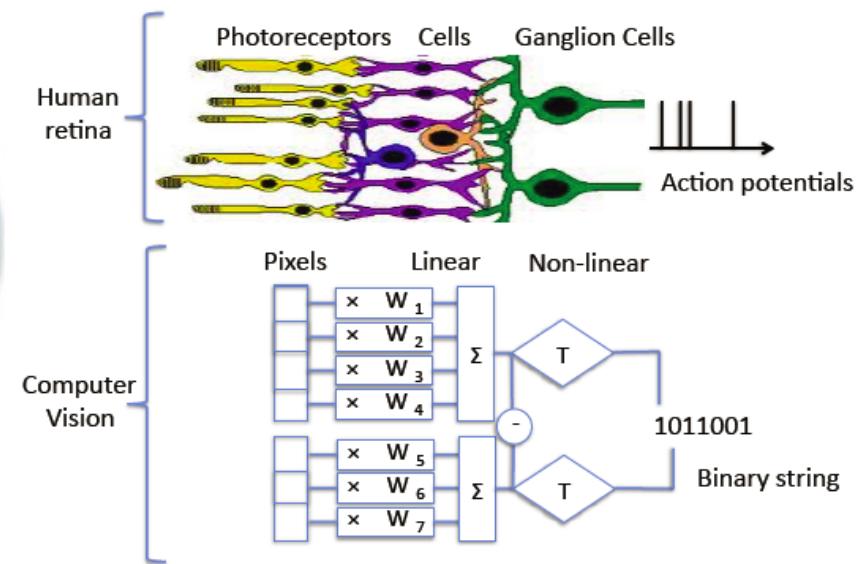
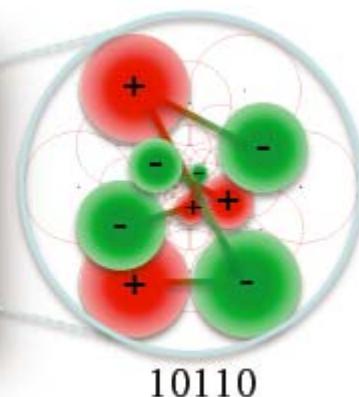
分散が大きなテスト群



分散が大きく相関の低いペア群

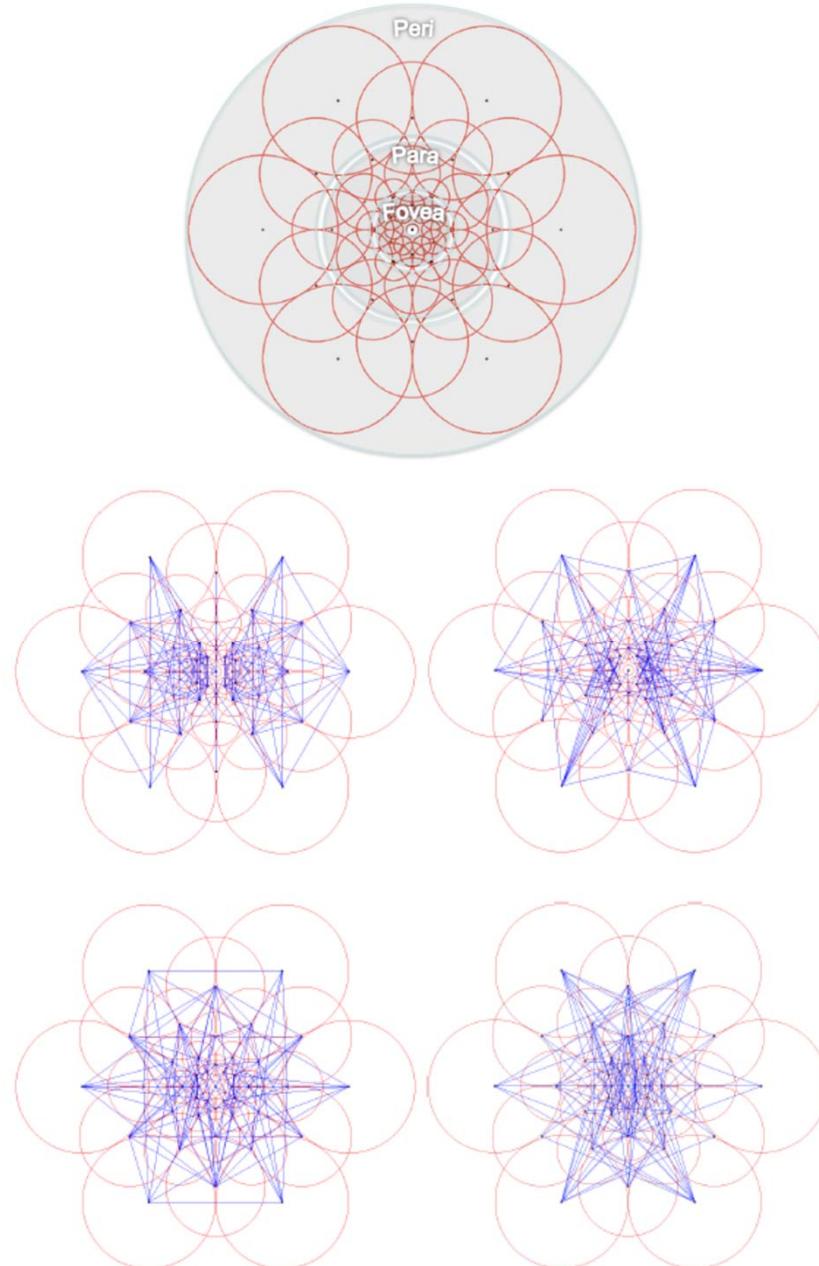
FREAK

- Alexandre Alahi, Raphaël Ortiz, Pierre Vandergheynst. FREAK: Fast Retina Keypoint. CVPR, 2012.
- CVPR 2012 Open Source Code Award



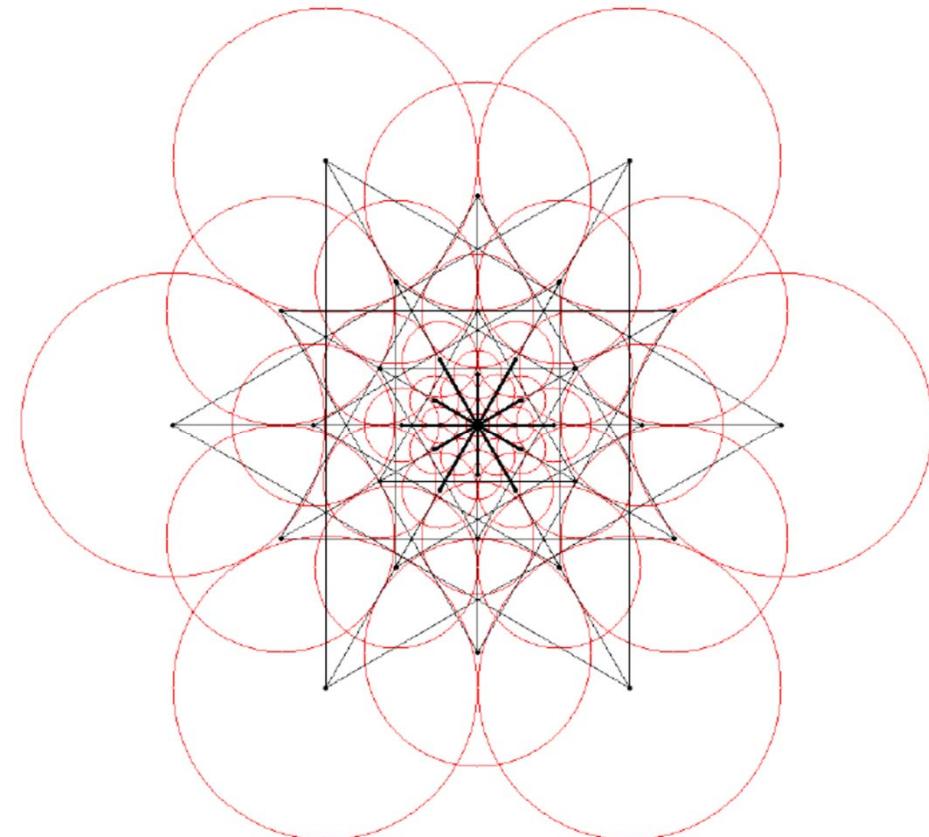
FREAK sampling pattern

- ・網膜神経節細胞の分布と類似したサンプリングパターン
- ・ピクセル間の輝度の大小でバイナリ特徴を生成
- ・テストの選択はORBのように学習によって選ぶ



Orientation

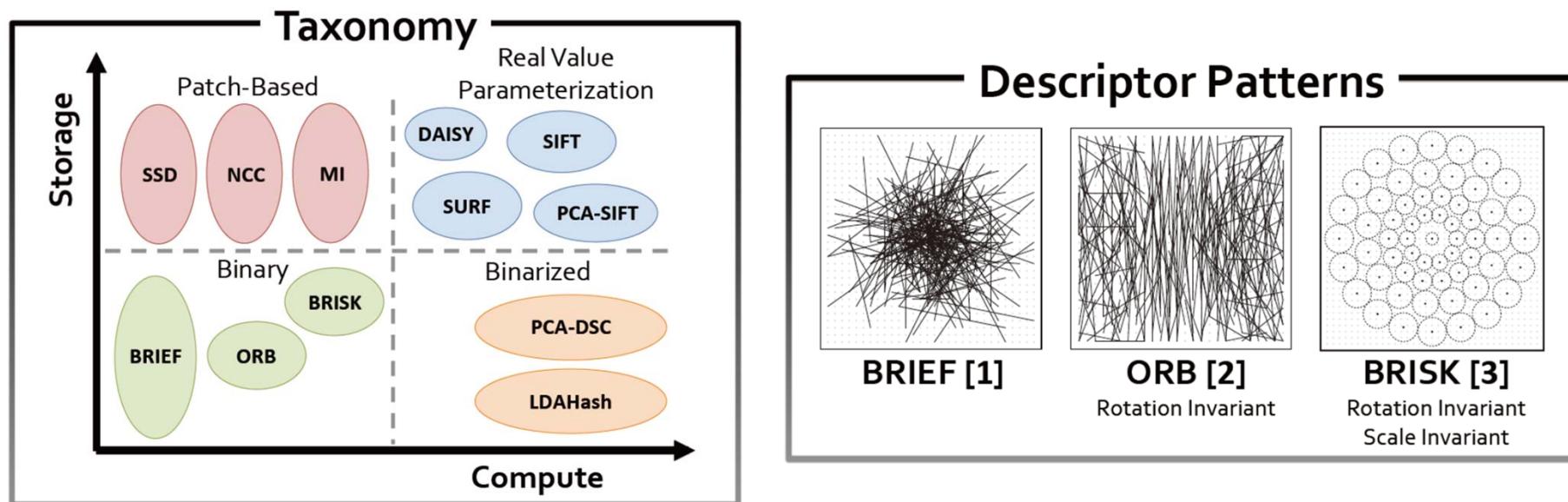
- BRISKと同じように、適切なペアを選択して、局所的な輝度勾配の平均をそのパッチの Orientationと推定する。



$$O = \frac{1}{M} \sum_{P_o \in G} (I(P_o^{r_1}) - I(P_o^{r_2})) \frac{P_o^{r_1} - P_o^{r_2}}{\|P_o^{r_1} - P_o^{r_2}\|},$$

Comparative Evaluation of Binary Features

- Jared Heinly, Enrique Dunn, and Jan-Michael Frahm.
Comparative Evaluation of Binary Features. ECCV, 2012.



Computation Requirements

Detector/Descriptor	BRIEF	ORB	BRISK	SURF	SIFT	FAST	Harris	MSER
Detector ms/image	n/a	17	43	377	572	2.7	78	117
Descriptor μ s/feature	4.4	4.8	12.9	143	314	n/a	n/a	n/a
Storage bytes/feature	16,32,64	32	64	64(256)	128(512)	n/a	n/a	n/a

Comparative Evaluation of Binary Features

- Jared Heinly, Enrique Dunn, and Jan-Michael Frahm.
Comparative Evaluation of Binary Features. ECCV, 2012.

Detectors

H = Harris

MR = MSER

FT = FAST

OB = ORB

BK = BRISK

SF = SURF

ST = SIFT

Blur

Putative Match Ratio

Compression

Putative Match Ratio

Exposure

Putative Match Ratio

Day-Night Illumination

Putative Match Ratio

Precision

Precision

Precision

Precision

Matching Score

Matching Score

Matching Score

Matching Score

Recall

Recall

Recall

Recall

Matching Score

Matching Score

Matching Score

Matching Score

Recall

Recall

Descriptors

BF = BRIEF

OB = ORB

BK = BRISK

SF = SURF

ST = SIFT

幾何学的変換、透視変換がない状況ならBRIEFがよい

Comparative Evaluation of Binary Features

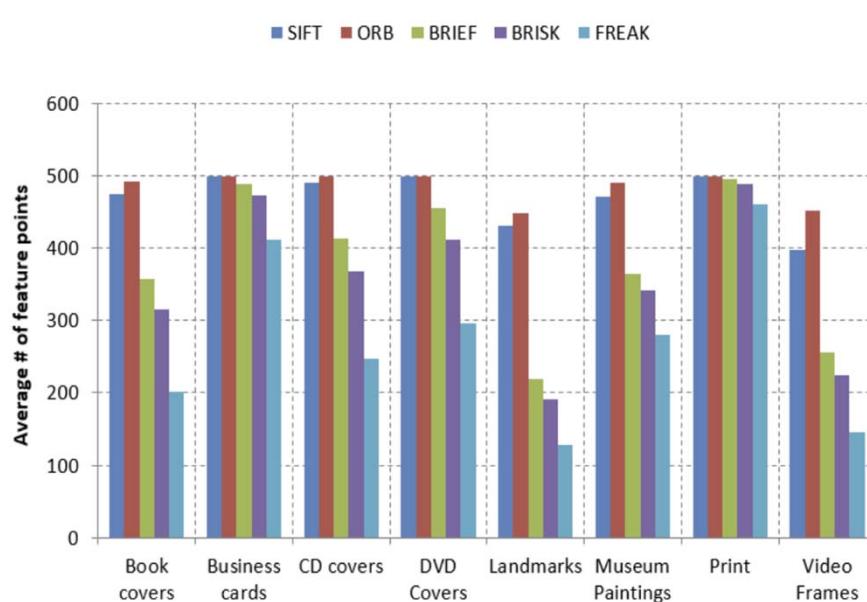
- Jared Heinly, Enrique Dunn, and Jan-Michael Frahm.
Comparative Evaluation of Binary Features. ECCV, 2012.

Detectors	Scale					Rotation					Scale & Rotation					Perspective Planar Scene					Perspective Non-Planar					Photo Collection									
	Putative Match Ratio					Putative Match Ratio					Putative Match Ratio					Putative Match Ratio					Putative Match Ratio					Putative Match Ratio									
	BF	OB	BK	SF	ST		BF	OB	BK	SF	ST		BF	OB	BK	SF	ST		BF	OB	BK	SF	ST		BF	OB	BK	SF	ST		BF	OB	BK	SF	ST
H = Harris	19	10	12			H	3	69	64			H	4	6	6			H	29	14	17			H	11	6	6			H	11	6	6		
MR = MSER	10	8	9			MR	5	44	42			MR	6	6	5			MR	15	9	9			MR	11	7	7			MR	11	7	7		
FT = FAST	15	7	12			FT	3	62	68			FT	2	4	6			FT	25	13	17			FT	9	5	5			FT	9	5	5		
OB = ORB	5	21	5			OB	1	62	58			OB	1	8	4			OB	12	15	13			OB	4	6	4			OB	4	6	4		
BK = BRISK	11	7	40			BK	2	52	51			BK	2	4	13			BK	19	11	13			BK	7	4	7			BK	7	4	7		
SF = SURF	15	8	8	56		SF	4	44	38	43		SF	3	4	4	14		SF	18	10	15			SF	10	6	5	11		SF	10	6	5	11	
ST = SIFT	14	7	10		57	ST	3	41	50		58	ST	3	4	4		13	ST	18	9	13		24	ST	8	4	5		11	ST	8	4	5		11
Precision																																			
H	80	80	87			H	14	93	88			H	34	65	81			H	88	87	90			H	60	66	73			H	81	78	84		
MR	69	48	53			MR	18	90	78			MR	19	40	35			MR	76	68	70			MR	55	39	37			MR	76	72	63		
FT	77	73	89			FT	15	91	93			FT	23	64	83			FT	84	85	89			FT	73	73	81			FT	84	77	86		
OB	79	92	91			OB	12	86	94			OB	20	83	86			OB	89	88	92			OB	77	80	85			OB	87	84	88		
BK	74	73	78			BK	12	89	86			BK	27	68	84			BK	86	83	87			BK	63	62	72			BK	83	77	86		
SF	75	67	70	81		SF	13	88	79	82		SF	20	48	58	68		SF	78	73	75	73		SF	58	54	58	61		SF	79	73	75	79	
ST	76	72	82		96	ST	13	93	84		95	ST	18	55	72		84	ST	83	80	81		86	ST	69	68	75		84	ST	76	71	75		83
Matching Score																																			
H	15	8	11			H	0	64	56			H	1	4	5			H	26	13	15			H	13	9	10			H	9	5	5		
MR	7	4	5			MR	1	40	33			MR	1	2	2			MR	11	6	6			MR	8	5	4			MR	8	5	4		
FT	11	6	10			FT	0	57	64			FT	1	3	5			FT	21	11	15			FT	7	4	5			FT	7	4	5		
OB	4	19	5			OB	0	53	55			OB	0	7	3			OB	11	13	12			OB	7	10	8			OB	4	5	3		
BK	8	5	31			BK	0	46	44			BK	0	3	11			BK	16	9	11			BK	6	3	6			BK	6	3	6		
SF	11	5	6	45		SF	0	38	30	35		SF	1	2	2	9		SF	14	7	7	11		SF	10	5	5	11		SF	8	4	4	8	
ST	11	5	8		54	ST	0	38	42		55	ST	0	2	3		11	ST	15	7	11		20	ST	11	6	10		19	ST	6	3	4		10
Recall																																			
H	24	13	17			H	0	68	62			H	2	6	9			H	36	18	22			H	30	19	23			H	8	5	5		
MR	10	6	7			MR	1	50	39			MR	5	9	8			MR	25	13	14			MR	32	15	13			MR	8	5	4		
FT	19	9	17			FT	0	63	74			FT	1	4	7			FT	28	8	21			FT	30	17	24			FT	7	4	5		
OB	6	29	6			OB	0	62	53			OB	0	8	3			OB	13	15	12			OB	12	18	11			OB	4	5	3		
BK	11	7	42			BK	0	52	48			BK	1	3	14			BK	22	7	15			BK	19	10	20			BK	6	3	5		
SF	17	8	8	69		SF	1	57	45	55		SF	1	5	6	25		SF	32	17	17	25		SF	28	14	15	33		SF	8	4	4	9	
ST	16	7	10		80	ST	1	50	48		79	ST	1	5	7		30	ST	28	14	17		42	ST	24	13	18		44	ST	6	3	3		10

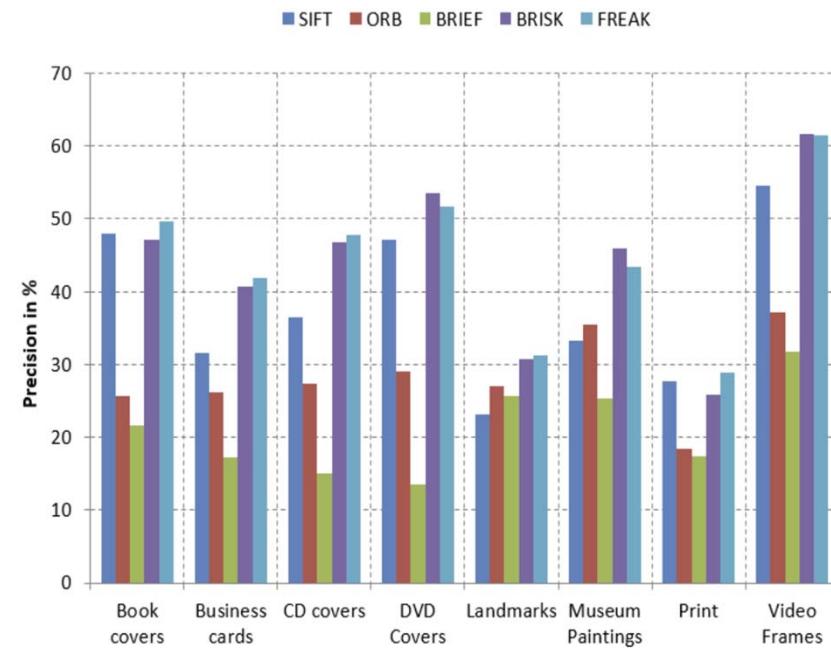
幾何学的変換のない状況ならSIFTがベスト

Evaluation of binary keypoint descriptors

- Dagmawi Bekele, Michael Teutsch, Tobias Schuchert. Evaluation of binary keypoint descriptors. ICIP, 2013.
- The stanford mobile visual search data set
 - <http://www.stanford.edu/~dmchen/datasets.html>



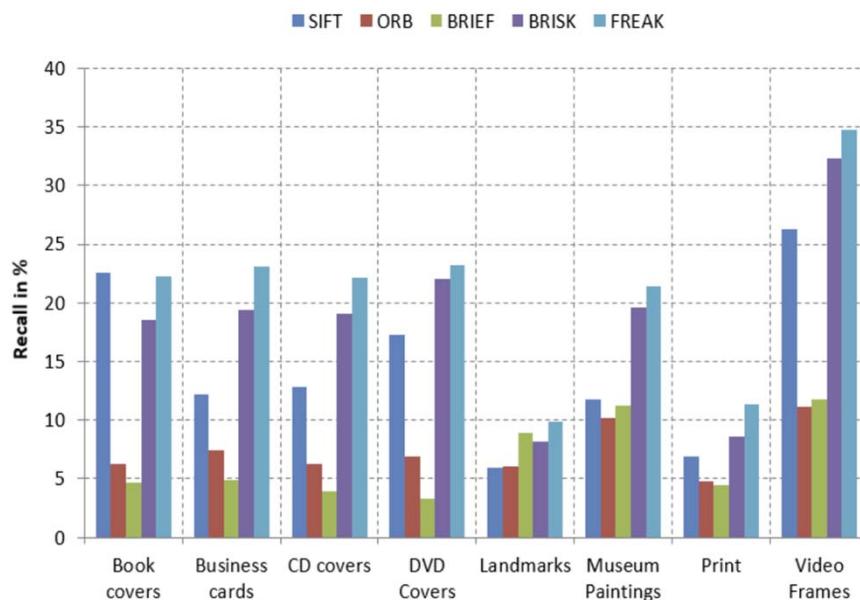
Average number of feature points for binary descriptors



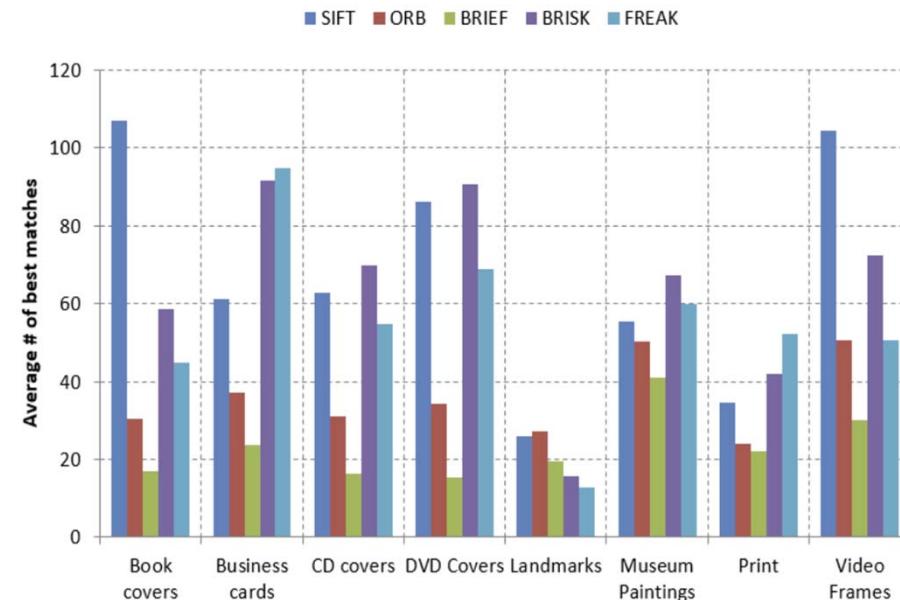
Precision in percentage

Evaluation of binary keypoint descriptors

- Dagmawi Bekele, Michael Teutsch, Tobias Schuchert. Evaluation of binary keypoint descriptors. ICIP, 2013.
- The stanford mobile visual search data set
 - <http://www.stanford.edu/~dmchen/datasets.html>



Recall in percentage

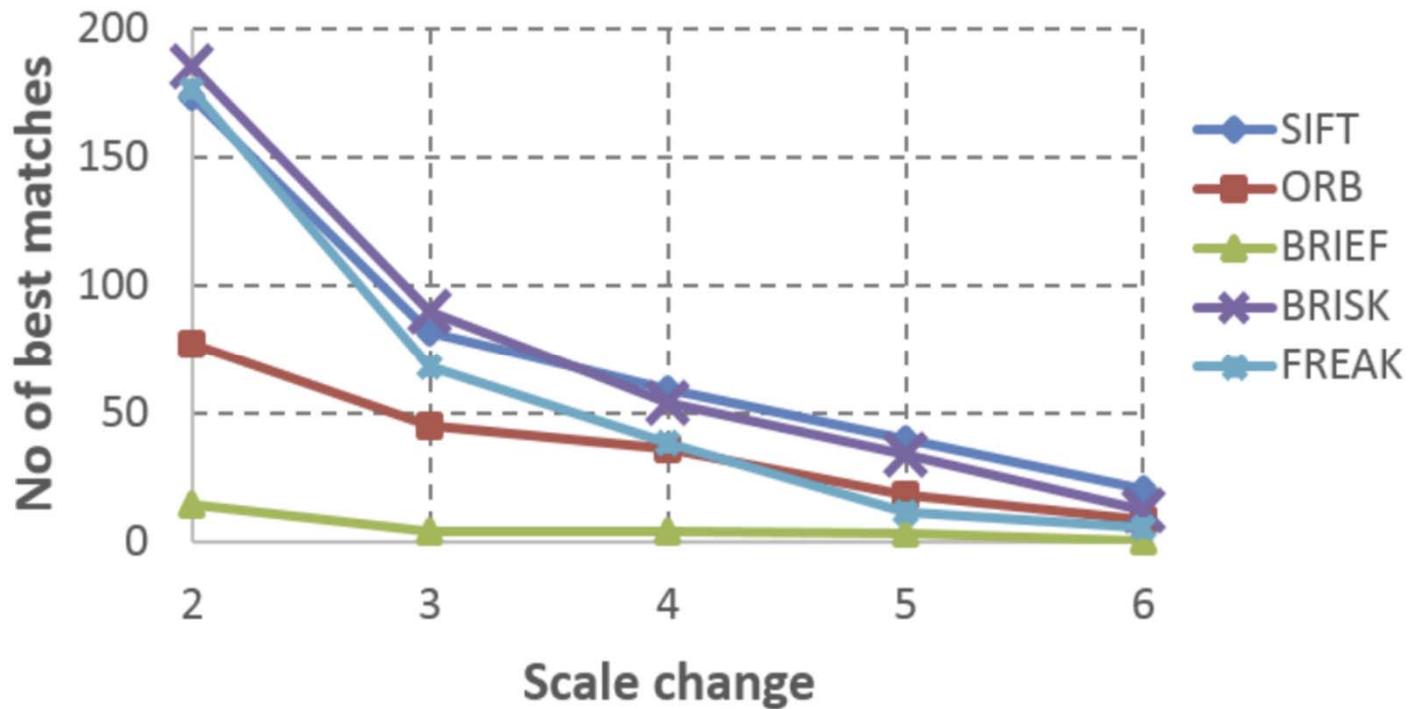


Average number of best matches

BRISKがSIFTと同程度に良さそう

Evaluation of binary keypoint descriptors

- Dagmawi Bekele, Michael Teutsch, Tobias Schuchert. Evaluation of binary keypoint descriptors. ICIP, 2013.
- Bark sub data from OxFord dataset
 - <http://www.robots.ox.ac.uk/~vgg/research/affine/index.html>



SIFT = BRISK > FREAK > ORB > BRIEF

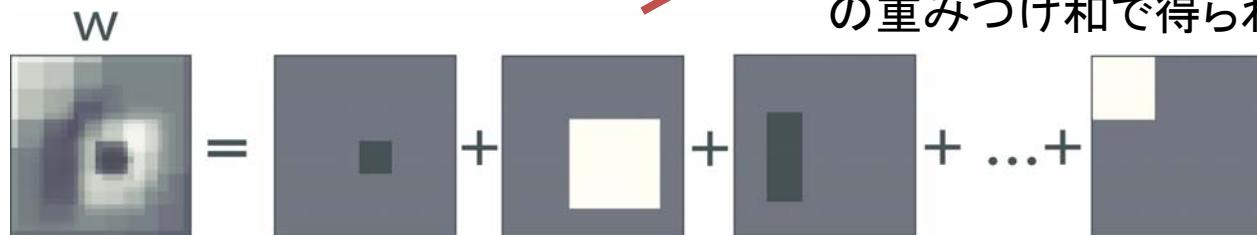
Discriminative BRIEF

- Tomasz Trzcinski and Vincent Lepetit. Efficient Discriminative Projections for Compact Binary Descriptors. ECCV 2012.

バイナリ値を出力する重みと閾値を学習で得る

$$\forall i \in 1, \dots, N \quad b_i = \text{sign}(\mathbf{w}_i^\top \mathbf{x} + \tau_i)$$

重みは簡単なフィルタ群(辞書)
の重みづけ和で得られるとする



下記の評価関数の最小化問題を解く

$$\begin{aligned} \min_{\{(\mathbf{s}_i, \tau_i)\}} \quad & \sum_{i \in 1, \dots, N} \sum_{(\mathbf{x}, \mathbf{x}') \in \mathcal{N}} \text{sign}((D\mathbf{s}_i)^\top \mathbf{x} + \tau_i) \text{sign}((D\mathbf{s}_i)^\top \mathbf{x}' + \tau_i) - \\ & \sum_{(\mathbf{x}, \mathbf{x}') \in \mathcal{P}} \text{sign}((D\mathbf{s}_i)^\top \mathbf{x} + \tau_i) \text{sign}((D\mathbf{s}_i)^\top \mathbf{x}' + \tau_i) + \lambda |\mathbf{s}_i|_1 \end{aligned}$$

subject to $(D\mathbf{s}_i)^\top (D\mathbf{s}_j) = \delta_{ij}$,

ネガティブパッチ間のバイナリ値
はなるべく一致しないように

なるべく少数の辞
書の組み合わせで

ポジティブパッチ間のバイナリ値
はなるべく一致するように

Discriminative BRIEF

- Tomasz Trzcinski and Vincent Lepetit. Efficient Discriminative Projections for Compact Binary Descriptors. ECCV 2012.

Sign関数はいやらしいので、連続値として計算する

$$\min_{\{\mathbf{s}_i\}} \sum_i \frac{\sum_{(\mathbf{x}, \mathbf{x}') \in \mathcal{P}} ((D\mathbf{s}_i)^\top (\mathbf{x} - \mathbf{x}'))^2}{\sum_{(\mathbf{x}, \mathbf{x}') \in \mathcal{N}} ((D\mathbf{s}_i)^\top (\mathbf{x} - \mathbf{x}'))^2} + \lambda |\mathbf{s}_i|_1$$

subject to $(D\mathbf{s}_i)^\top (D\mathbf{s}_j) = \delta_{ij}$

ポジティブパッチ間の値はなるべく一致するように

なるべく少数の辞書の組み合わせで

ネガティブパッチ間の値はなるべく一致しないように

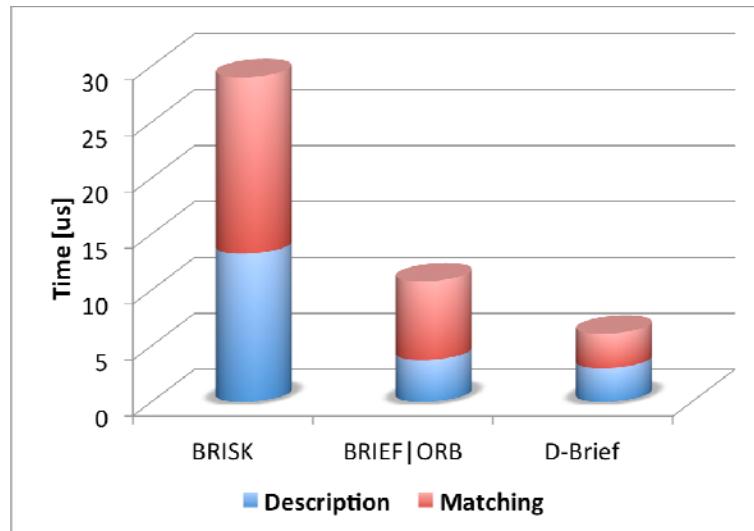
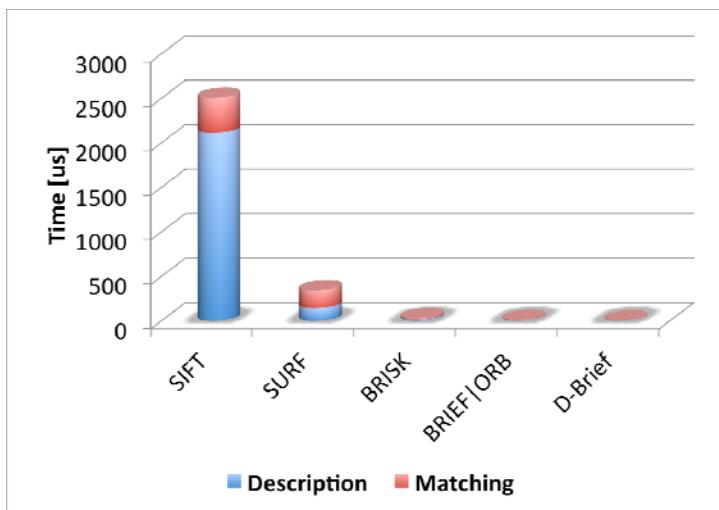
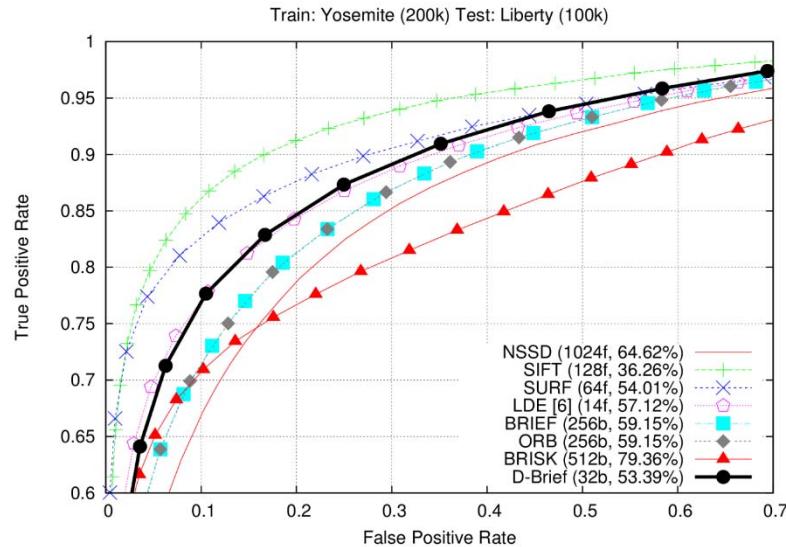
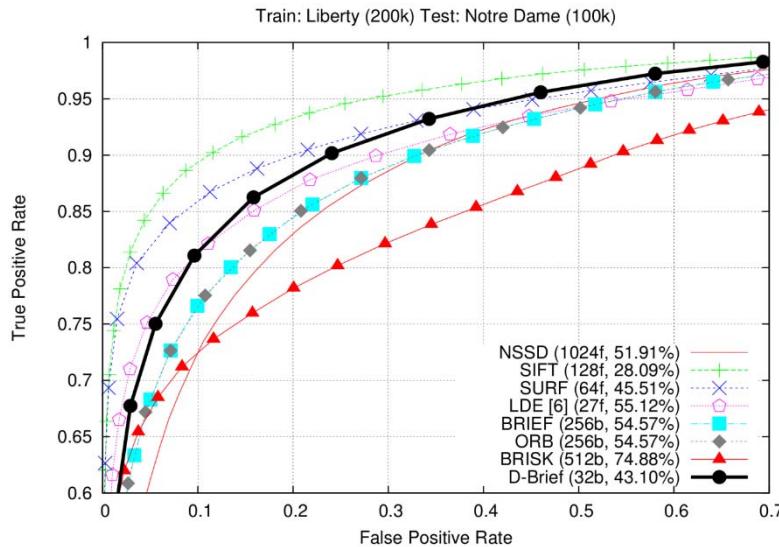
初期値の計算

$$\{\mathbf{w}_i^0\} = \arg \min_{\{\mathbf{w}_i\}} \sum_i \frac{\sum_{(\mathbf{x}, \mathbf{x}') \in \mathcal{P}} (\mathbf{w}_i^\top (\mathbf{x} - \mathbf{x}'))^2}{\sum_{(\mathbf{x}, \mathbf{x}') \in \mathcal{N}} (\mathbf{w}_i^\top (\mathbf{x} - \mathbf{x}'))^2}$$

$$\{\mathbf{s}_i^0\} = \arg \min_{\mathbf{s}_i} \|\mathbf{w}_i^0 - D\mathbf{s}_i\|_2^2 + \lambda |\mathbf{s}_i|_1$$

Discriminative BRIEF

- Tomasz Trzcinski and Vincent Lepetit. Efficient Discriminative Projections for Compact Binary Descriptors. ECCV 2012.



BinBoost Descriptor

- Tomasz Trzcinski, Mario Christoudias, Pascal Fua and Vincent Lepetit. Boosting Binary Keypoint Descriptors. CVPR, 2013.
- ハッシュ関数をBoostingを利用して学習

$$C_d(\mathbf{x}) = \text{sgn} (\mathbf{b}_d^T \mathbf{h}_d(\mathbf{x}))$$

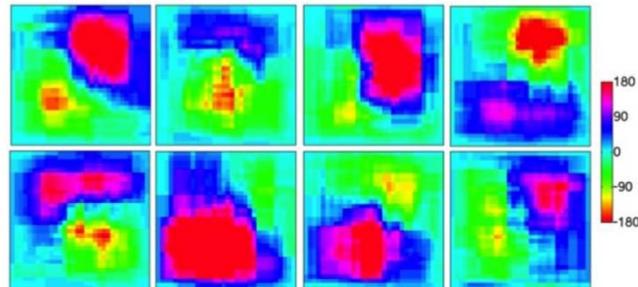
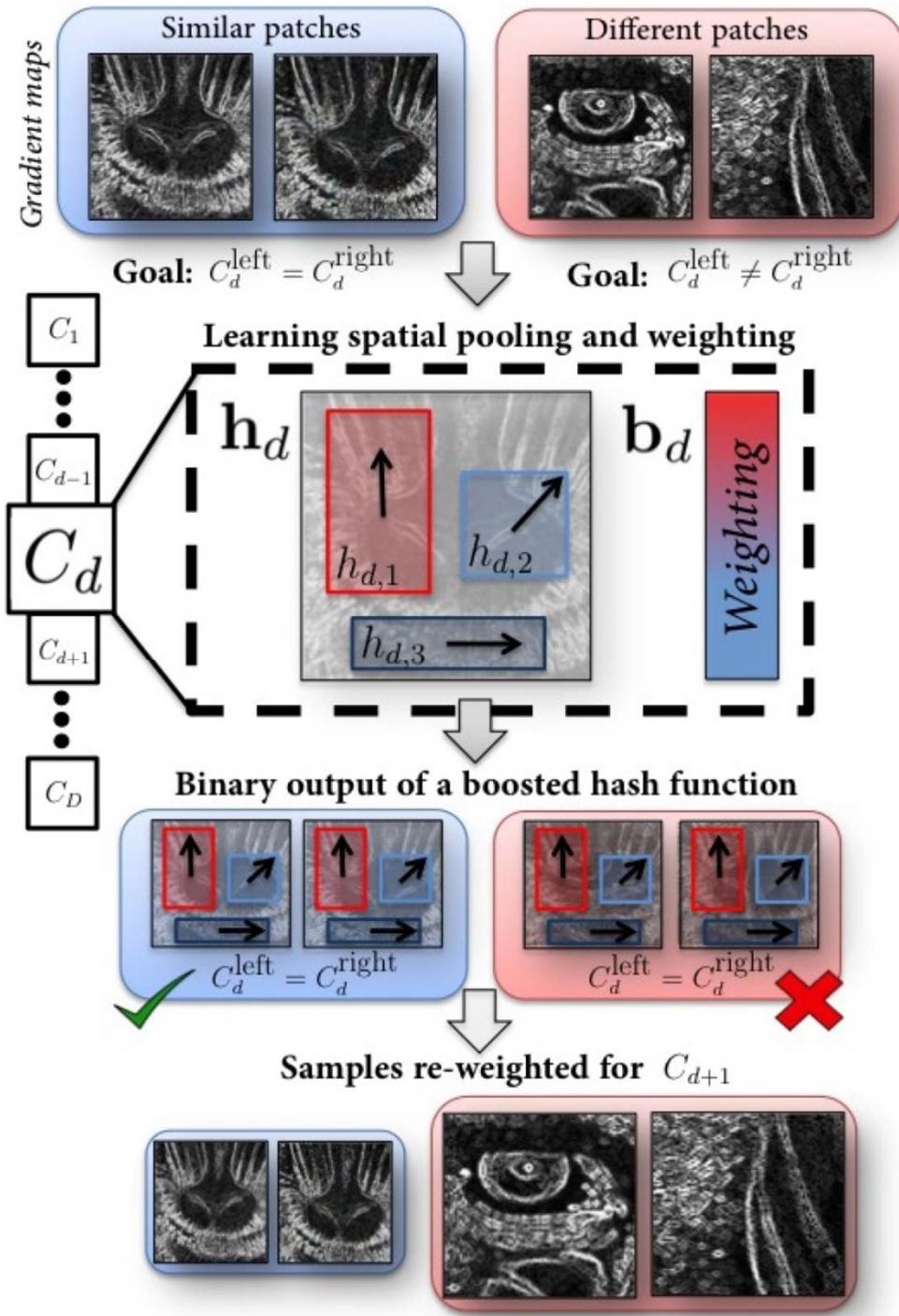


Figure 4. Visualization of the selected weak learners for the first 8 bits learned on 200k pairs of 32×32 patches from the Notre Dame dataset (best viewed on screen). For each pixel of the figure we show the average orientation weighted by the weights of the weak learners \mathbf{b}_d . For different bits, the weak learners cluster about different regions and orientations illustrating their complementary nature.



BinBoost Descriptor

- Tomasz Trzcinski, Mario Christoudias, Pascal Fua and Vincent Lepetit. Boosting Binary Keypoint Descriptors. CVPR, 2013.

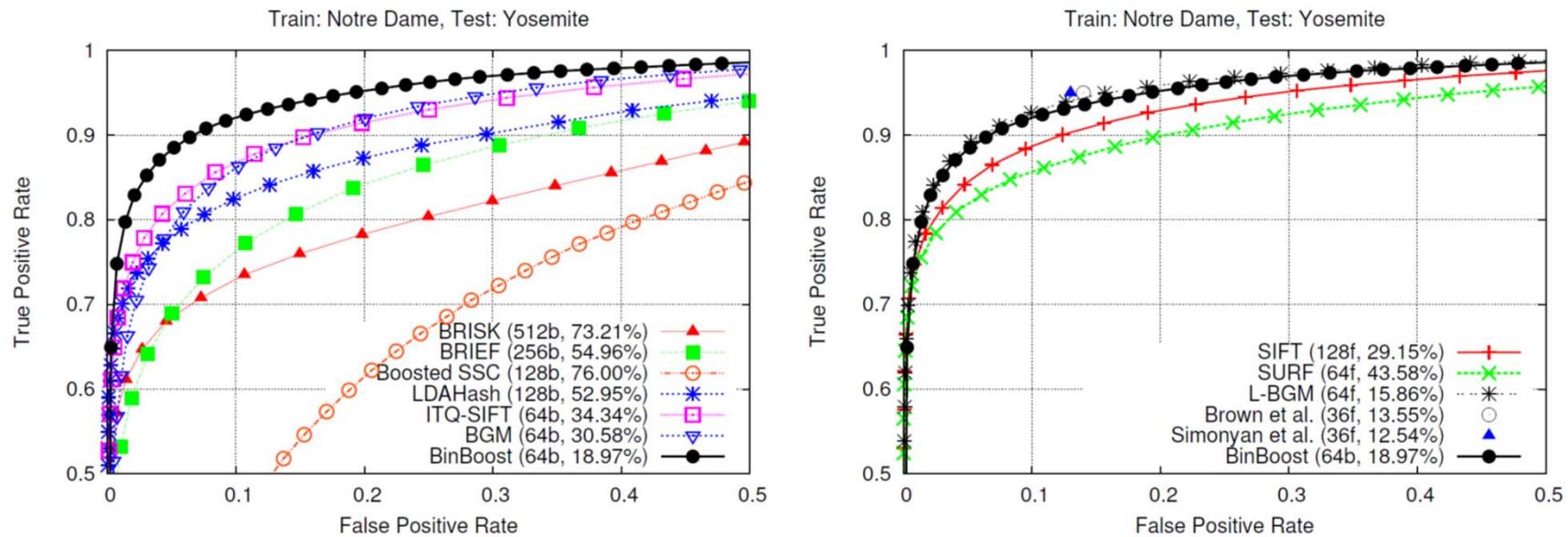


Figure 7. Comparison of our BinBoost descriptor to the state-of-the-art binary (**left**) and floating-point (**right**) descriptors. In parentheses: the number of floating-point (f) or binary (b) dimensions and the 95% error rate. Our BinBoost descriptor significantly outperforms its binary competitors for all false positive rates. It also outperforms SIFT and provides similar performances to the recent floating-point descriptors, even though it is much faster to match and has a lower memory footprint. More results are shown in the supplementary material.