
知能制御システム学

画像追跡 (1)
— 特徴点の検出と追跡 —

東北大学 大学院情報科学研究科

鏡 慎吾

swk(at)ic.is.tohoku.ac.jp

2010.07.13

今日の内容

前回までの基本的な画像処理の例を踏まえて、ビジュアルサーボシステムの構成要素となる画像追跡の代表的手法を概説する

画像上の「ある点」の追跡

- オプティカルフローの拘束式
- 追跡しやすい点 (Harris オペレータ)
- Lucas-Kanade 法 (KLTトラッカ)

点の追跡

移動ベクトル:

ある画像フレーム内の「ある点」が次フレームでどこに移動しているか

オプティカルフロー:

移動ベクトルの分布

- 両者を区別せずに使う場合もある
- いま我々が興味を持っているのは「移動ベクトル」だが、必要に応じて「オプティカルフロー」という用語も用いる(主に、そのトピックが「普通は」どの文脈で語られるかによる)

オプティカルフローの拘束式

追跡している点の明るさは変わらないとして,

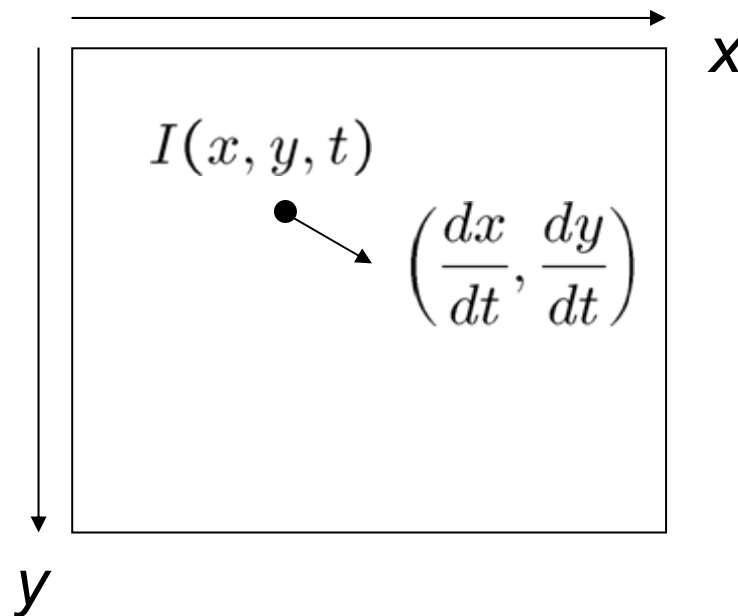
$$\begin{aligned} I(x, y, t) &= I(x + dx, y + dy, t + dt) \\ &= I(x, y, t) + \frac{\partial I}{\partial x} dx + \frac{\partial I}{\partial y} dy + \frac{\partial I}{\partial t} dt + \epsilon \end{aligned}$$

2次以上の項を無視すると

$$\frac{\partial I}{\partial x} \frac{dx}{dt} + \frac{\partial I}{\partial y} \frac{dy}{dt} + \frac{\partial I}{\partial t} = 0$$

求めたいオプティカルフロー $\left(\frac{dx}{dt}, \frac{dy}{dt}\right)$
を定めるには拘束が足りない

[Horn and Schunck 1981]



物理的解釈

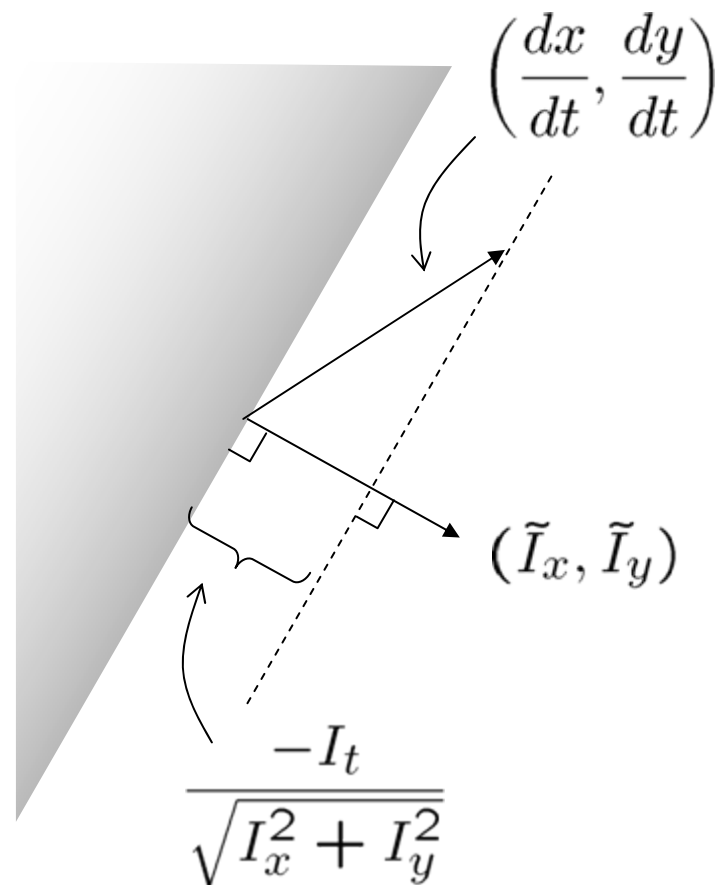
$$\frac{\partial I}{\partial x} = I_x, \quad \frac{\partial I}{\partial y} = I_y, \quad \frac{\partial I}{\partial t} = I_t \quad \text{と書いて}$$

$$(I_x, I_y) \left(\frac{dx}{dt}, \frac{dy}{dt} \right)^T = -I_t$$

$$(\tilde{I}_x, \tilde{I}_y) \left(\frac{dx}{dt}, \frac{dy}{dt} \right)^T = \frac{-I_t}{\sqrt{I_x^2 + I_y^2}}$$

フローの濃度勾配方向の成分は定まるが、それに垂直な成分は定まらない

直線エッジが動くのを小さな窓から覗いている状況を想像するとよい
(窓問題, aperture problem)



$(\tilde{I}_x, \tilde{I}_y)$ は (I_x, I_y) と同じ向きの単位ベクトル

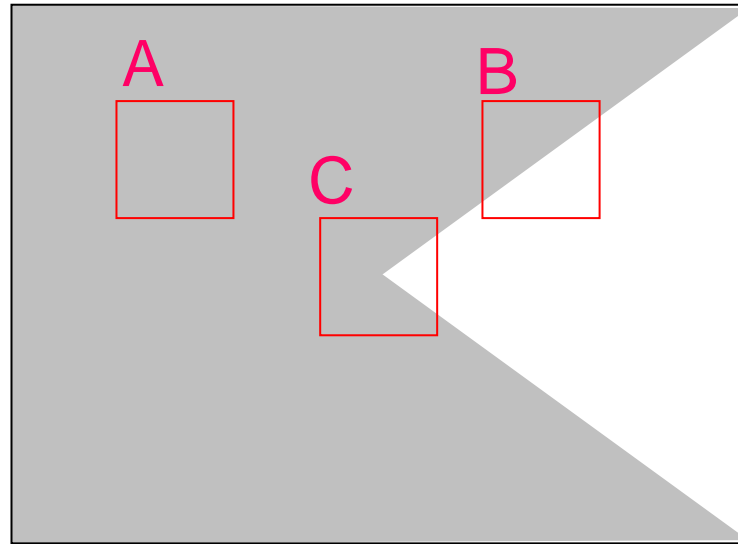
拘束条件の追加

従って、オプティカルフローは局所的な濃度勾配から完全に定めることはできなく、何らかの拘束条件を追加して考える必要がある

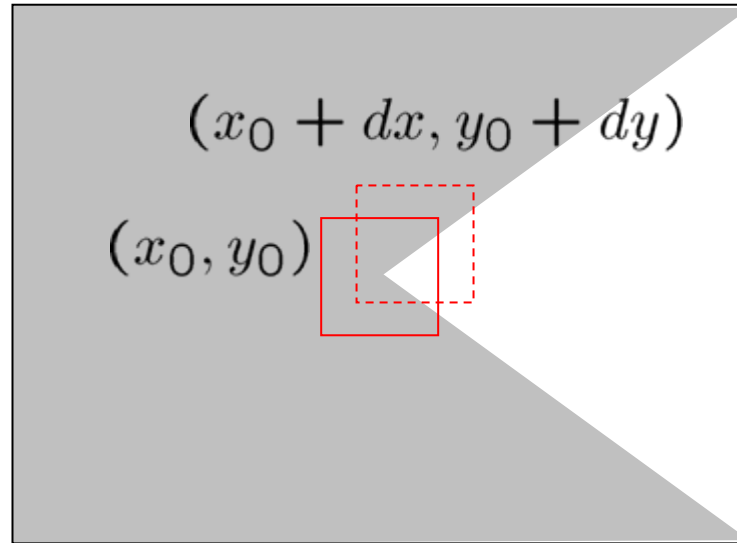
例1) オプティカルフローは空間的に滑らかに変化するという条件 [Horn and Schunck 1981]

例2) その点の周囲の小領域を考えて、その小領域内でのオプティカルフローは同一であるという条件
(以降で詳しく説明)

動きを求めやすい点の条件



- Aのように濃度が一様な領域では当然定まらない(拘束式が0本)
- Bのように一方向のエッジのみしか含まれない領域でも求まらない(拘束式が実質的に1本のみ)
- Cのように複数の方向のエッジ成分が含まれているような点でないと、フローは安定して定まらない → **どのように見つけるか?**



- ある点 (x_0, y_0) を中心とする小領域と,
- そこから少しだけ動いた $(x_0 + dx, y_0 + dy)$ を中心とする小領域を考える.

さまざまな微小変位 (dx, dy) のいずれに対しても,
両領域が似ていなければよい

画素ごとの差分の2乗和を「似てなさ」の指標 $E(dx, dy)$ とすると

$$E(dx, dy)$$

$$\equiv \sum_{u,v} \{I(x_0 + dx + u, y_0 + dy + v) - I(x_0 + u, y_0 + v)\}^2 w(u, v)$$

ただし $w(u, v)$ は適当な重みで例えばガウス関数とする。

Taylor展開して2次以降の項を無視すると

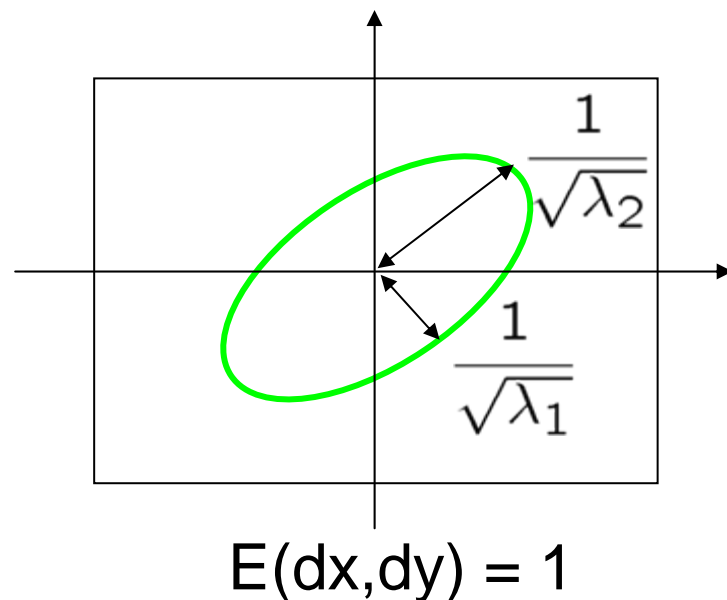
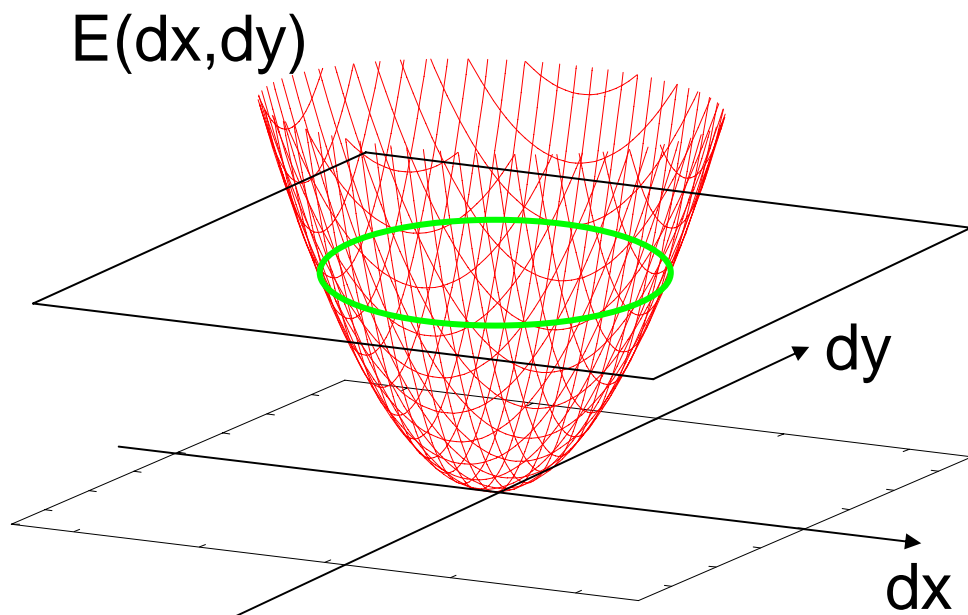
$$E(dx, dy)$$

$$= \sum_{u,v} \{I_x(x_0 + u, y_0 + v)dx + I_y(x_0 + u, y_0 + v)dy\}^2 w(u, v)$$

$$\sum_{u,v} I_x(x_0 + u, y_0 + v) = \sum I_x \text{ などと略記すると}$$

$$E(dx, dy) = \sum w I_x^2 dx^2 + 2 \sum w I_x I_y dx dy + \sum w I_y^2 dy^2$$

$$= (dx, dy) \begin{pmatrix} \sum w I_x^2 & \sum w I_x I_y \\ \sum w I_x I_y & \sum w I_y^2 \end{pmatrix} \begin{pmatrix} dx \\ dy \end{pmatrix}$$



$$E(dx, dy) = (dx, dy) \begin{pmatrix} \sum w I_x^2 & \sum w I_x I_y \\ \sum w I_x I_y & \sum w I_y^2 \end{pmatrix} \begin{pmatrix} dx \\ dy \end{pmatrix}$$

$$= (dx, dy) G \begin{pmatrix} dx \\ dy \end{pmatrix} = 1$$

なる楕円が小さくかつ真円に近いほどよい。すなわちGの固有値 λ_1, λ_2 がともに十分大きく、かつ互いと大きく違わないことが条件

(線形代数忘れた人用)

$$(dx, dy)G \begin{pmatrix} dx \\ dy \end{pmatrix} = 1$$

で G が対称行列であることに注意すると, G は直交行列 P (つまり $P^{-1} = P^T$) によって $P^T G P = \text{diag}(\lambda_1, \lambda_2)$ と対角化できる.

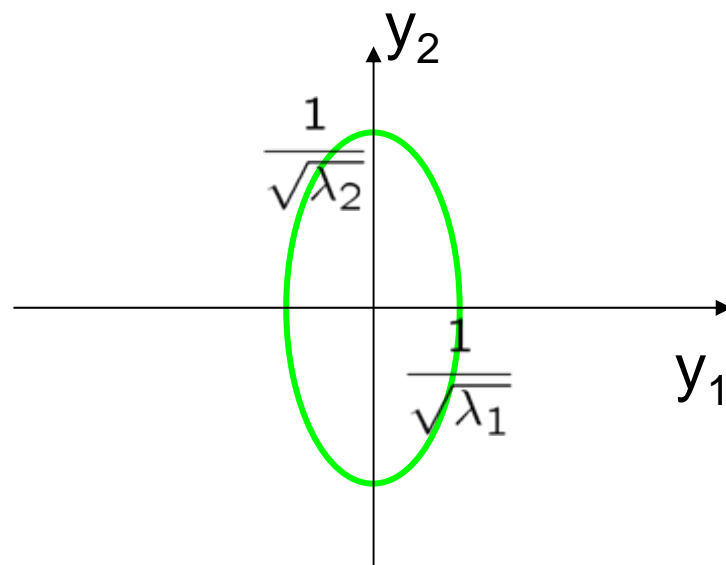
$$(dx, dy)P \begin{pmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{pmatrix} P^T \begin{pmatrix} dx \\ dy \end{pmatrix} = 1$$

$\mathbf{y} = P^T (dx, dy)^T$ の座標系で見ると

$$\mathbf{y}^T \begin{pmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{pmatrix} \mathbf{y} = 1$$

あるいは

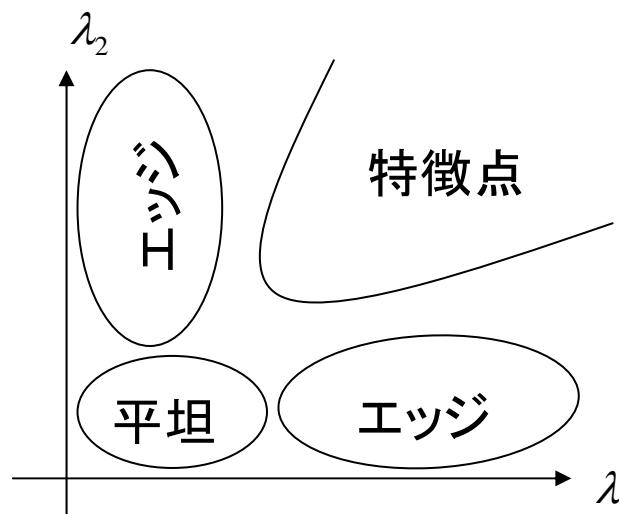
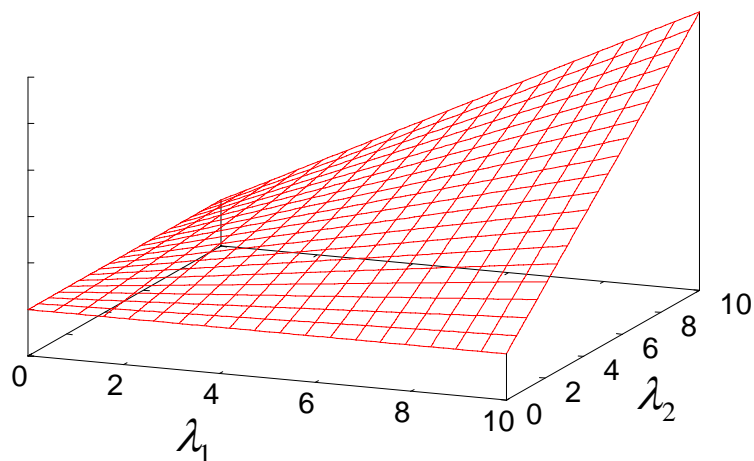
$$\lambda_1 y_1^2 + \lambda_2 y_2^2 = 1$$



特徴点検出子

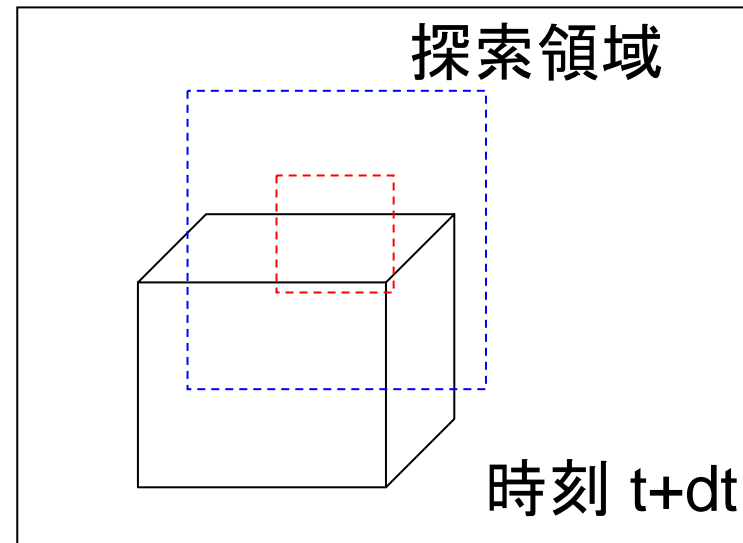
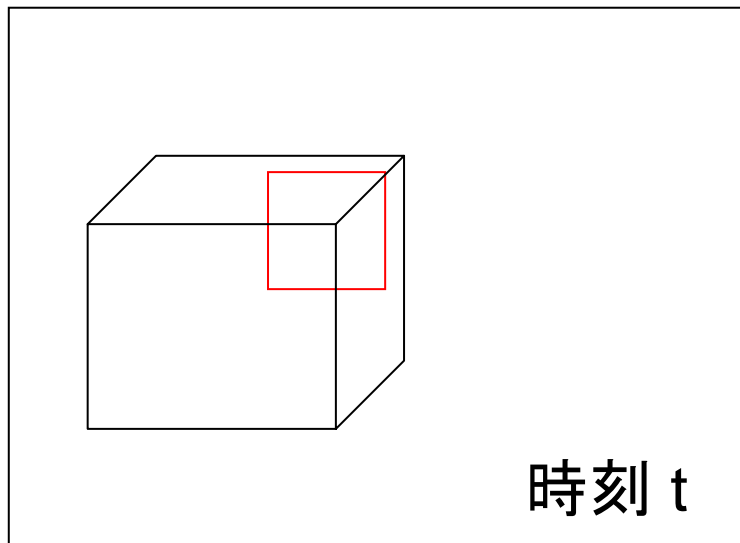
$$H(x_0, y_0) = \det G - k(\operatorname{tr} G)^2$$
$$= \lambda_1 \lambda_2 - k(\lambda_1 + \lambda_2)^2$$

: Harris オペレータ
[Harris and Stephens 1988]



このように「動きを計算しやすい点」「対応づけを得やすい点」を「特徴点」「キーポイント」「コーナ点」などと呼び、それを検出するフィルタを一般に feature point detector, interest operator などと呼ぶ

特徴点の追跡 — ブロックマッチング



領域内における画素値の差分2乗和, あるいは差分絶対和などの評価値を, 現在位置の周辺の各点で計算しながら, 最適な点を探索

→ 比較的安定して追跡できるが, 計算量が多い

「評価値」の代表例 (前時刻の参照領域を $T_{i,j}$ と書いて):

$$d_{\mathbf{SSD}}(x, y) = \sum_{i=0}^{m-1} \sum_{j=0}^{n-1} (T_{i,j} - I_{x+i,y+j})^2 \quad (\text{差分二乗和}) \text{を最小化}$$

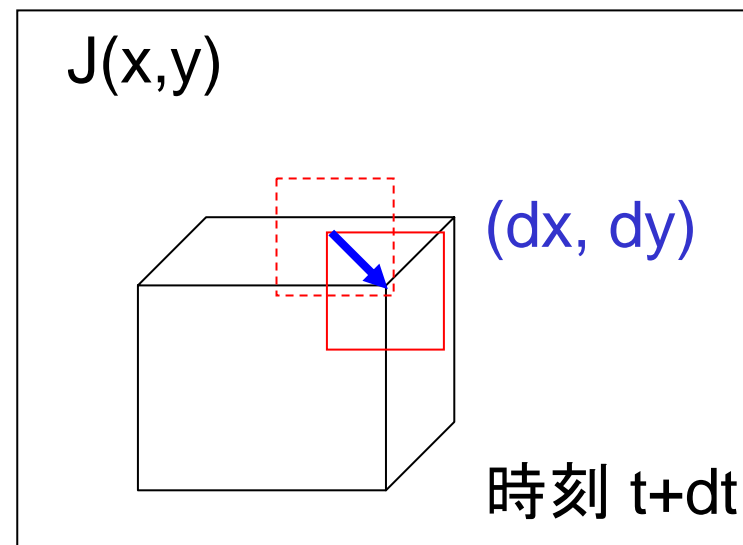
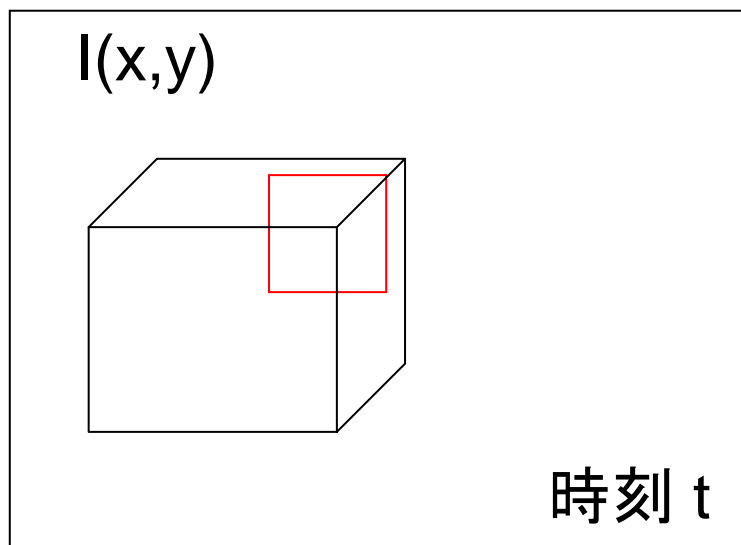
$$d_{\mathbf{SAD}}(x, y) = \sum_{i=0}^{m-1} \sum_{j=0}^{n-1} |T_{i,j} - I_{x+i,y+j}| \quad (\text{差分絶対和}) \text{を最小化}$$

$$C(x, y) = \sum_{i=0}^{m-1} \sum_{j=0}^{n-1} T_{i,j} I_{x+i,y+j} \quad (\text{相互相関}) \text{を最大化}$$

$$C_n(x, y) = \frac{\sum_{i=0}^{m-1} \sum_{j=0}^{n-1} (T_{i,j} - \bar{T})(I_{x+i,y+j} - \bar{I}_{x,y})}{\sqrt{\sum_{i=0}^{m-1} \sum_{j=0}^{n-1} (T_{i,j} - \bar{T})^2 \sum_{i=0}^{m-1} \sum_{j=0}^{n-1} (I_{x+i,y+j} - \bar{I}_{x,y})^2}}$$

(正規化相互相関)を最大化

特徴点の追跡 — 濃度勾配の利用



$$I(x, y, t) = I(x + dx, y + dy, t + dt)$$

$$I(x - dx, y - dy, t) = I(x, y, t + dt)$$

改めて $I(x, y, t) = I(x, y)$, $I(x, y, t + dt) = J(x, y)$ と書いて

$$I(x - dx, y - dy) = J(x, y)$$

小領域内で (dx, dy) が一定だとして、領域内の画素値の差分2乗和 ϵ が最小となるような (dx, dy) を求める

$$\begin{aligned}\epsilon &= \sum_{u,v} \{I(x - dx + u, y - dy + v) - J(x + u, y + v)\}^2 w(u, v) \\ &= \sum_{u,v} \{I - I_x dx - I_y dy - J\}^2 w \\ &= \sum \{h - I_x dx - I_y dy\}^2 w \quad (h \equiv I - J) \\ &= \sum \left\{ h - (I_x, I_y) \begin{pmatrix} dx \\ dy \end{pmatrix} \right\}^2 w\end{aligned}$$

(dx, dy) に関する微分が0になる条件から、

$$\sum \left\{ h - (I_x, I_y) \begin{pmatrix} dx \\ dy \end{pmatrix} \right\} \begin{pmatrix} I_x \\ I_y \end{pmatrix} w = 0$$

整理して,

$$\sum \begin{pmatrix} I_x^2 & I_x I_y \\ I_x I_y & I_y^2 \end{pmatrix} \begin{pmatrix} dx \\ dy \end{pmatrix} w = \sum h \begin{pmatrix} I_x \\ I_y \end{pmatrix} w$$
$$\begin{pmatrix} \sum w I_x^2 & \sum w I_x I_y \\ \sum w I_x I_y & \sum w I_y^2 \end{pmatrix} \begin{pmatrix} dx \\ dy \end{pmatrix} = \begin{pmatrix} \sum w (I - J) I_x \\ \sum w (I - J) I_y \end{pmatrix}$$
$$G \begin{pmatrix} dx \\ dy \end{pmatrix} = e$$

この方程式を解くことで (dx, dy) が得られる.

これを Lucas-Kanade 法と呼ぶ. [Lucas and Kanade 1981]

- このままでは (dx, dy) が十分に小さいときにしか正しく求まらないため, 実際にはこれを収束するまで繰り返す
- 平行移動だけではなく, より一般の座標変換へも対応できる

追跡可能な条件

安定した追跡は $G \begin{pmatrix} dx \\ dy \end{pmatrix} = e$ が安定して解けるときに可能となる

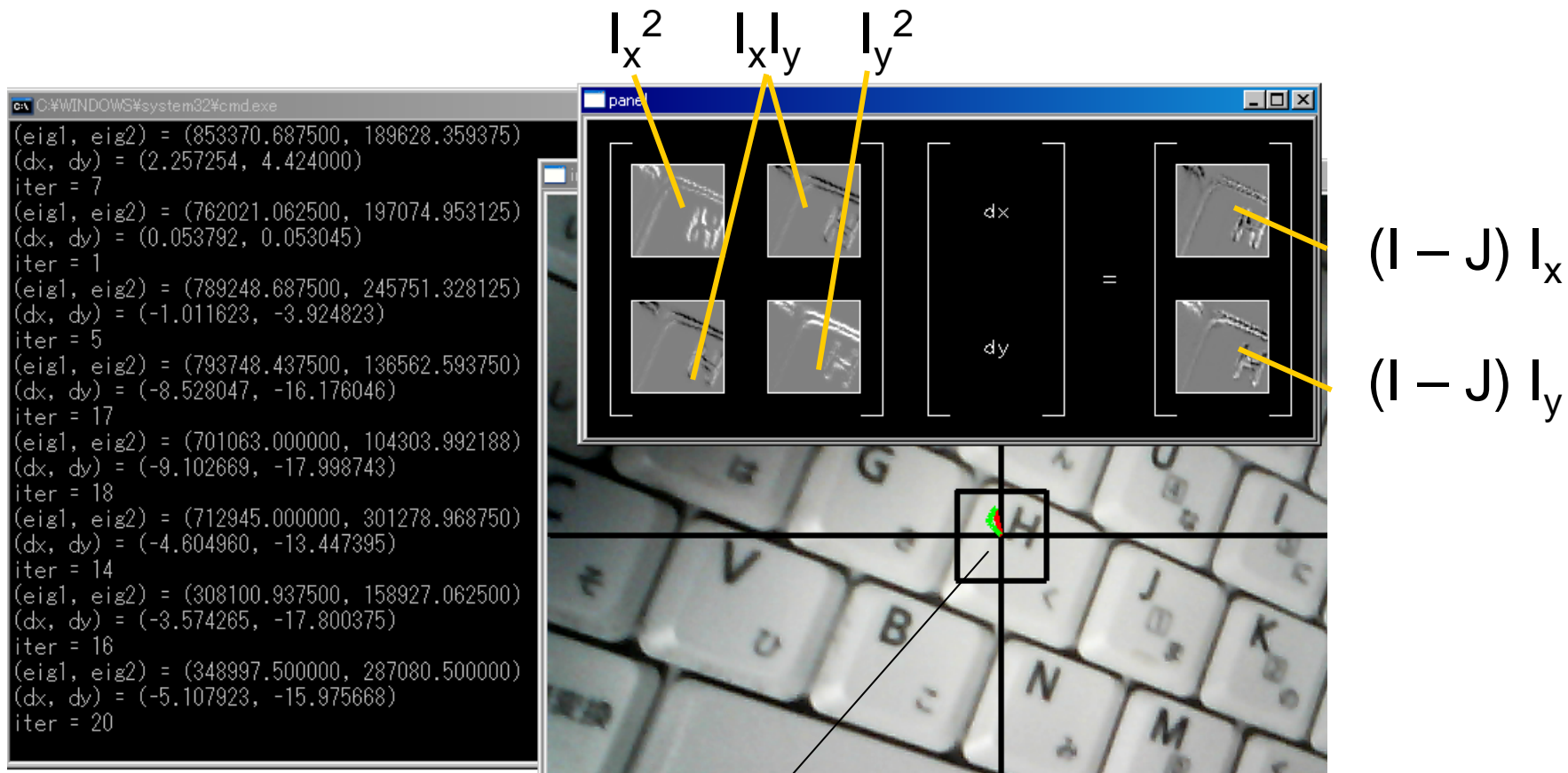
すなわち G の2つの固有値が十分に大きく、両者が大きく違わないことが条件となる → **Harris オペレータと等価**

実際には、画素値が有限なので2つの固有値に極端な差が出ることはなく、「 G の2つの固有値が十分に大きい」(= 小さい方の固有値があるしきい値以上) で実用上十分とされる
[Tomasi and Kanade 1991]

このように特徴点検出と追跡を統合した手法を Kanade-Lucas-Tomasi (KLT) トラッカと呼ぶ

サンプルプログラム1

画面中央に固定されたブロックのフレーム間移動ベクトルを計算する



赤: 移動ベクトルの計算結果

緑: 繰り返しごとの計算結果の軌跡

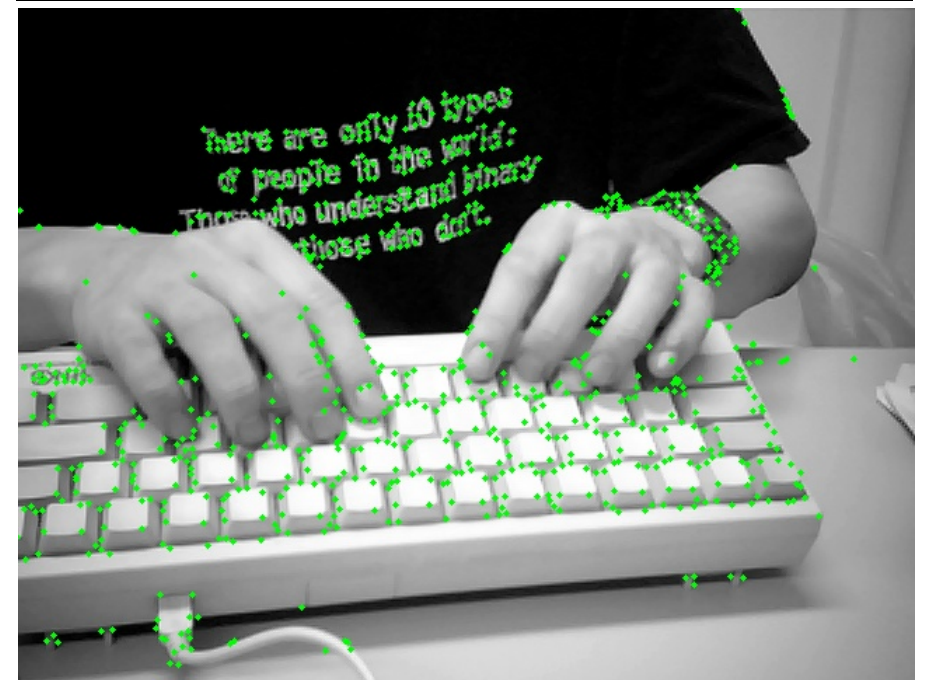
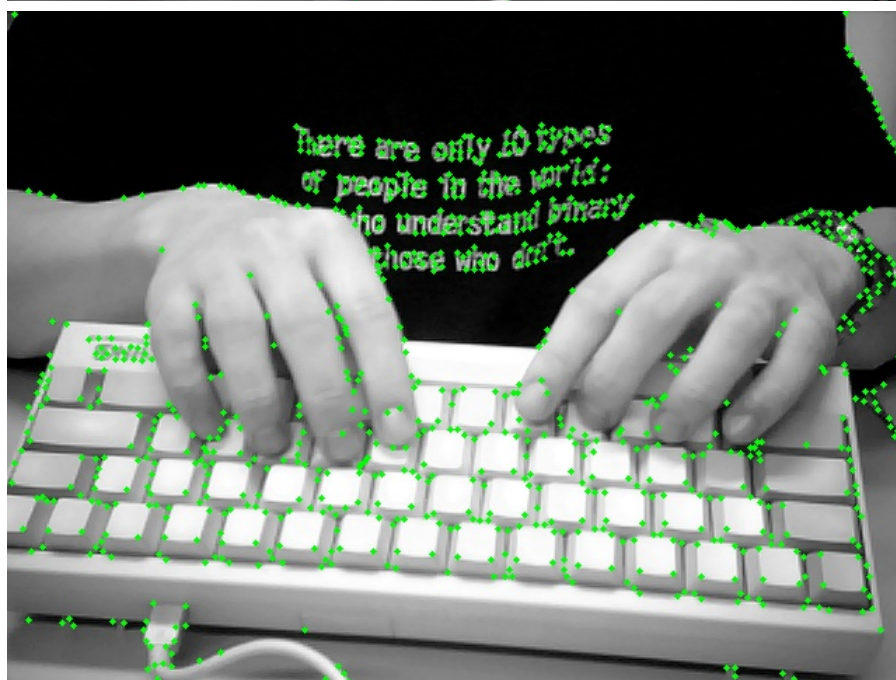
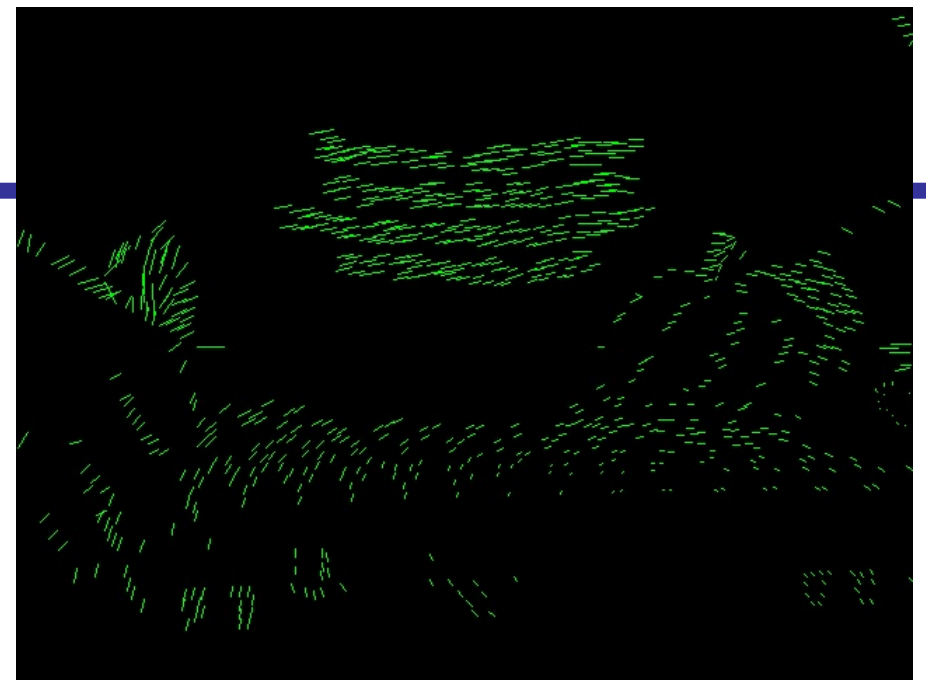
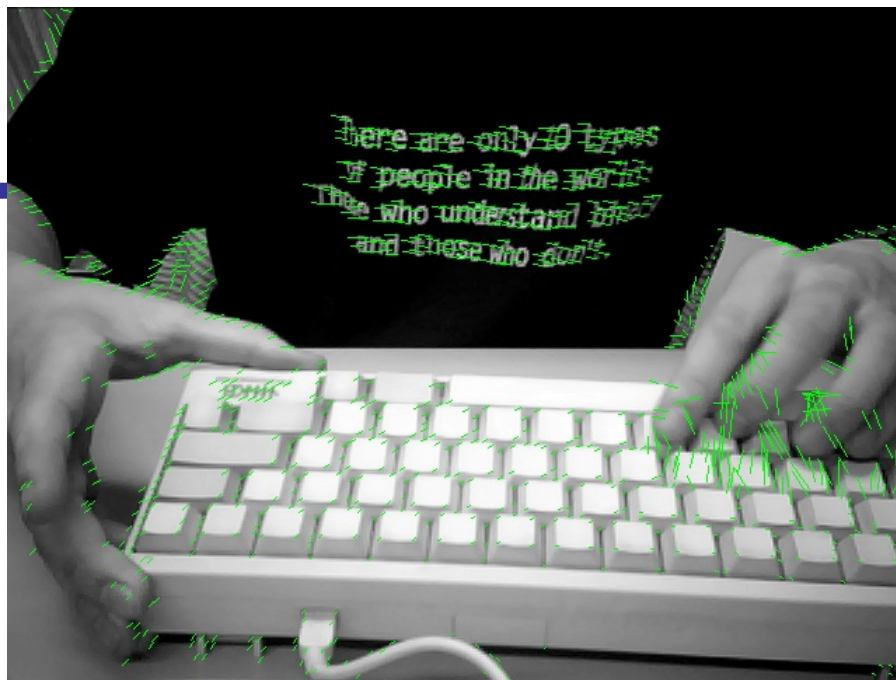
sample program:

- lk_illustrated.cpp

サンプルプログラム2

- 特徴点群の検出・追跡の例. OpenCVの標準関数に全面的に依存している(手抜き)
- 特徴点の選択は `cvGoodFeaturesToTrack()` で実行.
 - 第10引数によって Tomasi-Kanade の条件と Harris-Stephens の条件のどちらを使うか選べる.
 - 第5引数 `count` に検出された特徴点の数が返る. ただし, 呼出時の `count` の値が特徴点の最大数と扱われることに注意.
- 特徴点の追跡は `cvCalcOpticalFlowPyrLK()` で実行.
 - 今日説明したLucas-Kanade法を多重解像度画像に対して階層的に適用することで追跡精度を上げた手法が実装されている.

sample program:
• `klt_flow.cpp`



References

- B. K. P. Horn and B. G. Schunck: Determining Optical Flow, Artificial Intelligence, vol.17, pp.185-203, 1981.
- C. Harris and M. Stephens: A Combined Corner and Edge Detector, Proc. 14th Alvey Vision Conference, pp.147-151, 1988.
- B. D. Lucas and T. Kanade: An Iterative Image Registration Technique with an Application to Stereo Vision, Proc. 7th International Conference on Artificial Intelligence, pp.674-679, 1981.
- C. Tomasi and T. Kanade: Detection and Tracking of Point Features, Shape and Motion from Image Streams: a Factorization Method –Part 3, Technical Report CMU-CS-91-132, School of Computer Science, Carnegie Mellon University, 1991.