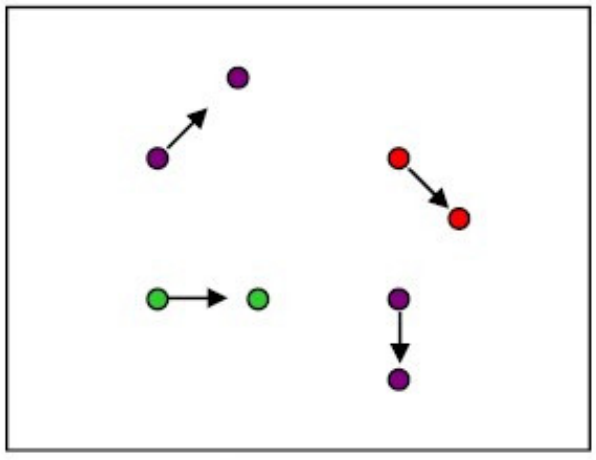
Flujo Óptico

Araguás, Gastón Redolfi, Javier

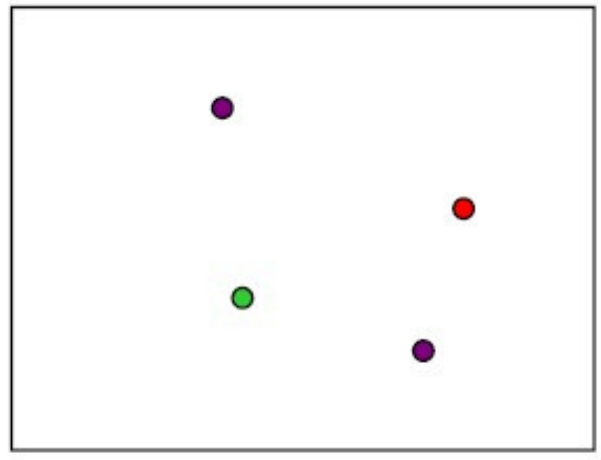
12 de Junio del 2019

Flujo Óptico

- Es el patrón aparente de movimiento de un objeto entre 2 imágenes



$I(x,y,t-1)$



$I(x,y,t)$

Lucas-Kanade Piramidal

- Permite seguir (trackear) puntos en una secuencia de imágenes (video).
- Los puntos los elegimos con **`cv2.goodFeaturesToTrack`** (esquinas de Shi-Tomasi) en la primera imagen.
- Luego estos puntos son trackeados usando el algoritmo de flujo óptico de Lucas-Kanade (**`cv2.calcOpticalFlowPyrLK`**).
- Esta función recibe la imagen previa, un conjunto de puntos definidos sobre ella y la imagen actual en donde queremos buscar el nuevo conjunto de puntos.
- Devuelve la nueva ubicación de los puntos, con un estado que indica si ese punto fue encontrado.
- Estos nuevos puntos, con la imagen actual son pasados en forma iterativa para encontrar los puntos en una nueva imagen.

Lucas-Kanade Piramidal - Código

```
#! /usr/bin/env python
# -*- coding: utf-8 -*-
import numpy as np
import cv2
cap = cv2.VideoCapture('vtest.avi')
feature_params = dict(maxCorners=100, qualityLevel=0.3, minDistance=7,
                      blockSize=7)
lk_params = dict(winSize=(15, 15), maxLevel=2,
                 criteria=(cv2.TERM_CRITERIA_EPS | cv2.TERM_CRITERIA_COUNT, 10, 0.03))
color = np.random.randint(0, 255, (100, 3))
ret, old_frame = cap.read()
old_gray = cv2.cvtColor(old_frame, cv2.COLOR_BGR2GRAY)
p0 = cv2.goodFeaturesToTrack(old_gray, mask=None, **feature_params)
mask = np.zeros_like(old_frame)

while(1):
    ret, frame = cap.read()
    frame_gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
    p1, st, err = cv2.calcOpticalFlowPyrLK(old_gray, frame_gray, p0, None, **lk_params)
    good_new = p1[st == 1]
    good_old = p0[st == 1]

    for i, (new, old) in enumerate(zip(good_new, good_old)):
        a, b = new.ravel()
        c, d = old.ravel()
        mask = cv2.line(mask, (a, b), (c, d), color[i].tolist(), 2)
        frame = cv2.circle(frame, (a, b), 5, color[i].tolist(), -1)

    img = cv2.add(frame, mask)
    cv2.imshow('frame', img)
    k = cv2.waitKey(30) & 0xff
    if(k == 27):
        break

    old_gray = frame_gray.copy()
    p0 = good_new.reshape(-1, 1, 2)

cv2.destroyAllWindows()
cap.release()
```

Tracking

- Consiste en localizar un objeto en imágenes sucesivas de un video.
- Si logramos seguir al objeto varias imágenes podemos construír:
  - Un modelo de movimiento
    - Velocidad
    - Dirección
    - Filtro de Kalman u otro
  - Un modelo de apariencia
    - Template
    - Clasificador
- Existen distintos tipos de trackers:
  - Basados en Flujo Óptico
  - Basados en Detección de Objetos
  - Combinación de Ambos

Trackers de las OpenCV

- Las **opencv** contienen varios trackers,
  - BOOSTING:
  - MIL: Multiple Instance Learning
  - KCF: Kernelized Correlation Filters
  - MF: Median Flow
  - TLD: Tracking-Learning-Detection
  - GOTURN:

BOOSTING

- La idea es construir un clasificador robusto con varios clasificadores débiles.
- La bounding box inicial se toma como muestra positiva.
- Se buscan muestras negativas fuera de la bounding box.
- Se entrena en forma online un clasificador.

MIL

- Similar al anterior, pero se crean más muestras positivas desplazando levemente la bounding box.
- Se forma una bolsa de muestras positivas.
- Se buscan muestras negativas fuera de la bounding box.
- Se entrena en forma online un clasificador.

KCF

- Basado en filtros de correlación.

MF

- Median Flow o mediana del flujo.
- Está basado en Lucas-Kanade.
- El cómputo de la bounding box se basa en la mediana.
- Agrega el concepto de fordward-backward error, esto sería aplicar tracking de puntos hacia atrás y hacia adelante.

TLD

- Tracking-Learning-Detection.
- Cambinación de Median Flow con un algoritmo de detección y aprendizaje.
- La parte de detección contiene varios clasificadores en cascada.

GOTURN

- Basado en CNN.
- Aprende un modelo offline.
- No lo pude correr en las OpenCV…

¿Cuál es el mejor?

ver imgs/real-time-face-tracking.gif

Trackers de las OpenCV - Código

No entra en la pantalla, lo vemos en **`vim`**.