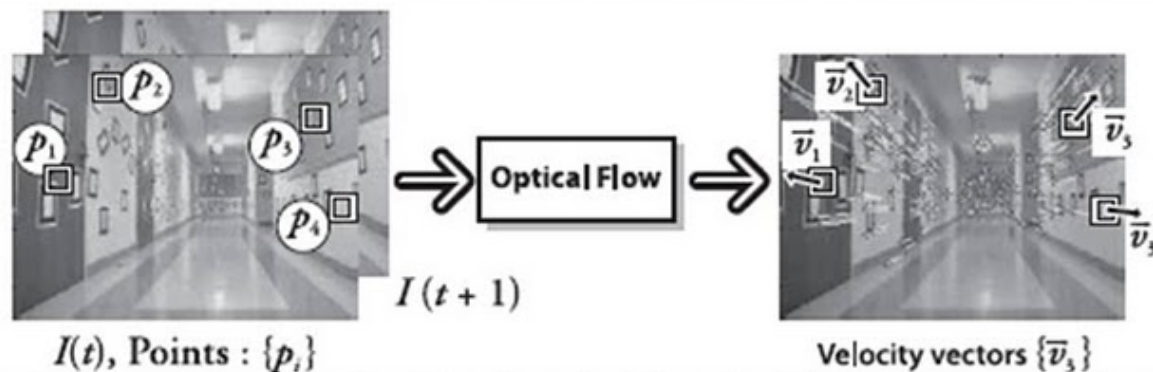


Оптический поток

Оценка вектора скорости. Оптический поток

lkdemo.exe

$$\sum_{x=p_x-\omega_x}^{p_x+\omega_x} \sum_{y=p_y-\omega_y}^{p_y+\omega_y} (A(x, y) - B(x + \nu_x, y + \nu_y))^2 \longrightarrow \min$$
$$\bar{\nu} = [\bar{\nu}_x \ \bar{\nu}_y]^T$$



picture from:
Gary Bradski, Adrian Kaehler, "Learning OpenCV:
Computer Vision with the OpenCV Library"

Оптический поток

```
void calcOpticalFlowPyrLK(const Mat& prevImg, const Mat& nextImg,  
    const vector<Point2f>& prevPts, vector<Point2f>& nextPts,  
    vector<uchar>& status, vector<float>& err, Size winSize=Size(15,15),  
    int maxLevel=3, TermCriteria  
    criteria=TermCriteria(TermCriteria::COUNT+TermCriteria::EPS, 30,  
    0.01), double derivLambda=0.5, int flags=0 );
```

- **status** - элемент вектора равен 1, если удалось подсчитать оптический поток, 0 иначе
- **err** - каждый элемент содержит разницу между патчами текущего и предыдущего изображения
- **winSize** - размер окна
- **maxLevel** - максимальный уровень пирамиды (от 0), 0 - не использовать пирамиду, 1 - два слоя пирамиды, ...
- **criteria** - критерий завершения алгоритма итеративного поиска
- **derivLambda** - удельный вес влияния производных изображения на оптический поток. Если равен 0, используются только интенсивности, derivLambda=1 - только производные используются, и т.д.
- **flags** - флаги операции: OPTFLOW_USE_INITIAL_FLOW

```
void goodFeaturesToTrack( const Mat& image,  
    vector<Point2f>& corners, int maxCorners, double  
    qualityLevel, double minDistance, const Mat&  
    mask=Mat(), int blockSize=3, bool  
    useHarrisDetector=false, double k=0.04 );
```

- `maxCorners` - максимальное количество возвращаемых точек
- `qualityLevel` - пусть лучшей точке соответствует отклик 1500 и `qualityLevel=0.01`, тогда точки с откликом больше 15 будут возвращены
- `minDistance` - минимальная евклидова дистанция между возвращаемыми точками
- `blockSize` - используемая окрестность пиксела

- **Background – foreground subtraction**
- **Mean filter**

$$B(x, y) = \frac{1}{N} \sum_{i=1}^N V(x, y, t - i)$$

$$|V(x, y, t) - B(x, y)| > Th$$

- N – количество предыдущих изображений, которые используются для усреднения
- N зависит от видео потока
- Можно использовать медиану вместо среднего значения
- Можно использовать только те значения пикселей, где нет движения по оптическому потоку

Object tracking

```
struct Object
```

```
{
```

```
    Rect bb;
```

```
    int non_detected_time; - выход объекта из кадра
```

```
    int life_time; -избавление от false positives
```

```
};
```