

视频分析与对象跟踪

背景消除建模

BS基本原理：

- 图像分割（GMM），靠干扰性强。
- 机器学习（KNN）

适合于背景不常常运动的情况。

对象检测与跟踪（基于颜色）

- inRange()过滤
- 形态学操作提取
- 轮廓查找
- 外接矩形获取
- 位置标定

光流的对象与跟踪

- 基本原理

移动对象跟踪的三要素：

- 图像表示
 - 外观模型
 - 移动模型
- 稀疏光流-KLT

※需要对特征点进行检测。

※一些假设

- 亮度恒定——检测角点（取二阶导数，来达到亮度不变性）
- 近距离移动——在不同空间进行检测
- 空间一致性——物体本身会有变换，旋转等——从数学的角度加以保证

※步骤

- 首先需要对图像（第一帧图像）进行特征点检测（程序例子中进行的是Shi-Tomas角点检测），并保存。
- 从第二帧开始——开始追踪
- 跟踪特征点
- 删除损失特征点
- --LOOP START--
- 保存跟踪特征点
- 用第二帧图像替换第一帧图像
- 用后续输入帧替换第二帧
- 选择新的特征点来代替损失特征点数据
- 保存特征点

- --LOOP END--

※ API : `calcOpticalFlowPyrLK()`

追跡する点を決めるために **goodFeaturesToTrack()** 関数を使います。

1枚目の画像を撮影し、Shi-Tomasiのコーナーを検出します。それ以降、Lucas-Kanade法を使ってこれらの点を繰り返し追跡します。

関数 **calcOpticalFlowPyrLK()** を使う場合、前フレーム、前フレームでの店の位置、現フレームを入力します。

返戻値は次のフレームでの点の位置と状態変数です。状態変数は次の画像中で点が見つければ1、そうでなければ0になります。

新しく検出した点を更に次のフレームでの入力に使用し、この処理を繰り返し行います。

- 稠密光流-HF (Dense Optical Flow) 很少使用

Lucas-Kanade法が出力するオプティカルフローは疎な特徴です(上記のコードではShi-Tomasiアルゴリズムによって検出したコーナーです)。

OpenCVは密なオプティカルフローを検出するためのアルゴリズムを別に用意しています。

- API : `calcOpticalFlowFarneback()`

画像中の全画素に対してオプティカルフローを計算します。Gunner Farnebackが2003年に発表した“Two-Frame Motion Estimation Based on Polynomial Expansion”で提案されたアルゴリズムに基づいています。

出力として、オプティカルフローベクトル (u, v) を格納した2チャンネルの配列が返ってきます。オプティカルフローの強度と方向を計算し、カラーコード化した画像を下に示します。方向はHue成分、強度はValue成分によって表されています。

某些場合会有良好表现，且不会出现跟踪丢失的情况。

CAMShift対象跟踪

- MeanShift算法介绍

通过均值计算评估中心点，迭代位移的过程。

Meanshiftアルゴリズムのアイディアの直感的理解は単純です。

点の集合(ヒストグラムの逆投影法のような画素の分布など)があるとして、

あなたのタスクは、与えられた小さなウィンドウを移動し、画素の分布密度(もしくは画素数)が最大になる領域にウィンドウの位置を合わせることです。

普通はヒストグラムを逆投影した画像と対象物体の初期位置を指定します。物体が動くと、その動きはヒストグラムを逆投影した画像中にはっきりと反映されます。結果として、meanshiftアルゴリズムはウィンドウを密度が最大となる場所へと移動させます。

- CanShift算法介绍

MeanShift 问题点：追跡対象である車が近づいてきているのに対して、追跡に使用したウィンドウは常に同じサイズです。これはよくありません。追跡対象の見え方によってウィンドウのサイズを調節する必要があります。

まず初めにmeanshiftアルゴリズムを適用します。meanshiftアルゴリズムの計算が収束したら $s = 2 \times \sqrt{\frac{M_{00}}{256}}$ という風にウィンドウサイズを変更します。同時に対象物体に最もフィットする楕円の回転角を計算します。再びmeanshiftアルゴリズムを適用しますが、サイズを変更したウィンドウと前回の収束場所から計算を始めます。このように、meanshiftアルゴリズムとウィンドウサイズの更新を収束条件が満たされるまで繰り返すのがCAMshiftアルゴリズムです。

- 工作流程

- 1, 是第一帧? 2:4

- 2, 选择ROI区域

- 3, HSV空间H通道的直方图表达

RGB空间的图像对光照变化比较敏感, 为了减少此变化对跟踪效果的影响, 首先将图像从RGB空间转到HSV空间。

然后对H分量做直方图, 得到不同H分量出现的次数即概率, 即得到了颜色概率查找表。

- 4, 直方图反向映射

对图像中每个像素值用其颜色出现的概率对替换, 得到颜色概率分布图。-> 反向映射 (灰度图)

- 5, CAMShift位置跟踪

meanshift算法是一种密度函数梯度估计的非参数方法, 通过迭代寻优找到概率分布的极值来定位目标。

* meanshift-> 单张图像 camshift-> 视频

- 6, 绘制位置更新显示

总结: camshift能有效解决目标变形和遮挡的问题, 对系统资源要求不高, 时间复杂度低, 在简单背景下能够取得良好的跟踪效果。但当背景较为复杂, 或者有许多与目标颜色相似像素干扰的情况下, 会导致跟踪失败。因为它单纯的考虑颜色直方图, 忽略了目标的空间分布特性, 所以这种情况下需加入对跟踪目标的预测算法。

视频中移动对象统计

- 基本思想与步骤

基于BSM (背景差分) 模型

提取前景ROI区域对象轮廓

排除干扰与统计

扩展模块中的跟踪方法介绍

- 略

扩展模块中的多对象跟踪

- 略

--END

