

四川大學

# 本科生毕业论文（设计）论文



题    目 基于光流法和背景减除法的运动目标

检测

学    院 电气工程学院

专    业 自动化

学生姓名 许高阳

学    号 2018141442002 年级 2018

指导教师 涂海燕

教务处制表

二〇二二年四月二十八日

# 基于光流法和背景减除法的运动目标检测

专业名称 自动化

学生 许高阳 指导老师 涂海燕

**[摘要]** 运动目标检测是机器视觉中的一个热门研究方向，在产品检测、医学图像分析等领域均有应用。本文针对光流法和单高斯背景减除法环境要求苛刻、易受初始图象序列的缺点，将光流法与背景减除法结合以实现算法的改进。首先，本文重点阐述了光流法与背景减除法的基本原理。接下来，完成了基础的金字塔型 LK 光流法，在光流的基础上进一步根据光流延伸焦点和光流向量幅值对三维场景的运动信息进行计算，结果证明该方法可以用于动态背景下的运动目标检测。然后，实现了基础的单高斯背景减除法，将该算法提出的轮廓简化为凸包的形式，通过判断光流法得到的运动目标坐标点与凸包的关系筛选轮廓。最后，利用 SBMnet 数据集对改进算法进行了试验，试验结果表明改进算法相较于单高斯背景减除法在性能上有较大的提升。

**[关键词]** 运动目标检测；光流法；背景减除法

# Moving target detection based on optical flow and background subtraction

Automation

Student: Xu Gaoyang Adviser: Tu HaiYan

**[Abstract]** Moving target detection is a popular direction in machine vision and it has been applied in product detection, medical image analysis and other fields. In this paper, the optical flow method is combined with the background subtraction method to improve the algorithms, which are vulnerable to the initial image sequence and environment. First, the basic principles of optical flow method and background subtraction method are introduced. Then, the pyramid LK optical flow method is implemented. On the basis of the optical flow, the motion information of 3D scene is calculated according to the focus of expansion and the amplitude of optical flow. The result shows that this method can be used for moving target detection in dynamic background. Then, the basic single Gaussian background subtraction method is implemented, and the contours proposed by the algorithm are simplified to the convex hulls. According to the positional relationship between the coordinates obtained from optical flow and the convex hulls, the contours are screened. Finally, the SBMnet dataset is used to test the proposed algorithm. As the result shows, the proposed algorithm has better performance than the single Gaussian background subtraction method.

# 目 录

<b>1 综述 .....</b>	<b>1</b>
1.1 研究背景和意义.....	1
1.2 国内外研究现状.....	1
1.2.1 光流法研究现状.....	1
1.2.2 背景减除法研究现状.....	2
1.2.3 神经网络运动目标检测方法研究现状.....	3
1.2.4 运动目标检测算法评估方法研究现状.....	3
1.3 本文研究的主要内容概述.....	3
<b>2 理论背景 .....</b>	<b>5</b>
2.1 OPENCv 机器视觉库.....	5
2.2 光流法理论背景.....	5
2.2.1 光流.....	5
2.2.2 光流场.....	6
2.3 光流法基本原理.....	6
2.3.1 光流法的基本假设条件.....	6
2.3.2 光流法的基本约束方程.....	6
2.3.3 光流法的分类.....	7
2.4 LUCAS-KANADE 光流法 .....	7
2.5 背景减除法基本原理 .....	9
2.6 单高斯背景建模法 .....	9
2.7 本章小结 .....	10
<b>3 光流法的实现及改进.....</b>	<b>11</b>
3.1 图像序列读取.....	11
3.2 角点检测.....	11

3.3 光流计算 .....	13
3.4 基于光流法的三维场景信息计算.....	15
3.4.1 光流延伸焦点 FOE.....	15
3.4.2 Kmeans 算法计算 FOE.....	16
3.4.3 光流向量幅值检测运动目标.....	18
3.5 本章小结 .....	20
<b>4 静态背景运动目标检测 .....</b>	<b>21</b>
4.1 单高斯模型背景建模 .....	21
4.1.1 使用 accumulateWeighted 函数计算均值.....	21
4.1.2 使用 accumulateSquare 函数计算方差.....	21
4.2 背景差分.....	22
4.3 光流法-单高斯背景减除法.....	23
4.3.1 单高斯背景模型建模局限性.....	23
4.3.2 光流法-单高斯背景减除法基本思路.....	24
4.3.3 轮廓提取.....	25
4.3.4 轮廓筛选.....	26
4.4 本章小结 .....	28
<b>5 算法测试 .....</b>	<b>29</b>
<b>6 总结与展望.....</b>	<b>32</b>
6.1 总结 .....	32
6.2 展望 .....	32
<b>参考文献 .....</b>	<b>33</b>
<b>声 明 .....</b>	<b>36</b>
<b>致 谢 .....</b>	<b>37</b>
<b>附录 1.....</b>	<b>38</b>

# 1 综述

## 1.1 研究背景和意义

视觉,是人们获取外部信息的主要渠道之一,是人们了解和感受自然界的主要途径之一。在一些特殊的场合,人类往往会通过视觉来捕捉环境中的运动物体。例如,在高速公路,需要通过视觉捕捉运动的车辆;在机场,需要通过视觉来捕捉跑道上有无可能影响起飞或下降的运动物体。在这些场合中,能否检测到环境中的运动对象将影响到人类的安全。因此在生活中对运动目标的检测十分重要。

在计算机技术蓬勃发展的今天,通过计算机视觉代替人眼,来捕捉环境中的运动对象,已经成为了一个十分热门的话题。所谓计算机视觉,就是计算机为了获取关于环境中事物的信息,处理摄像机等光学传感器反馈信息的过程。类比视觉对于人类的重要性,计算机视觉对于计算机的意义不言而喻。如今,产品检测<sup>[1]</sup>、机器视觉导航<sup>[2]</sup>、医学图像分析<sup>[3]</sup>等应用成果说明计算机视觉技术早已渗透了人类日常生活的每个角落。运动目标检测<sup>[4]</sup>指的是将由多帧图像序列组成的视频中发生空间位置变化的目标作为前景提取出来并进行表示的过程,是计算机视觉技术的重要组成部分之一。

目前已知的移动目标监测方法大致包括三种,分别是帧间差分法<sup>[4]</sup>、光流法<sup>[5]</sup>、背景减除法<sup>[6]</sup>,而三个类方法都有着各自的特点。帧差分法具有较好的即时性,算法结构简单、计算量小,但对周围环境噪声比较敏感,且对于视频中体积较大且单色的运动目标,可能会在目标轮廓内部产生空洞而造成无法完整地提取运动目标;光流法无需预知场景的任何信息即可检测到运动目标,在背景发生运动的情况下也可以完成运动目标检测的任务,不过由于对环境中亮度的要求较苛刻,实用性较差,此外光流法计算复杂,实时性也难以得到保证;背景减除法对背景建模及更新的算法具有很强的依赖性,当背景较为准确时效果较好,但背景发生动态变化时效果无法得到保证,需要更复杂的背景建模算法来进行辅助,同时也增加了计算量。

综上所述,由于目前的三类运动目标检测算法在环境相对严苛时通常都无法取得理想的成效,所以在现有的算法上对继续进行完善以提高运动目标检测计算的鲁棒性与准确度,有着一定的现实意义和使用价值。

## 1.2 国内外研究现状

### 1.2.1 光流法研究现状

目前已经出现了各种类型的光流法,例如 LK 光流法<sup>[7]</sup>、HS 光流法<sup>[8]</sup>等。HS 光流法对图像中所有像素点的光流加以估算,该算法构建了一个能量函数,利用能量函数的最小值来计算光流。LK 光流法对图像中的部分像素点的光流进行估算,该算法对比相邻帧图像间的光流场,通过计算特征点的稀疏光流来代替所有像素点的光流。

光流法最大的局限性就在于其有着环境中亮度恒定且空间连续的假设, 实际环境往往难以满足这些较为苛刻的假设, 故光流法的实用性较差。关于这一问题, **Black** 等人给出了一种基于鲁棒估计框架的解决方案, 该框架有效降低了因不满足假设而产生的离群值<sup>[9]</sup>。**Serdar** 等人在计算光流向量的过程中利用多帧图像来计算被遮挡区域中的光流来识别障碍物, 进而解决了空间不连续性问题<sup>[10]</sup>。**Zach** 等人提出了一种利用总变化规律和鲁棒系数的替代方案, 以此保持光流场的连续性并解决了环境亮度发生变化对算法造成的负面影响<sup>[11]</sup>。光流法的另一个局限性是难以在初始阶段检测到目标, 由于光流法对与环境要求较高, 因此背景的检测很可能不够准确。**Simon** 等人提出了一种将运动分割与光流法相组合的算法, 通过扩展运动分割来计算光流向量并进行运动检测, 这种方法仅针对运动的像素点计算光流, 从而减少了计算量和离群值的出现, 但是该算法只能确定水平方向上的光流量, 垂直光流向量不够准确<sup>[12]</sup>。上述的算法虽然针对光流法的局限性进行了各种改进, 但仍然只适用于一些特定的环境, 在不同的环境下进行运动目标检测时仍具有一定的局限性。

### 1.2.2 背景减除法研究现状

背景减除法的关键在于如何获得准确的背景模型。最简单的方法是取视频中的一段图像帧直接作为背景。**Gloyer** 等人通过取像素值中值的方法对背景进行建模, 但该方法占用内存大, 计算量大, 且当环境亮度发生变化或环境为动态时背景模型的精度降低<sup>[13]</sup>。**Stauffer** 和 **Grimson** 共同提出混合高斯背景模型, 将用于每个像素点背景模型的正态分布个数提高至  $k$  个, 该算法具有更少的内存消耗率和良好的适应性, 故能够用来检测室外自然环境中的运动目标, 但同时大大提高了计算量<sup>[14]</sup>。为克服上述算法计算量过大的问题, **Zivkovic** 等人提出针对不同情况改变高斯模型数量的算法, 提升了算法的实时性<sup>[15]</sup>。王典还提出了一种更高效的混合高斯建模算法, 该算法只对背景中发生混乱或反复变换的区域进行像素点更新, 减少了混合高斯建模的工作量, 但这种算法需要预先将像素点分割为两类, 具有一定的难度<sup>[16]</sup>。**Elgammal** 等人提出的基于核密度估计模型的算法不需要预先设定概率分布模型, 也不需要进行复杂的参数模型的估计与优化, 但所需存储空间较大<sup>[17]</sup>。**Barnich** 等提出一种新的检测算法 **ViBe**, 该算法在视频序列的第一帧就按照一定规则随机选取像素点进行建模, 这种算法具有更高的检测效率但对第一帧中的前景干扰的抑制效果较差<sup>[18]</sup>。**Liu** 等人给出了一种通过空间梯度平均值来确定 **ViBe** 算法的检测半径, 并使用用空间梯度削弱阴影干扰<sup>[19]</sup>。张钦礼等人针对 **ViBe** 算法的局限性提出了一种基于中位数、分割度系数和空间一致性的改进 **ViBe** 算法<sup>[20]</sup>。**Yang** 等人提出一种分层背景模型法就是一种典型算法, 该算法结合纹理与颜色特征, 去除了检测目标的运动阴影<sup>[21]</sup>。**Wu** 等人提出的基于多分辨率的自适应阈值算法利用像素点与其邻域的关系来实现运动目标的检测<sup>[22]</sup>。

### 1.2.3 神经网络运动目标检测方法研究现状

近年来,随着人工智能的不断发展,深度学习被运用于运动目标检测这一领域。Braham 等人利用卷积神经网络(CNN)对不同视频训练出一个卷积神经网络来提取背景,之后 T-CNN 将目标检测与跟踪结合起来,增强了算法的性能<sup>[23]</sup>。

### 1.2.4 运动目标检测算法评估方法研究现状

尽管目前已经出现了众多运动目标检测算法,但是这些运动目标检测算法检测的结果没有一个统一的指标,因此目前仍然没有一个能够量化地去描述运动目标检测算法性能的评估标准,但是针对各种类型的运动目标检测算法,都已经各自有一些评估指标和标准。Elisa 等人提出了一种主观和客观分割质量评估的通用框架,通过主观实验,基于感知因素推导了一种用于分割评价的客观质量评价方法。所提出方法的性能在不同先进的前景/背景分割算法上进行了展示。此外,在主观结果的基础上,将加权策略引入到所提出的指标中,以满足视频压缩、视频监控和混合现实等不同分割应用的特殊性。实验结果验证了所提方法的有效性<sup>[24]</sup>。Correia 等人针对主观测试视频分割质量昂贵且耗时的问题,选择使用客观的分割质量评估技术。提出了单个对象和完整分割分区的评估方法和客观分割质量指标,为了涵盖缺少参考分割或用于比较的情况,开发了独立和相对评估指标<sup>[25]</sup>。Mech 等人针对无法区分位于检测对象轮廓周围的估计误差的问题,提出了一种基于二维形状估计误差的平均值和标准值对空间精度和时间相干性评估的评价方法<sup>[26]</sup>。

## 1.3 本文研究的主要内容概述

本文主要研究运动目标检测算法,首先针对光流法和背景减除法进行深入分析和改进,从最简单的运动目标检测算法理论入手,分析算法在实际应用过程中的局限性和其原因,运用数据分析知识对算法进行完善以期达到较为理想的检测性能,最终通过公开数据集对改进后的算法进行试验,将其与其他改进算法进行横向对比。本文的结构安排如下:

绪论讲述了本文研究课题的研究背景,并介绍了国内外运动目标检测算法的研究现状,最后对本文的内容结构进行简单安排;

第二章介绍了本文将会使用的软件平台和编程语言,光流法、背景减除法的基本原理和实现流程,分析了 LK 光流法和单高斯背景减除法自身存在的优劣性;

第三章介绍了动态背景下的运动目标检测,首先实现金字塔 LK 光流法,接着基于光流从光流延伸角点和光流向量幅值两个角度分别计算三维场景中的运动信息。

第四章介绍了静态背景下的运动目标检测,首先实现单高斯背景减除法,经过对结果的分析,将该算法与光流法结合以实现改进。



第五章使用 **SBMnet** 数据集对提出的光流-单高斯背景减除法进行。

## 2 理论背景

### 2.1 OpenCv 机器视觉库

OpenCv 是一个开源的跨平台机器视觉软件库，该库主要由大量 C 函数和 C++类构成，同时还能偶通过 Python、MATLAB 等语言使用。OpenCv 主要关注真实世界的实时应用，使用优化后的 C 语言为程序执行速度带来了可观的提升，OpenCv 与当前主流其他视觉函数库的性能对比如图 3-1 所示。

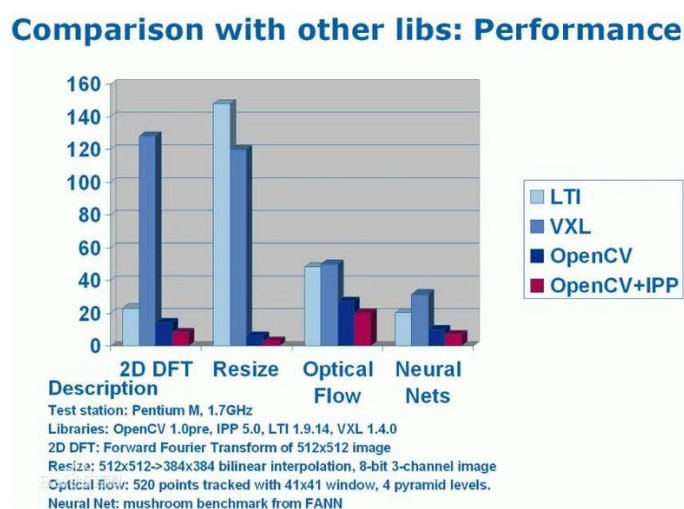


图 2-1 OpenCv 库其他函数库性能对比

从图 3-1 可以看到，OpenCv 的性能相较于其他视觉函数库更为强大，因此本文使用 OpenCv3.4.1 版本在 VisualStudio2017 中，使用 C++语言来实现算法。

### 2.2 光流法理论背景

#### 2.2.1 光流

光流(optical flow) 这一概念由 Gibson 在 1950 年首先提出，是指三维空间中的运动物体在二维成像平面上的像素的瞬时速度，光流利用图像序列中像素在时域上的变化和相邻帧之间的关联来寻找相邻帧中像素之间的对应关系，从而计算相邻帧中包含的运动目标信息。

就实际情况而言，因为无法保证相机的绝对静止，所以光流往往来自于场景中运动目标与相机本身的相对运动，故可以认为光流包含了运动目标的运动信息，可以被视为运动目标的一种特征。

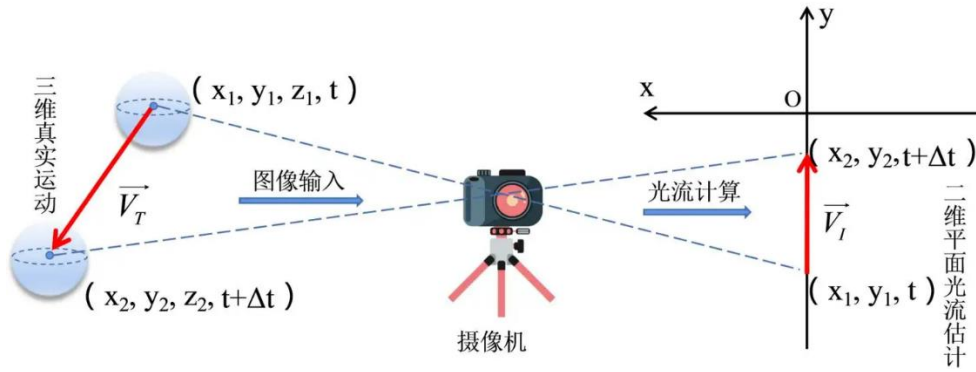


图 2-2 三维运动在二维平面的投影

### 2.2.2 光流场

在三维空间中，可以通过运动场来对运动目标进行描述，而在二维平面中，运动场则转化为图像序列中不同图像的灰度分布，也就是光流场。换句话说，光流场就是三维空间中运动场在二维平面的映射。

光流场是一个二维量场，每一条向量代表了图像中每一个像素点灰度值的变化，也可以认为是像素点在二维平面上运动而产生的瞬时速度场。通过研究光流场，就可以通过二维平面计算三维运动场。

## 2.3 光流法基本原理

### 2.3.1 光流法的基本假设条件

光流法有以下两个基本假设前提：

1. 亮度恒定不变，即同一目标在二维平面上的像素点灰度值不会随时间发生变化，这一条件是光流法基本约束方程的前提；
2. 时间连续且运动速度较小，即运动目标不会在短时间内不会发生较大的位移。

### 2.3.2 光流法的基本约束方程

设像素点在某一帧的光强度为  $I(x, y, t)$ ， $x$ ， $y$ ， $t$  分别为像素点在当前帧的位置坐标和时间，在下一帧中，该像素点产生了  $(dx, dy)$  的位移，且这两帧之间的时差为  $dt$ ，根据第一个假设前提，同一像素点的光强度并未发生改变，因此有式(2.1)：

$$I(x, y, t) = I(x + dx, y + dy, t + dt) \quad (2.1)$$

将式(2.1)右端进行泰勒展开，可以得到式(2.2)：

$$I(x + dx, y + dy, t + dt) = I(x, y, t) + \frac{\partial I}{\partial x} dx + \frac{\partial I}{\partial y} dy + \frac{\partial I}{\partial t} dt + \varepsilon \quad (2.2)$$

将式(2.2)右端中代表二阶无穷小项的  $\varepsilon$  项忽略不计，将式(2.2)代入式(2.1)的结果，之后在方程两端同时除以  $dt$ ，即可得到式(2.3)：

$$\frac{\partial I}{\partial x} \frac{dx}{dt} + \frac{\partial I}{\partial y} \frac{dy}{dt} + \frac{\partial I}{\partial t} \frac{dt}{dt} = 0 \quad (2.3)$$

令 $u$ ,  $v$ 分别为光流向量沿 $x$ 轴、 $y$ 轴的分量, 因此有式(2.4):

$$u = \frac{dx}{dt}, v = \frac{dy}{dt} \quad (2.4)$$

令 $I_x = \frac{\partial I}{\partial x}$ ,  $I_y = \frac{\partial I}{\partial y}$ ,  $I_t = \frac{\partial I}{\partial t}$ , 综合式(2.4)代入式(2.3), 可以得到式(2.5):

$$I_x u + I_y v + I_t = 0 \quad (2.5)$$

观察式(2.5)可以发现,  $I_x$ ,  $I_y$ ,  $I_t$ 均为可求量, 而 $(u, v)$ 即为所求的二维光流向量。

### 2.3.3 光流法的分类

因为光流向量存在两个分量, 且只有一个约束方程, 要想求得具体的光流向量, 就必须引入其他约束条件, 这个问题被称为光流法的光圈问题。基于约束条件的不同, 光流法也被细分为五种: 基于梯度(微分)的光流法、基于匹配的光流法、基于能量(频率)的光流法、基于相位的光流法和神经动力学光流法。

基于匹配的光流算法分为基于特征和基于区域两类。基于匹配的光流法对目标的剧烈运动和亮度变化有一定的鲁棒性, 但是计算的光流比较稀疏, 且计算亚像素精度的光流时有困难。

基于能量的光流法又称为基于频率的光流法, 该方法必须对图像进行时空滤波预处理, 这将会造成光流的计算精度降低, 而且运算工作量也很大。

基于相位的光流法处理图像序列的范围较广, 而且速度计算相对精确, 不过这种算法中所用的模型存在着很大的时间复杂度, 对图像序列的时间混叠也更为敏感。

神经动力学光流法使用了神经网络所构建的生物视觉运动感知的神经动力学模式, 该方法对生物视觉系统进行了仿真, 这是目前光流技术最具意义的发展方向。

基于梯度的光流法又称为微分法, 其基本原理是使用时变图像灰度的空间梯度函数来估计光流, 因为该方法的计算工作量相对较小而且计算结果也较好, 因此获得了普遍的应用, 其经典的代表方法是 Lucas-Kanade 光流法<sup>[7]</sup>和 Horn-Schunck 光流法<sup>[8]</sup>, LK 光流法是一个估计局部稀疏光流的算法, 使用特征点的稀疏光流来代替整个图像中的稠密光流。HS 光流法计算整个图像的稠密光流, 该算法构建了一个能量函数, 把光流的计算问题转换为求能量函数最小值的问题。为了减少计算量以提升算法的实时性, 本文采用 LK 光流法对运动目标进行检测。

## 2.4 Lucas-Kanade 光流法

由于 LK 光流法通过对图像中特征点的稀疏光流的估计代替对整个图像中稠密光流的估计, 为了保证光流计算的可靠, 在光流法的两个基本假设前提的基础上, LK 光流法给出了第三个假设: 空间一致性假设, 即整个场景中同一表面上相邻的点运动也相似, 而这些点在二维平面上的投影也都位于邻近区域。

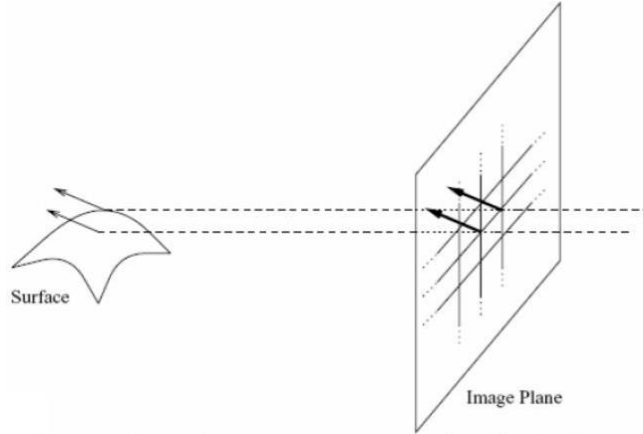


图 1-3 空间一致性假设

基于运动目标在短时间内不会发生剧烈位移的假设，对于某区域中的 $n$ 个像素点，每个像素点的光流都满足式(2.6)：

$$\begin{cases} I_{x_1}u + I_{y_1}v + I_{t_1} = 0 \\ I_{x_2}u + I_{y_2}v + I_{t_2} = 0 \\ \dots \\ I_{x_n}u + I_{y_n}v + I_{t_n} = 0 \end{cases} \quad (2.6)$$

基于亮度恒定假设，光流法基本约束方程的误差如式(2.7)：

$$E_c(u, v) = \iint [I_x u + I_y v + I_t]^2 dx dy \quad (2.7)$$

在区域内，LK 光流法的误差如式(2.8)：

$$E_{LK}(u, v) = \iint W^2(x, y) * [I_x u + I_y v + I_t]^2 dx dy \quad (2.8)$$

式(2.8)中， $W(x, y)$ 表示区域内每个像素的权重，该值由区域中心开始向区域边界递减。

使用矩阵来对式(2.8)进行表示，将光流的分量提取出来，可得式(2.9)：

$$\begin{bmatrix} \sum_{i=1}^n w_i^2 I_{x_i}^2 & \sum_{i=1}^n w_i^2 I_{x_i} I_{y_i} \\ \sum_{i=1}^n w_i^2 I_{x_i} I_{y_i} & \sum_{i=1}^n w_i^2 I_{y_i}^2 \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix} = - \begin{bmatrix} \sum_{i=1}^n w_i^2 I_{x_i} I_{t_i} \\ \sum_{i=1}^n w_i^2 I_{y_i} I_{t_i} \end{bmatrix} \quad (2.9)$$

其中 $I_{x_i}$ ， $I_{y_i}$ ， $I_{t_i}$ 分别为相应像素点亮度在 X 轴、Y 轴方向上和时间上的梯度值。

令 $A = \begin{bmatrix} I_{x_1} & \dots & I_{x_n} \\ I_{y_1} & \dots & I_{y_n} \end{bmatrix}$ ， $W = \text{diag}(w_1, \dots, w_n)$ ， $b = [I_{t_1}, \dots, I_{t_n}]^T$ ，可将式(2.9)转化为式(2.10)：

$$A^T W^2 A \begin{bmatrix} u \\ v \end{bmatrix} = A^T W^2 b \quad (2.10)$$

由式(2.10)的解即为光流，如式(2.11)所示：

$$\begin{bmatrix} u \\ v \end{bmatrix} = (A^T W^2 A)^{-1} A^T W^2 b \quad (2.11)$$

## 2.5 背景减除法基本原理

背景减除法是一种有效的运动目标检测算法，其思路是使用背景的参数模型来近似背景图像的像素值，将当前帧与背景图像进行差分对比后，把差值较大的区域看作是运动目标，否则就看作是背景。背景减除法中对背景进行建模是必须步骤，而背景图像又必须随着环境变化而进行实时更新，所以背景减除法的核心就在于其背景建模与更新。

## 2.6 单高斯背景建模法

单高斯分布模型是一种图像处理背景提取的处理方法，单高斯分布模型较为简单，采用了参数迭代方式，不要求在每一帧图像都对背景进行建模，适用于背景单一不变的情况。

高斯分布即正态分布，是最常用的概率分布模型，在图像处理、模式识别、计算机视觉中经常被用来描述一些随机量的变化情况，如噪声、特征分布、像素灰度，且正态分布函数还经常被选取为加窗函数进行局部化处理，如平滑滤波、Gabor 变换等。这是因为一方面正态分布体现了自然界中普遍存在的有关变化量的某种统计规律，一方面还因为正态分布函数有着很好的数理特性，存在各阶连续的导数，在时域和频域具有相同的函数形式等等，便于分析。

单高斯背景建模法认为，对于一个背景模型，像素点像素值的分布满足高斯分布，对于任意点的像素值满足  $B(x, y) \sim N(\mu, \sigma^2)$ ，如式(2.12)所示：

$$I(x) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(x-\mu)^2}{2\sigma^2}} \quad (2.12)$$

在提取目标的过程中，首先计算一组图像序列中每个点的均值和方差，然后根据这两个属性对背景进行建模。当对每一张包含前景的图像进行运动目标检测时，首先需要测定该图像中每一个像素点的平均值和方差，接着进行阈值检测，从而判断该点是否属于前景，如式(2.13)所示：

$$\frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(IG(x,y)-IB(x,y))^2}{2\sigma^2}} > T \quad (2.13)$$

式(2.13)中， $T$  是一个人为设定的阈值，该值越低，运动目标检测越灵敏，但受噪声影响也越大，反之同理。

为了跟随环境的变化对背景建模进行更新，需要不断调整背景中每个像素点的参数，如式(2.14)和式(2.15)所示：

$$\mu(x, y, t) = lr * \mu(x, y, t-1) + (1-lr) * I(x, y, t) \quad (2.14)$$

$$\sigma(x, y, t) = lr * \sigma(x, y, t-1) + (1-lr) * (I(x, y, t) - \mu(x, y, t))^2 \quad (2.15)$$

式(2.14)和式(2.15)中， $lr$  是一个设定的参数， $lr$  值越小，背景更新越快，但受背景模型运动目标影响越大，反之同理。

由于单高斯背景建模法较为依赖图像初始序列的质量，因此该方法在实际应用中效果较差，需要进一步改善。

## 2.7 本章小结

本章重点阐述了本文将会使用的软件平台和光流法、背景减除法的基本原理。本文使用 OpenCv 机器视觉开源库在 VisualStudio2017 中使用 C++语言来实现算法。本章主要从光流、光流场等概念入手，系统介绍了光流法的假设、类型和基本原理，并着重介绍了 LK 光流法。本章介绍了背景减除法的基本原理，分析了背景建模对于该方法的意义，此外着重介绍了单高斯背景建模方法以及该方法的优劣性。

### 3 光流法的实现及改进

#### 3.1 图像序列读取

使用 VideoCapture 函数读取图像序列，将当前帧保存在路径下，由于光流法需要在当前帧中对前一帧中的特征点进行跟踪，因此在读入第二帧后开始对图像进行光流法处理，从第三帧开始，删除无需再进行光流法处理的帧。使用 cvLoadImage 函数从预设路径中读入相邻的两帧图像（当前帧与上一帧）。

#### 3.2 角点检测

如上文所述，在 LK 光流法中，特征点光流的二维分量如式(2.11)所示，如果希望该特征点光流有解，那么就需要  $(A^T A)$  可逆，此时  $(A^T A)$  满秩，即存在两个特征向量。在实际情况中，要想达到该要求，必须跟踪区域中具有至少两个方向的运动的纹理特征，在这种情况下，当跟踪区域在图像中的一个角点区域居中时， $(A^T A)$  将具有最佳属性。简而言之，角点对光流法来说具有很好的追踪特征，非常适合作为 LK 光流法中计算系数光流的特征点。

使用 cvGoodFeaturesToTrack 函数实现对角点检索，将当前帧的图像作为函数的输入，并调整角点的数量、质量、角点间最小距离等其他函数输入参数，该函数可以输出检索到的所有角点的坐标。

由于在本文中对于角点的处理涉及到几何测量值的提取和处理而并非简单的特征提取，因此需要更高的分辨率，将角点的坐标从整数坐标转化为实数坐标，要实现这一目标，需要使用亚像素角点检测技术，其基本原理如图 3-1 所示。

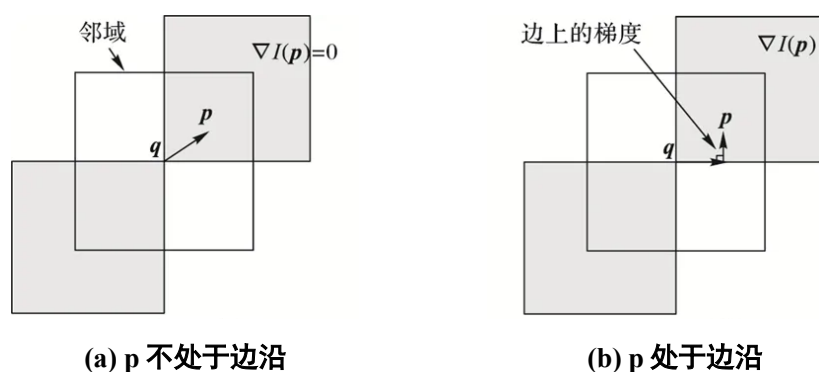


图 3-1 寻找角点到亚像素精度

假设起始角点位置  $q$  在实际亚像素角点位置附近，并检查从点  $q$  指向点  $p$  的向量。在图 3-2(a)中，从点  $q$  指向点  $p$  的向量。点  $p$  在附近的均匀区域，梯度为 0。而图 3-2(b)中，向量  $q - p$  与边界重合，那么处于边界上的  $p$  点的梯度与向量  $q - p$  正交。在任意情况下，点  $p$  处



的梯度和向量 $q - p$ 之间的点积为 0。根据上述推导，可以在邻点 $p$ 处梯度和对应向量 $q - p$ 上组合多个这样的梯度对，将其的点积设置为 0，同时求解上述组合的方程组，该方程组的解将为点 $q$ 的准确定位提供亚像素精度级别的坐标。

使用 `cvFindCornerSubPix` 函数来寻找角点的亚像素级别坐标，将当前帧图像以及 `cvGoodFeaturesToTrack` 得到的角点初始坐标作为输入，并设置搜索角点的窗口大小设为 21 像素宽，通过该函数，可以得到当前图像帧中的亚像素级角点坐标。

为了使角点检测达到较好的效果，需要对 `cvGoodFeaturesToTrack` 函数输入参数中的 `maxCorners`，`qualityLevel`，`minDistance` 参数进行调整。为了保证算法的实时性，将 `maxCorners` 固定为 100，接着调整 `qualityLevel`，`minDistance` 参数使检测到的角点数量接近一百，保证在 `maxCorners` 的限制下尽可能获得图像中所有的运动目标信息，具体调整过程如下：

首先固定 `minDistance` 参数值为 50，分别将 `qualityLevel` 设置为 0.1，0.2，0.3，0.4，角点检测效果如图 3-2 所示。



(a) `qualityLevel=0.1`



(b) `qualityLevel=0.2`



(c) `qualityLevel=0.3`



(d) `qualityLevel=0.4`

图 3-2 `qualityLevel` 不同取值时角点检测效果

观察图 3-2 可以发现，当 `qualityLevel` 取为 0.3 时，既可以检测出图像中多数需要追踪的角点，又不会检测出过多没有运动信息的角点。`qualityLevel` 取为 0.1 和 0.2 时，会

检测处图像中一些没有运动信息的角点，导致浪费计算量，影响算法的实时性。 $qualityLevel$  取为 0.4 时，角点检测效果与  $qualityLevel$  取为 0.3 的情况较相似，考虑到  $qualityLevel$  过高可能会导致对一些关键角点造成忽略，选择  $qualityLevel$  值为 0.3。

接着对  $minDistance$  参数值进行调整，将  $qualityLevel$  固定为 0.3，分别将  $minDistance$  设置为 20, 50, 80, 110，角点检测效果如图 3-3 所示。



(a)  $minDistance=20$



(b)  $minDistance=50$



(c)  $minDistance=80$



(d)  $minDistance=110$

图 3-3  $minDistance$  不同取值时角点检测效果

观察图 3-3 可以发现，当  $minDistance$  不断增大时，图像中检测得到的角点数不断减少，对于图像中存在的并排行走的路人的情况，往往只能检测到其中一个路人身上的角点，考虑到后续需要基于角点对光流进行计算，希望获得尽可能多的运动信息，因此将  $minDistance$  取为 20，获得尽可能多的角点以保证不会对图像中的运动信息有遗漏。

经过对比和分析，将  $maxCorners$  取为 100， $minDistance$  取为 20， $qualityLevel$  取为 0.3，作为  $cvGoodFeaturesToTrack$  函数的输入参数，接下来根据检测到的角点进行对光流的计算。

### 3.3 光流计算

对于大多数 30Hz 的摄像机，由于大幅度且非相干的运动较为普遍，所以需要一个大

窗口来捕捉大幅运动，但这与光流法目标不会在短时间内发生大幅位移的假设相冲突，为了克服该问题，需要使用图像金字塔。如图 3-5 所示，先对顶层中的光流进行计算，接着利用得到的光流运动估计作为下层的起始点，通过重复该过程到达最底层。通过这种方式，我们既可以追踪更快更长的运动，也可以最大限度地减少违反我们的运动假设。该方法称为“金字塔 Lucas-Kanade 光流法”。

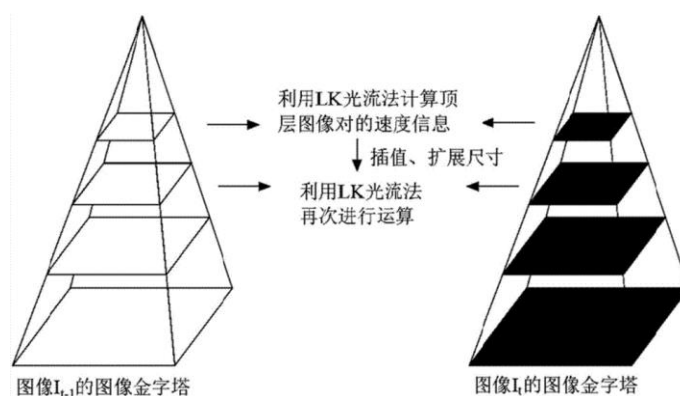


图 3-4 金字塔 Lucas-Kanade 光流法

使用 `cvCalcOpticalFlowPyrLK` 函数可以实现金字塔 Lucas-Kanade 光流法，该函数的前两个输入为上一帧和当前帧图像，然后还需要输入来自上一帧图像且需要在当前帧图像中进行跟踪的角点列表和跟踪成功的输出角点列表，此外还可以根据 `status` 和 `err` 数组来检测对应角点的跟踪是否成功。`status` 中的每个元素都会告知是否发现前一帧图像中的相应角点，当且仅当在当前帧中发现该角点是，该元素将为非零。`err` 中的元素将指示在当前帧中发现的任何角点的错误度量，若未发现该角点，则该元素则不定义。与 `cvFindCornerSubPix` 函数相同，可以对搜索窗口的大小进行设置，此外还可以通过 `maxlevel` 设置金字塔的 `cengshu`，若 `maxlevel` 为 0，那么不使用金字塔 LK 光流法。

计算出的光流如图 3-5 所示。



图 3-5 光流计算结果



### 3.4 基于光流法的三维场景信息计算

观察图 3-5 可以发现，光流法可以实现对视频中的运动目标进行跟踪和检测，但是也会对图像中一些静止的物体上的角点误识别为运动角点，无法获取准确的三维场景中的运动信息，因此需要对获取的光流信息进行进一步的筛选。本文主要从光流延伸焦点 FOE（Focus of Expansion）<sup>[27]</sup>和光流向量幅值两个特征来通过光流对三维场景信息进行计算。

#### 3.4.1 光流延伸焦点 FOE

FOE 来自于运动目标与摄像机本身的相对运动，如果将同一相对运动产生的光流向量无限延长，那么理论上这些光流会汇聚于二维图像中的一点，该点即为光流延伸焦点 FOE，所以计算得到的光流将以 FOE 为中心向其他方向发散，且越接近 FOE 的光流幅值越小，FOE 处的光流幅值为 0。

根据这一理论，将两个来自同一相对运动的光流无限延长，计算两延长光流向量的交点，即可得到该相对运动的 FOE。由于每一个运动目标与摄像机的相对运动都是独特的，因此每一个运动目标上的光流都有一个独特的 FOE，换句话说，FOE 可以被视作图像中运动目标的一个特征，FOE 与运动目标之间存在一一对应的关系。因此，通过对 FOE 的计算，就可以检测到相应的运动目标。

利用摄像机坐标系，将摄像机光心的位置当作三维场景坐标系  $OXYZ$  中的原点  $O$ ，二维平面的  $x$  轴和  $y$  轴作为三维坐标系的  $X$  轴和  $Y$  轴，摄像机的光轴作为三维坐标系的  $Z$  轴，如图 3-6 所示。

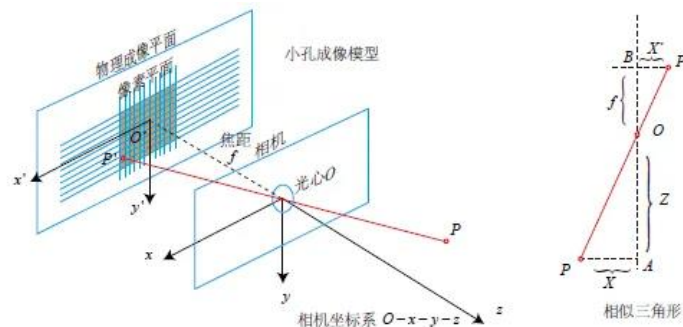


图 3-6 摄像机坐标系

将摄像机与运动目标的相对运动  $M$  分解为平移运动  $T = (T_x, T_y, T_z)^T$  和旋转运动  $R = (R_x, R_y, R_z)^T$ ，映射到三维场景坐标系  $OXYZ$  中，其运动描述如式(3.1)所示：

$$M = \begin{cases} -T_x - R_y Z + R_z Y \\ -T_y - R_x Z + R_z X \\ -T_z - R_x Y + R_y X \end{cases} \quad (3.1)$$

对式(3.1)求导，可以得到运动方程如式(3.2)所示：

$$\begin{cases} v_x = \frac{dx}{dt} = \frac{T_z x - T_x f}{Z} - R_y f + R_z y + \frac{R_x x y}{f} - \frac{R_y x^2}{f} \\ v_y = \frac{dy}{dt} = \frac{T_z y - T_y f}{Z} - R_x f + R_z x + \frac{R_y x y}{f} - \frac{R_x x y}{f} \end{cases} \quad (3.2)$$

在摄像机移动的过程中，运动目标和摄像机相对运动中的平移运动和旋转运动形成的光流是可以分开的，因为 FOE 的位置仅与平移运动有关，与旋转运动无关，所以可以忽略式(3.2)中代表旋转运动的分量，如式(3.3)所示：

$$\begin{cases} v_x = \frac{dx}{dt} = \frac{T_z x - T_x f}{Z} \\ v_y = \frac{dy}{dt} = \frac{T_z y - T_y f}{Z} \end{cases} \quad (3.3)$$

将式(3.4)右端分子分母同时除以 $T_z$ ，如式(3.4)所示：

$$\begin{cases} v_x = \frac{dx}{dt} = \frac{x - T_x f / T_z}{Z / T_z} \\ v_y = \frac{dy}{dt} = \frac{y - T_y f / T_z}{Z / T_z} \end{cases} \quad (3.4)$$

因为忽略旋转运动，所以根据投影变换公式即可求得光流 FOE 坐标 $(c_x, c_y)$ 如式(3.5)所示：

$$\begin{aligned} c_x &= \frac{T_x f}{T_z} \\ c_y &= \frac{T_y f}{T_z} \end{aligned} \quad (3.5)$$

联立式(3.4)和式(3.5)，可得式(3.6)：

$$v_x c_x - v_y c_y = v_x x - v_y y \quad (3.6)$$

式(3.6)中， $v_x$ ， $v_y$ 可以分别用光流向量 X 轴和 Y 轴方向的分量表示，要想求出光流 FOE 坐标 $(c_x, c_y)$ ，只需要计算处来自同一相对运动的两个光流向量，然后将两个方程联立即可。

### 3.4.2 Kmeans 算法计算 FOE

由于在运动目标检测场景中往往存在多个运动对象，无法直接使用所有光流来计算 FOE。为解决该问题，本文采用 Kmeans 算法对产生光流的角点进行聚类分析，接着再计算同一类光流的 FOE，具体步骤如图 3-7 所示：

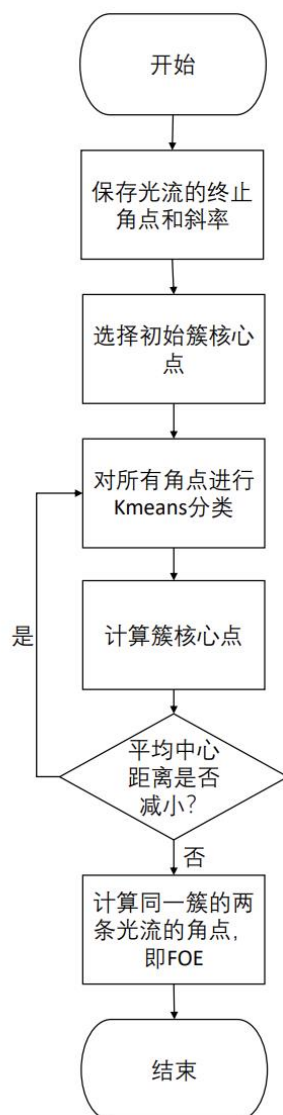


图 3-7 Kmeans 算法计算 FOE 流程图

- 1.每计算得出一条光流，将产生该光流的角点坐标和该光流的斜率保存；
- 2.当所有的光流计算完成，选择第 33，66，99 个角点作为初始聚集的簇核心点；
- 3.分别计算每一个角点到簇核心点的欧氏距离，找到距离该角点最近的簇核心点，将这个角点划入到相应的簇中，欧氏距离计算公式如式(3.7)所示：

$$d(x, y) = \sqrt{(x_1 - y_1)^2 + (x_2 - y_2)^2 + \cdots + (x_n - y_n)^2} = \sqrt{\sum_{i=1}^n (x_i - y_i)^2} \quad (3.7)$$

- 4.所有角点分类完成后，将每个簇的平均距离中心点作为每个簇新的簇核心点；
- 5.比较重新分类后的簇的平均中心距离值，当该值减小时，认为没有完成分类，继续迭代步骤 3、4，当该值不再减小时，分类完成，终止迭代；
- 6.随机取每一簇中的两个角点（对应两条光流），根据角点坐标和角点对应的光流斜率来计算两条光流延长线的交点，及 FOE；

使用 Kmeans 算法计算 FOE 的效果如图 3-8 所示。



图 3-8 Kmeans 算法计算 FOE 效果

在实际应用过程中，发现 Kmeans 算法对 FOE 的计算结果并不理想，尽管得到的 FOE 大部分情况能处于运动目标上，但是计算结果不稳定，很难给出一个准确的坐标值。经过分析认为原因如下：

1. 动态背景中存在很多运动角点，但只有少部分位于运动目标上，导致噪声严重影响 FOE 的计算，FOE 计算坐标不稳定，如图 3-9 所示；



图 3-9 Kmeans 计算 FOE 坐标不稳定

2. Kmeans 算法需要指定固定的簇数，但是三维运动场景中的运动目标数量往往不固定，当簇数与二维平面中出现的运动目标数量不匹配时，FOE 不准确，如图 3-11 所示。



图 3-10 Kmeans 簇数与运动目标数量不匹配

### 3.4.3 光流向量幅值检测运动目标

通过观察光流计算结果发现，运动目标上角点产生的光流向量的幅值明显大于其他

噪声角点，因此通过光流向量幅值对光流向量进行筛选，从而实现对运动目标的检测。具体步骤如图 3-11 所示。

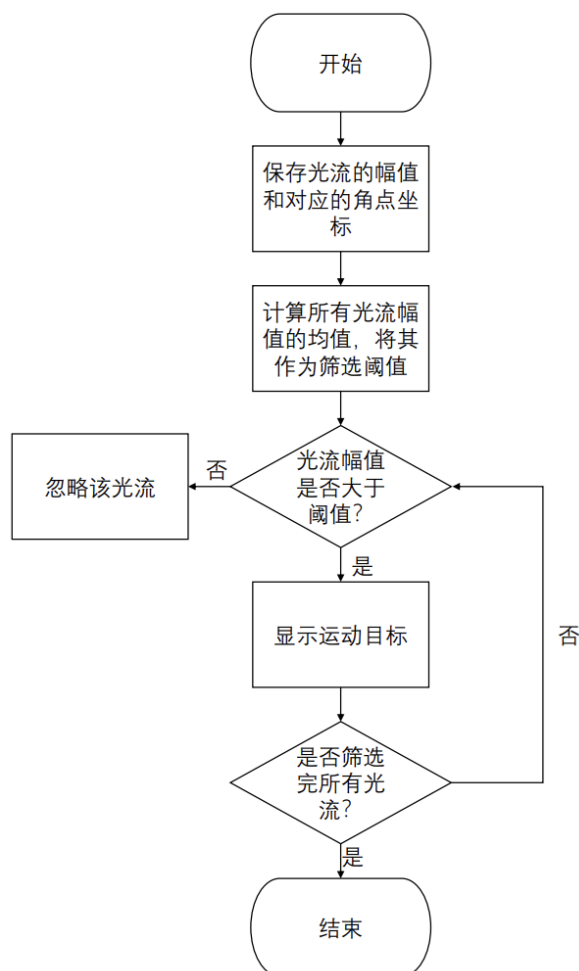


图 3-11 通过光流幅值计算运动目标流程图

- 1.每计算得出一条光流，将产生该光流的角点坐标和该光流的幅值保存；
- 2.当所有的光流计算完成，计算所有光流幅值的平均值，将其作为筛选光流幅值的阈值；
- 3.对每一条光流进行筛选，若大于阈值，将该光流和对应角点显示在平面上，若小于阈值则忽略该光流。

使用光流向量幅值来检测运动目标的效果如图 3-12 所示：





(a)使用光流幅值检测运动目标



(b)原始光流计算结果

图 3-12 使用光流幅值检测运动目标效果

图 3-12(a)为使用光流幅值筛选光流从而检测运动目标的结果，对比图 3-12(b)中的原始光流计算结果可以看到，这种方法可以有效去除非运动目标上角点产生的光流，实现对运动目标的检测。

### 3.5 本章小结

本章使用光流法对背景中的运动目标进行检测，将图像中的角点作为 LK 光流法使用的特征点，并将角点的坐标精确到亚像素精度以保证光流计算的精度。使用金字塔 LK 光流法，在不违反运动假设的前提下，实现对快速运动的追踪。为了实现对运动目标的准确检测，基于光流信息进一步对三维场景中的运动信息通过两种方式进行计算。使用 Kmeans 对光流进行聚类之后计算 FOE 的方法容易受到动态背景中噪声的影响，且当固定的簇数与二维平面中出现的运动目标个数不匹配时效果较差。使用光流幅值对光流进行筛选之后从而实现对运动目标检测的方法效果较好，可以将运动目标上的角点和光流从原始光流中提取出来。

## 4 静态背景运动目标检测

### 4.1 单高斯模型背景建模

一个一维高斯模型由一个平均值和一个方差构成，使用一个一维高斯模型来表示一个像素点的变化就必须对该点像素值的平均值和方差进行计算。

#### 4.1.1 使用 accumulateWeighted 函数计算均值

使用 running average(移动平均)方法计算像素值均值，该方法可以消除像素值随机波动的影响，计算方法如式(4.1)所示：

$$acc(x, y) = (1 - \alpha) \cdot acc(x, y) + \alpha \cdot image(x, y) \quad (4.1)$$

相较于简单的求均值方法，移动平均法中最近出现的值对结果的贡献之前的值更大，以 $\alpha = 0.5$ ，样本值为2，3，4的情况进行说明。

对于简单的求均值方法，累加的方式为直接求和，因此均值为：

$$avg = \frac{(2 + 3 + 4)}{3} = 3$$

对于移动平均法，第一次计算会得到：

$$(1 - 0.5) \times 2 + 0.5 \times 3 = 2.5$$

第二次计算会得到：

$$(1 - 0.5) \times 2.5 + 0.5 \times 4 = 3.25$$

对比可知，移动平均法计算得到的均值相较于简单的平均值更大，说明结果受到最近出现的值4的影响更大，这对背景减除法中模型的有利。

为实现在整个图像序列中进行累加，使用 OpenCv 提供的函数 accumulateWeighted，该函数的输入参数为数组 dst，图像 src，alpha 值，图像 accumulate。其中数组 dst 为累加产生的数组，数组 src 为带累加的新图像，alpha 值是权重参数，参数 alpha 值控制先前帧产生的影响淡去的速率，该值越小，先前帧产生的影响淡去的速率越快。图像 accumulate 是掩码图像，在一个几乎没有前景物体的场景中计算背景模型时，掩码图像十分有用，原因在于可以通过一些其他信息确认该部分不属于背景。

#### 4.1.2 使用 accumulateSquare 函数计算方差

对于一个有限总体，其方差定义如式(4.2)：

$$\sigma^2 = \frac{1}{N} \sum_{i=0}^{N-1} (x_i - \bar{x})^2 \quad (4.2)$$

如果根据式(4.2)进行计算，那么需要遍历两次数据才能计算出结果，因此将式(4.2)转换为式(4.3)：

$$\sigma^2 = \left( \frac{1}{N} \sum_{i=0}^{N-1} x_i^2 \right) - \left( \frac{1}{N} \sum_{i=0}^{N-1} x_i \right)^2 \quad (4.3)$$

通过式(4.3)计算方差，可以在一次遍历中同时累加像素的值和它们的平方，再通过像素平方的均值减去像素的均值来得到其方差。为了避免数据溢出，使用 OpenCv 提供的函数 `accumulateSquare` 来计算方差。该函数的输入参数与 `accumulateWeighted` 类似，唯一的区别在于无需给定 `alpha` 值。

单高斯模型背景建模效果如图 4-1 所示：



图 4-1 单高斯模型背景建模

## 4.2 背景差分

实现了对背景图像的建模后，通过检测背景与当前图像的差分图像的像素值即可实现对运动目标的提取。

为了优化运动目标提取的效果，对于每一帧待检图像均使用 `cvSmooth` 函数进行高斯滤波，以去除图像中的噪声，使图像尽可能平滑。

接下来借助 OpenCv 提供的 `cvAbsDiff` 函数计算当前帧与背景模型的差分图像，再使用 `cvThreshold` 函数来对差分图像的灰度值进行阈值检测，该函数的 `threshold` 参数即为阈值，通过调整该参数来达到较理想的运动目标检测效果，具体调参过程如下：

设置 `threshold` 参数值为 5, 10, 20, 40，运动目标检测效果如图 4-2 所示：

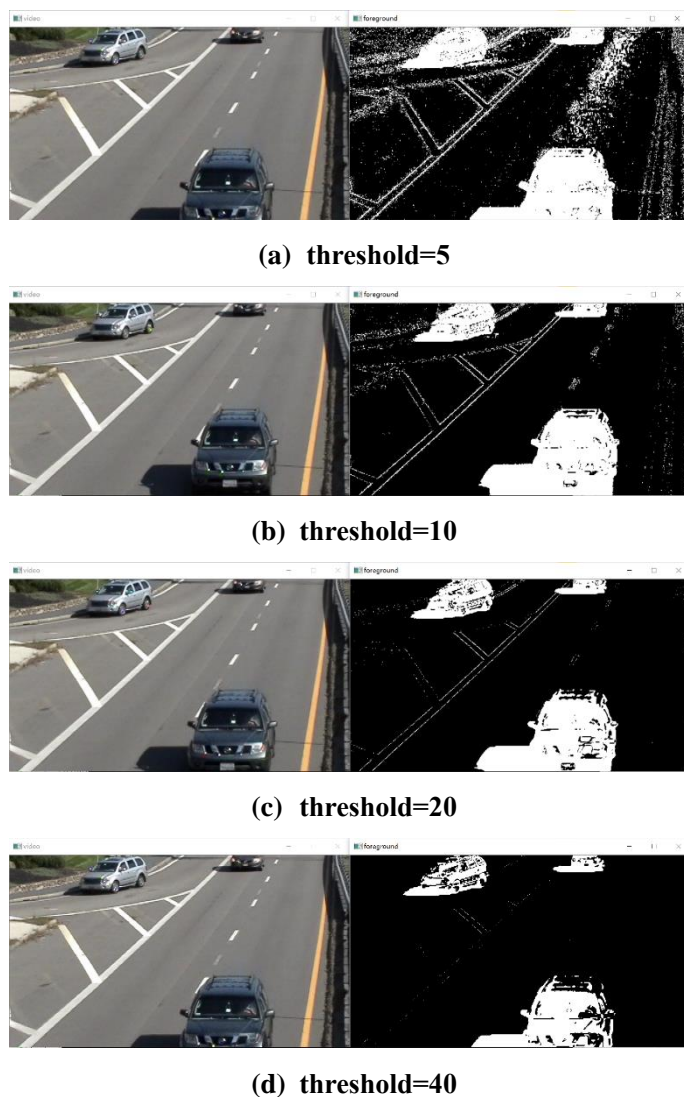


图 4-2 threshold 取不同值时运动目标提取效果

如图 4-2 所示，当 threshold 值取为 5，10 时，虽然能够提取到图像中的运动目标，但是由于 threshold 过低，检测过于灵敏，会将图像中的噪声和静止物体一起提取出来；当 threshold 值取为 40 时，虽然不会提取出其他目标，但是运动目标内部出现了空洞，说明阈值过高导致运动目标无法被完整地提取。综合比较来看，threshold 值取为 20 时，运动目标提取效果最好。

### 4.3 光流法-单高斯背景减除法

#### 4.3.1 单高斯背景模型建模局限性

单高斯模型对背景进行建模的方法往往对初始图像序列的要求较高，一旦运动目标在初始图像序列中出现，则获得的背景往往就包含了运动目标，这样导致在求背景与当前帧图像的差分图像时，初始图像序列中的运动目标会一直出现在前景中，如图 4-3 所示：

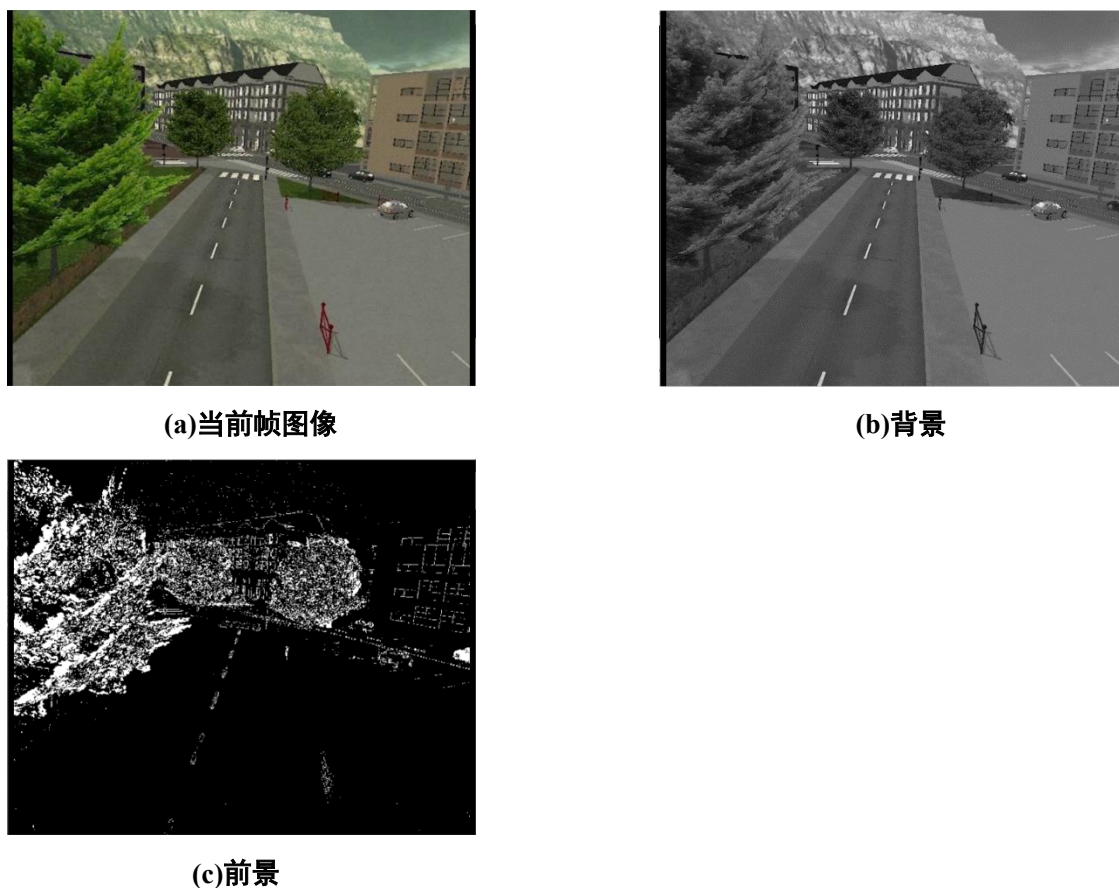


图 4-3 单高斯背景模型受初始图像序列影响

根据图 4-3 可以看出，单高斯背景模型受初始图像序列影响较严重，因此需要对该方法做进一步改进。

#### 4.3.2 光流法-单高斯背景减除法基本思路

单高斯背景模型背景减除法能够从当前图像提取运动目标的轮廓，但受初始图像序列影响，提取的轮廓可能包含了除运动目标以外的轮廓，因此需要对提取的轮廓进行进一步筛选。

在第三章动态背景检测中，已经通过光流法获得图像中运动目标的坐标。本文通过光流法获得的运动目标坐标来对轮廓进行筛选，保留运动目标的轮廓，从而提升单高斯模型背景减除法的效果，削弱不准确背景模型的影响，具体思路如图 4-4 所示。

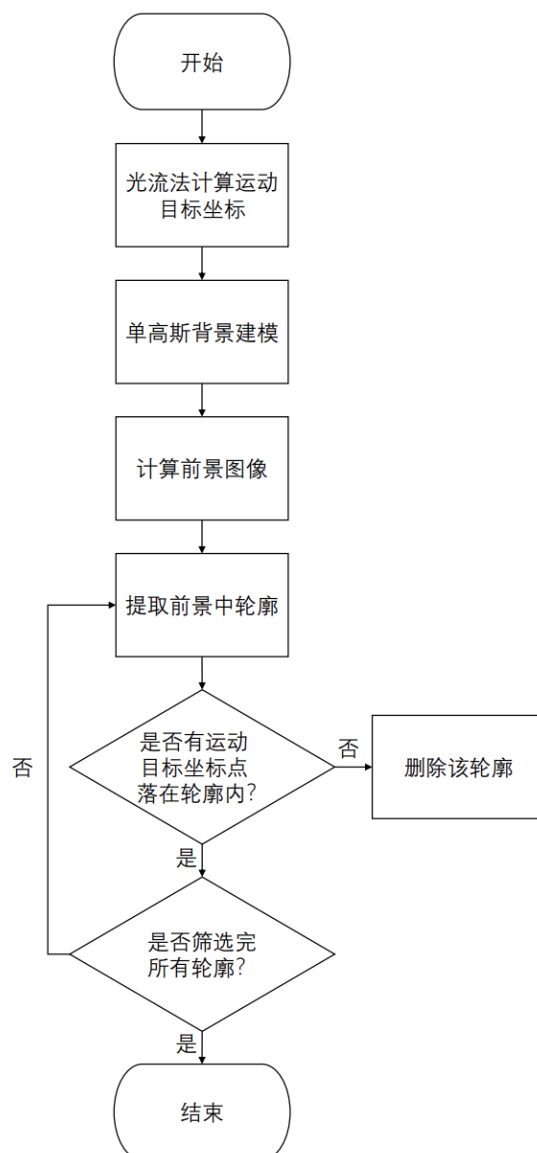


图 4-4 光流法-单高斯背景减除法流程图

1. 使用光流法检测图像中运动目标，保存运动目标的位置坐标；
2. 使用单高斯模型对图像序列的背景建模；
3. 求当前图像与背景的差分图像，通过阈值检测差分图像，获得前景图像；
4. 提取前景中所有的轮廓，计算轮廓的中心与所有运动目标坐标的距离；
5. 判断运动目标坐标点是否落在轮廓内，若是则保留，若否则删除该轮廓。

#### 4.3.3 轮廓提取

在图像中，每一条轮廓都对应一组点，这些点以一定的形式表现为图像中的一条曲线，在 OpenCv 中，使用向量来表示构成轮廓的一组点。

使用 `findContours` 函数来检测前景图像中的轮廓，该函数的主要输入参数为输入图像，



该图像必须为 8 位单通道二值图像，而上文得到的前景图像满足该条件，可以作为输入图像，考虑到 `findContours` 函数在测量轮廓时直接对图像进行了涂改操作，因此将前景图像复制后再输入 `findContours` 函数。第二个输入参数为 `mode`，该参数指定了所有轮廓的检测方式，将 `mode` 设为不同值，`findContours` 函数会选择检索最外层轮廓、检索所有轮廓、检索所有轮廓并组织成双层结构、检索所有轮廓并建立网状结构这四种不同的方式。由于本文只对轮廓的位置感兴趣，故选择只检索最外层轮廓。

#### 4.3.4 轮廓筛选

在判断运动目标坐标点与轮廓的位置关系之前，需要对前景中的轮廓进行几何学上的处理以将其简化，以保证筛选过程的准确率。

首先通过多边形来逼近一个轮廓，减少轮廓的顶点数，选择使用 Douglas-Peucker 算法来实现该过程。先在组成轮廓的点中选出两个相距最远的点，接着将两点相连成一条线段，之后在轮廓点中选出距离该线段最远的点，不断重复该过程，直到点与线段的距离达到一个指定的参数。使用 OpenCv 中的 `approxPolyDP` 函数来实现该算法。

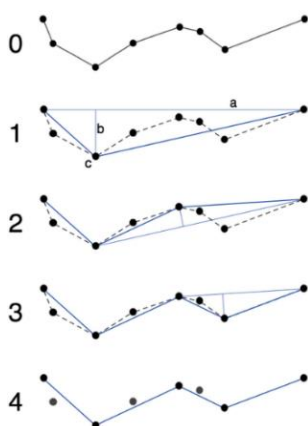


图 4-5 Douglas-Peucker 算法

通过上述过程逼近得到的多边形仍然不够简化，需要通过寻找多边形的凸包来对其进行简化。一个多边形的凸包，指的是一个完全包含原多边形且由原多边形上的点组成的多边形，该多边形的每一处都必须是凸的（多边形上的任意连续三个点内角相遇 $180^\circ$ ）。使用 Graham 扫描算法来得到多边形凸包，首先根据极角坐标对原多边形上的点集进行排序，以对原本较杂乱的点积进行梳理，接着按照顺序将每一点加入表示凸包的栈中，当新加入的点形成的边与上一条边成左转关系时则继续加入新的点，成右转关系则将右转点弹出栈，直到新边与上一条边成左转关系，通过这种方式保证栈内的点始终组成一个凸包，当所有的点扫描完成后，即可得到原多边形的凸包。使用 OpenCv 中的 `convexHull` 函数来实现该算法。

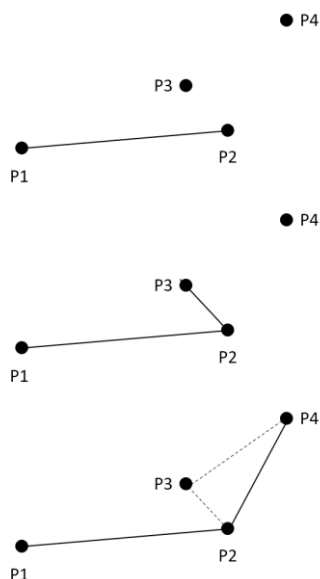


图 4-6 Graham 扫描算法

将前景中的轮廓简化为凸包后，通过判断光流法得到的运动目标坐标点与凸包的位置关系即可实现轮廓的筛选。考虑到光流法输出的点较多，为了减少计算量，修改 `cvGoodFeaturesToTrack` 函数的参数 `minDistance` 取为 70，尽量使每个运动目标上只有一个角点被检测跟踪，效果如图 4-5 所示。



图 4-7 减少跟踪角点数的光流法

使用 OpenCv 中的 `pointPolygonTest` 函数来计算光流法得到的运动目标坐标点与凸包的位置关系。将该函数的 `measureDist` 参数设置为 `false`，当函数返回值为 -1 时，表示点在凸包外部，为 0 时表示点在凸包上，为 1 时表示点在凸包内部。考虑到光流法跟踪的角点一般位于运动目标的边缘位置，为了防止误判，当函数返回值为 0 或 1 时，将该凸包对应的原轮廓保留在前景中，当函数返回值为 -1 时，将原轮廓从前景中删除。

光流法-单高斯背景减除法得到的前景如图 4-6 所示。



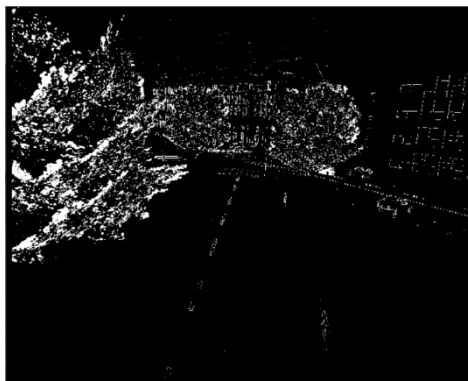


图 4-8 光流法-单高斯背景减除法的前景

对比图 4-3(c)中单高斯背景减除法得到的前景可以看出，光流法-单高斯背景减除法可以削弱图像初始序列中出现的运动目标的影响，将假运动目标的轮廓从前景中剔除。

#### 4.4 本章小结

本章首先将光流法与单高斯背景减除法进行结合以实现运动目标检测。先采用单高斯模型对图像序列中的背景进行建模，然后计算背景图像与当前帧图像的差分图像，再通过对检测差分图像中像素点的灰度值进行阈值检测以实现对运动目标的轮廓的提取。经过分析发现，上述方法较依赖初始图像序列，当运动目标频繁出现在初始图像序列中时，背景建模不准确，导致前景中出现错误的轮廓。针对这一缺点，本文将光流法与单高斯背景减除法进行结合以实现改进，首先使用光流法计算出运动目标在图像中的坐标，接着使用单高斯背景减除法提取出轮廓，在将轮廓简化为凸包过后，通过判断运动目标坐标点与轮廓的位置关系来对轮廓进行筛选，从而剔除单高斯背景减除法可能提取出的错误轮廓。结果表明，该方法可以削弱初始图像序列和噪声对算法的影响。

## 5 算法测试

本文选择使用 SBMnet 数据集<sup>[28]</sup>的 Basic 组共 17 个图像序列进行测试, 该组图像序列涵盖了多种运动目标检测的难点, 具有目前在监控、智能环境和视频数据库场景中捕获的典型室内和室外视觉数据的代表性。数据集测试结果如图 5-1 所示。

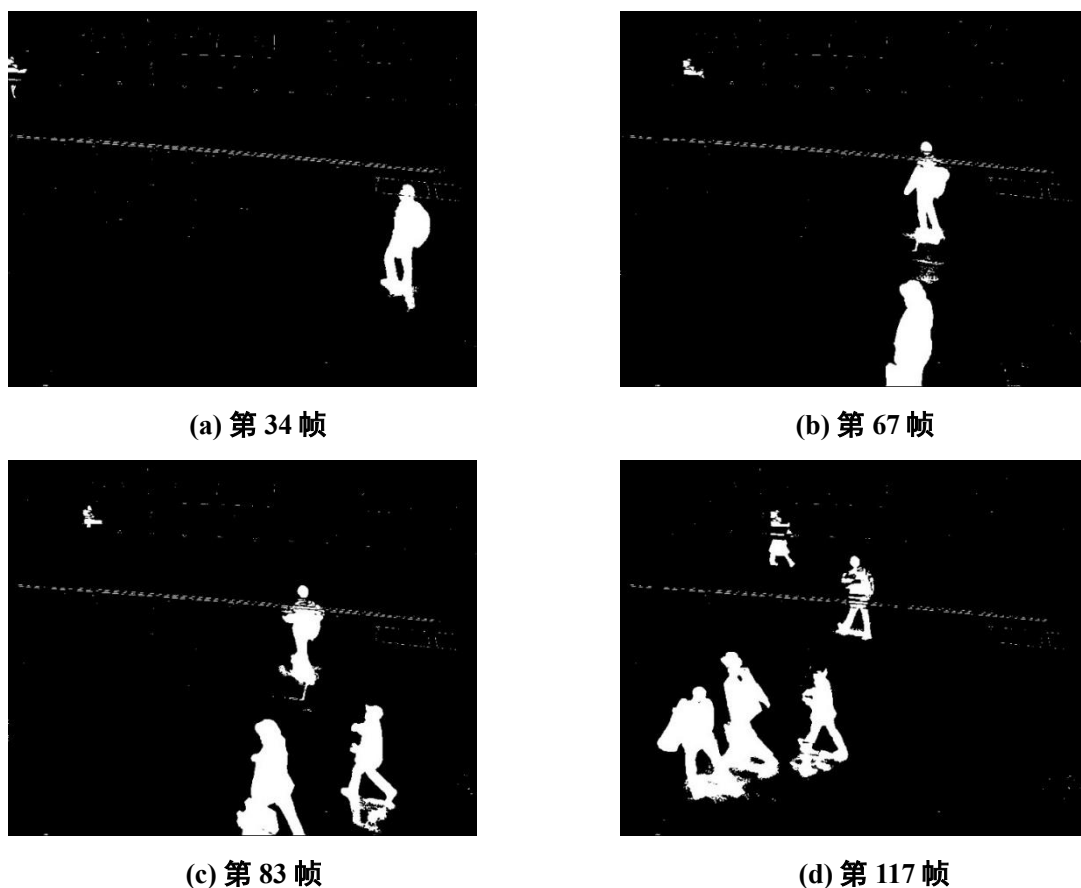


图 5-1 SBMnet 数据集测试结果

观察图 5-1 可以发现, 本文提出的改进算法可以完整的提取出图像中运动目标的轮廓, 并剔除前景中由于初始图象序列影响产生的错误轮廓。但是通过图 5-1 中依然可以观察到算法具有以下局限性:

在图 5-1 中可以观察到一条白线, 该轮廓来自于图像中的噪声, 这说明算法无法完全剔除前景中的噪声, 经过分析认为原因在于噪声在前景中表现为孤立的点, 无法被视作轮廓进行筛选, 因此无法被剔除。

此外, 在图 5-1(c)中可以观察到部分运动目标的轮廓无法被完整地提取, 经过分析认为造成该现象的原因是该运动目标的灰度值与背景灰度值相近, 导致差分图像中该区域的灰度值过小, 无法通过阈值检测提取完整轮廓。

在图 5-1(d)可以观察到运动目标的影子也会被视作运动目标的一部分, 这在部分应用

场景中可能会造成一些影响。

同时，本文也使用 SBMnet 数据集提出的六类指标来对算法的性能进行定量描述，指标具体如下：

AGE（平均灰度误差）：图像之间灰度差值的平均值；

pEPs（误差像素占比）：图像中误差像素（与 Ground Truth 中对应像素灰度值差超过一个阈值）占有所有像素的比例；

pCEPS（聚集误差像素占比）：图像中聚集误差像素（像素本身和其相邻四个像素都为误差像素）占有所有像素的比例；

MSSSIM（多尺度结构相似指数）：对感知到的视觉扭曲的估计；

PSNR（信噪比峰值）：计算公式如式(5.1)所示，其中 $L$ 为最大灰度值， $MSE$ 为图像与 Ground Truth 的均方误差；

$$PSNR = 10 \log_{10}^{(L-1)^2 / MSE} \quad (5.1)$$

CQM（彩色图像质量）：基于 YUV 颜色空间的可逆变换和在单个 YUV 波段计算的 PSNR，该值越高，检测效果越好。

本文提出的光流-单高斯背景减除法，并将测试结果与单高斯背景建模法和其他改进运动目标检测算法进行横向的对比和分析。

SBMnet 数据集的测试结果如表 5-1 所示。

表 5-1

算法 指标	FSBE <sup>[29]</sup>	RMR <sup>[30]</sup>	SG	Proposed
AGE	3.8960	5.8867	15.8439	5.5261
pEPs	0.0131	0.0480	0.1753	0.1427
pCEPS	0.0028	0.0167	0.1425	0.0949
MSSSIM	0.9784	0.9400	0.9075	0.9425
PSNR	32.7916	29.6069	26.4462	27.2659
CQM	33.4660	30.4044	27.3629	30.9821

根据表 5-1 数据集可以看出，FSBE 算法各方面指标都达到了最优，RMR 算法综合性能最劣。本文提出的光流-单高斯背景减除法相较于单高斯背景减除法在各项指标都有了提升，AGE、pEPs、MSSSIM 和 CQM 指标仅次于 FSBE 算法，但 pCEPS 指标，即图像中的聚集错误像素占大于 RMR 算法，而错误像素占比小于 RMR 算法，这说明算法得到的图像中错误像素多处于一个集中的区域。分析后认为这是由于前景中运动目标的轮廓中存在空洞造成。此外，本文算法的 PSNR 指标，即信噪比峰值指标也大于 RMR 算法，

分析后认为原因在于对前景中的错误轮廓剔除效果不够稳定，导致某一时刻的前景中依然会出现错误的轮廓，噪声影响较严重。

## 6 总结与展望

### 6.1 总结

本文对运动目标检测问题进行了研究，重点研究内容如下：

对运动目标检测算法的研究意义进行介绍，并总结了目前运动目标检测算法的国内外研究现状，总结了目前运动目标检测算法的局限性。

1. 介绍了本文用于改进的光流法和单高斯背景减除法的基本原理，并分析算法在实际应用过程中出现的问题。

2. 实现基本的金字塔 LK 光流法，以光流向量为基础，进一步对三维空间中的运动信息进行计算，分别通过光流延伸焦点和光流向量幅值两类特征对光流进行处理以获得运动目标的坐标，实验证明，通过光流向量幅值计算得到的运动目标坐标更为准确和稳定。

3. 实现基本的单高斯背景减除法，针对该算法容易受初始图象序列质量和噪声影响的缺点进行了改进，将上述的光流法与单高斯背景减除法结合，通过运动目标坐标对单高斯背景减除法得到的轮廓进行筛选，实验证明，该算法对于初始图象序列的依赖性减弱，且可以更好地抵抗噪声。

4. 通过公开数据集对提出的算法进行了试验,试验结果表明本文中给出的算法相较于单高斯背景减除法在性能上有较大的提升，但是轮廓中依然有空洞的出现，且对噪声的抑制性能要弱于其他改进算法。

### 6.2 展望

本文提出的改进算法虽然相较于原算法有了性能上的提升，但是本文的研究工作依然存在以下不足：

本文的改进算法在进行轮廓筛选的过程中需要对两个数组进行遍历，当图像序列中存在较多的运动目标时，算法的实时性较差，且筛选过程对噪声的抑制性较差。需要对轮廓筛选过程进行进一步优化，提高其实时性和准确性

本文的改进算法在动态背景场景下应用效果较差，需要提升背景建模算法的鲁棒性以适应各个场景下的检测挑战；

本文的改进算法在运动目标灰度值与背景灰度值相近地场景中无法完整地提取轮廓，需要对阈值检测过程进行进一步优化，保证轮廓的完整性。

本文的改进算法无法将运动目标本身和其影子分离，这在部分应用场景中可能会有影响，需要对图像中的运动目标及其影子进行识别从而达到分离的目的。

## 参考文献

- [1] 赵巧敏. 人工智能行业投资分析[J]. 机器人技术与应用, 2016, 5(05): 40-48.
- [2] Chang, Ming-Shaung, Chou, Jung-Hua. A novel machine vision-based mobile robot navigation system in an unknown environment [J]. International Journal of Robotics & Automation, 2010, 25(04): 206-3372.
- [3] Wu Tengfei, Yang Zhiguang. Animal tumor medical image analysis based on image processing techniques and embedded system[J]. Microprocessors and Microsystems, 2021, 81:103671.
- [4] Collins R.T., Lipton A.J., Kanade T, et al. A system for video surveillance and monitoring [J]. VSAM Final Report, 2000, pp.1-68.
- [5] Sengar S.S., Mukhopadhyay S. Detection of moving objects based on enhancement of optical flow[J]. Optik International Journal for Light & Electron Optics, 2017, 145:130-141.
- [6] P. Kaewtrakulpong, R. Bowden. An improved adaptive background mixture model for real-time tracking with shadow detection[C]. Video-based Surveillance Systems, 2002, pp.135-144.
- [7] Lucas D, Kanade T. An Iterative Image Registration Technique with an Application to Stereo Vision[C]. Proceeding of Imaging Understanding Workshop, 1981:121-130.
- [8] Horn, Schunck B. Determining Optical Flow[J]. Artificial Intelligence, 1981(17):185-203.
- [9] Black M J, Anandan P. A Framework for the Robust Estimation of Optical Flow[C]. International Conference on Computer Vision. IEEE, 1993:231-236.
- [10] Serdar I, Janusz K. Occlusion-aware Optical Flow Estimation[J]. IEEE Transactions on Image Processing: A Publication of the IEEE Signal Processing Society, 2008, 17(8): 1443-1451.
- [11] Zach C, Pock T, Bischof H. A Duality Based Approach for Realtime TV-L1 Optical Flow[C]. Joint pattern recognition symposium. 2007, 28(10):1232-1239.
- [12] Simon D, Vinod C, and Sridha S. An Adaptive Optical Flow Technique for Person Tracking Systems[J]. Pattern Recognition Letters, 2007, 28(10):1232-1239.
- [13] Gloyer Brian, Aghajan Hamid K, Siu KaiYeung, Kailath Thomas. Video-based freeway monitoring system using recursive vehicle tracking[J]. Univ. of California/Irvine (United States); Stanford Univ. (United States), 1995, 2421.
- [14] Stauffer C, Grimson W E L. Adaptive Background Mixture Models for Real-time Tracking [C]. Proceedings. 1999 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (Cat No PR00149). IEEE, 1999, 2:246-252.
- [15] Zickovic Z. Improved Adaptive Gaussian Mixture Model for Background Subtraction[C]. Proceeding of the 17<sup>th</sup> International Conference on Pattern Recognition, 2004. ICPR 2004. IEEE, 2004, 2: 28-31.

- [16] 王典. 基于混合高斯模型的背景建模与阴影一直算法研究[D]. 西安: 西北工业大学, 2006.
- [17] Elgammal A, Duraiswami R, Harwood D, et al. Background and Foreground Modeling Using Nonparametric Kernel Density Estimation for Visual Surveillance[J]. *Proceeding of the IEEE*, 2002, 90(7):1151-1163.
- [18] Barnich O, Van Droogenbroeck M. ViBe: A Universal Background Subtraction Algorithm for ViBe[C]. 2012 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops. IEEE, 2012: 32-37.
- [19] Liu S. An Improved ViBe Moving Object Detection Algorithm Based on Spatial-temporal Gradient of Image[C]. 2016 International Conference on Progress in Informatics and Computing(PIC). IEEE, 2016:192-197.
- [20] 张钦礼, 邱杰, 杨秀兰. 改进的 ViBe 算法及其在运动目标检测中的应用[J]. *南京师大学报(自然科学版)*, 2020, 43(04): 104-112.
- [21] Yang J, Wang J, Lu H. A Hierarchical Approach for Background Modeling and Moving Objects Detection[J]. *International Journal of Control, Automation and Systems*, 2010, 8(5):940-947.
- [22] Wu H H P, Chang J H, Weng P K, et al. Improved Moving Object Segmentation by Multiresolution and Variable Thresholding[J]. *Optical Engineering*, 2006,45(11):117003.
- [23] Braham M, Van Droogenbroeck M. Deep Background Subtraction with Scene-specific Convolutional Neural Networks[C]. 2016 International Conference on Systems, Signals and Image Processing(IWSSIP). IEEE, 2016:1-4.
- [24] E. D. Gelasca and T. Ebrahimi, "On Evaluating Video Object Segmentation Quality: A Perceptually Driven Objective Metric," in *IEEE Journal of Selected Topics in Signal Processing*, vol. 3, no. 2, pp. 319-335, April 2009, doi: 10.1109/JSTSP.2009.2015067.
- [25] P. Correia and F. Pereira, Objective evaluation of video segmentation quality. *IEEE Transactions on Image Processing*, vol. 12, pp. 186-200, 2003.
- [26] R. Mech and F. Marques, Objective evaluation criteria for 2d-shape estimation results of moving objects, in *WIAMIS*, Tampere, Finland, 16-17 May 2001.
- [27] 金钊. 基于光流算法的无人机动态航迹规划技术研究[D]. 电子科技大学, 2018.
- [28] Jodoin P-M, Maddaelena L., Petrosino A., Wang Y. Extensive Benchmark and Survey of Background Modeling Methods, *IEEE Transactions on Image Processing*, 26(11), 2017, p.5244-5256.
- [29] Achraf Djerida Robust background generation based on an effective frames selection method and an efficient background estimation procedure (FSBE) *Journal Signal Processing: Image Communication* DOI 10.1016/j.image. 2019.06.001.

[30] D. Ortego, J. C. SanMiguel, J. M. Martínez, "Rejection based multipath reconstruction for background estimation in video sequences with stationary objects", *Computer Vision and Image Understanding*, vol. 147, pp. 23-37, 2016.



## 声 明

本人声明所呈交的学位论文是本人在导师指导下进行的研究工作及取得的研究成果。据我所知，除了文中特别加以标注和致谢的地方外，论文中不包含其他人已经发表或撰写过的研究成果，也不包含为获得四川大学或其他教育机构的学位或证书而使用过的材料。与我一同工作的同志对本研究所做的任何贡献均已在论文中作了明确的说明并表示谢意。

本学位论文成果是本人在四川大学读书期间在导师指导下取得的，论文成果归四川大学所有，特此声明。

学位论文作者（签名）\_\_\_\_\_

论文指导教师（签名）\_\_\_\_\_

2022 年 04 月 27 日

## 致 谢

首先我在这里向养育和培养我的父母表示深深地思念和感谢，再向关心我，帮助我的老师和同学表示感谢！

能顺利圆满地完成我的设计，离不开自身的努力，离不开指导老师涂海燕的指点，更离不开四川大学这个大环境四年来在学习和生活的熏陶和培养。

大四上学期开始不久，我开始了我的毕业设计。在这几个月里，我得到了导师涂海燕老师的关心和教导，涂海燕老师在学术上的渊博知识让我在这几个月的学习和研究中深受其益。我向涂海燕老师表示深深地谢意；再向和我进行讨论问题，让彼此相互帮助和学习工作的同学表示感谢；向电气工程学院实验室中，陪着我们度过几多光阴的管理员和老师表示谢意。

毕业设计的完成，为我在川大的学习生涯画上了一个圆满的句号。在四川大学度过的美好时光将使我终生受益。

## 附录 1

由于源代码代码较多，这里只摘录代码主体部分。

```
if (i == 1)//若为第一帧图像，对背景减除法的前景、背景进行初始化
{
    imgFR = cvCreateImage(cvSize(imgCR->width, imgCR->height), IPL_DEPTH_8U, 1);//
初始化前景图像
    imgBK = cvCreateImage(cvSize(imgCR->width, imgCR->height), IPL_DEPTH_8U, 1);;//
初始化背景图像
    matFR = cvCreateMat(imgCR->height, imgCR->width, CV_32FC1);
    matBK = cvCreateMat(imgCR->height, imgCR->width, CV_32FC1);
    matCR = cvCreateMat(imgCR->height, imgCR->width, CV_32FC1);
    //仅当mat为IplImage*类型，且其depth为IPL_DEPTH_8U（8bit无符号整形）时，有较好的
显示效果
    //对于其他深度的IplImage*或者CvMat*类数据，可以使用CvConvert函数进行转换
    cvCvtColor(imgCR, imgBK, CV_BGR2GRAY);
    cvCvtColor(imgCR, imgFR, CV_BGR2GRAY);
    cvConvert(imgFR, matCR);
    cvConvert(imgFR, matFR);
    cvConvert(imgFR, matBK);
}
else
{
    //背景减除法
    //颜色空间转换
    cvCvtColor(imgCR, imgFR, CV_BGR2GRAY);
    cvConvert(imgFR, matCR);
    //高斯滤波
    cvSmooth(matCR, matCR, CV_MEDIAN, 3, 0, 0);
    //当前帧与背景图做差分，求得运动部分，结果存入前景
    cvAbsDiff(matCR, matBK, matFR);
    //二值化前景，阈值设为30，阈值越小，检测越敏感
    cvThreshold(matFR, imgFR, 30, 255.0, CV_THRESH_BINARY);
    //将当前帧按照0.003的权重累加到背景中，从而更新背景
```

```
cvRunningAvg(matCR, matBK, 0.003, 0);
//将背景转化为图像格式，用以显示
cvConvert(matBK, imgBK);

//光流法
//读入当前帧与上一帧图像
sprintf(imgAName, "%s%d%s", "Y:\\GraduationProject\\images\\img", i - 1,
".jpg");
sprintf(imgBName, "%s%d%s", "Y:\\GraduationProject\\images\\img", i, ".jpg");
IplImage *imgA = cvLoadImage(imgAName, CV_LOAD_IMAGE_GRAYSCALE); //导入上一帧图
像
IplImage *imgB = cvLoadImage(imgBName, CV_LOAD_IMAGE_GRAYSCALE); //导入当前帧图
像

CvSize img_sz = cvGetSize(imgA);
IplImage *eig_image = cvCreateImage(img_sz, IPL_DEPTH_32F, 1);
IplImage *tmp_image = cvCreateImage(img_sz, IPL_DEPTH_32F, 1);

cvGoodFeaturesToTrack(
    imgA,
    eig_image,
    tmp_image,
    cornersA,
    &corner_count, //角点数
    0.3, //角点质量
    70.0, //角点间最小距离
    0,
    3,
    0,
    0.04
); //寻找角点

cvFindCornerSubPix(
```

```

        imgA,
        cornersA,
        corner_count,
        cvSize(win_size, win_size),
        cvSize(-1, -1),
        cvTermCriteria(CV_TERMCRIT_ITER | CV_TERMCRIT_EPS, 20, 0.03)
    );//精确角点位置

    CvSize pyr_sz = cvSize(imgA->width + 8, imgB->height / 3);
    IplImage *pyrA = cvCreateImage(pyr_sz, IPL_DEPTH_32F, 1);
    IplImage *pyrB = cvCreateImage(pyr_sz, IPL_DEPTH_32F, 1);
    cvCalcOpticalFlowPyrLK(
        imgA,
        imgB,
        pyrA,
        pyrB,
        cornersA,
        cornersB,
        corner_count,
        cvSize(win_size, win_size),
        5,
        features_found,
        features_errors,
        cvTermCriteria(CV_TERMCRIT_ITER | CV_TERMCRIT_EPS, 20, 0.3),
        0
    );//光流法处理

    for (int r = 0; r < corner_count; ++r)
    {
        if (features_found[r] == 0 || features_errors[r] > 550)
        {
            cout << "Error is " << features_errors[r];
            continue;
        }
    }

```

```

        cout << "Got it!" << endl;

        point0[r] = cvPoint(cvRound(cornersA[r].x), cvRound(cornersA[r].y));
        point1[r] = cvPoint(cvRound(cornersB[r].x), cvRound(cornersB[r].y));
        length[r] = sqrt((cornersA[r].x - cornersB[r].x)*(cornersA[r].x -
cornersB[r].x) + (cornersA[r].y - cornersB[r].y)*(cornersA[r].y - cornersB[r].y));//计算
光流长度

        lengthsum += length[r];
        num++;
    }

    lengthstandard = 0.5;

    for (int index = 0; index < num; ++index)
    {
        if (length[index] > lengthstandard)
        {
            cvCircle(imgCR, point1[index], 6, (0, 0, 255), -1);
            cvLine(imgCR, point0[index], point1[index], CV_RGB(255, 0, 0), 4);//画
出光流向量

            movenum++;
            move[movenum] = point1[index];
        }
    }

    num = 0;
    lengthsum = 0;

    //检测光流法得到的运动目标坐标是否在背景减除法前景中运动目标区域
    //找到前景中的轮廓
    IplImage *result = NULL;
    Mat FR = cvarrToMat(imgFR);
    findContours(FR, contours, hierarchy, RETR_TREE, CHAIN_APPROX_SIMPLE, Point());

```

```

for (int h = 0; h < contours.size(); h++)//遍历所有轮廓
{
    if (contours[h].size() > 5)
    {
        for (int j = 0; j < (int)contours[h].size(); j++)//遍历一个轮廓中的所
        有点
        {
            CvPoint pt = cvPoint(contours[h][j].x, contours[h][j].y);//存储轮
            廓中的点

            avg_ptx += pt.x;
            avg_pty += pt.y;
        }
        avg_ptx = avg_ptx / (int)contours[h].size();
        avg_pty = avg_pty / (int)contours[h].size();
        for (int t = 0; t < movenum; t++)
        {
            distance = sqrt((avg_ptx - move[t].x)*(avg_ptx - move[t].x) +
            (avg_pty - move[t].y)*(avg_pty - move[t].y));//计算轮廓点与运动角点的距离
            if (distance > distancestandard)//距离过大，认为轮廓有问题，即背景
            减除法得到的运动区域有问题，将该区域涂黑
            {
                drawContours(FR, contours, h, CV_RGB(0, 253, 0), CV_FILLED);
                result = &IplImage(FR);
            }
        }
        avg_ptx = 0;
        avg_pty = 0;
    }
}

for (int o = 0; o < MAX_CORNERS; o++)
{
    move[o] = NULL;
}

```

```
//反转图片并显示
cvFlip(imgCR, imgCR, 0);
cvShowImage("result", imgFR);
Mat resultmat = cvarrToMat(imgFR);
sprintf(resultName, "%s%d%s", "Y:\\\\GraduationProject\\\\result\\\\img", i, ".jpg");
imwrite(resultName, resultmat);

cvFlip(imgCR, imgCR, 0);
cvShowImage("video", imgCR);

Mat videomat = cvarrToMat(imgCR);
sprintf(videoName, "%s%d%s", "Y:\\\\GraduationProject\\\\video\\\\img", i, ".jpg");
imwrite(videoName, videomat);
cvWaitKey(2);
}
```



## 附录 2

原文如下:

### **Global Path Planning Using Modified Firefly Algorithm**

Xiaochao Chena, Ming Zhou<sup>\*</sup>, Jian Huang<sup>a</sup>, Zhiwei Luob

<sup>a</sup> Laboratory of Image Processing and Intelligent Control, School of Automation, Huazhong University of Science and

Technology, No. 1037, Hongshan District, Wuhan, Hubei, 430074, China

<sup>b</sup> Advanced Research Organization, Kobe University, Kobe 657-8501, Japan

[huang\\_jan@mail.hust.edu.cn](mailto:huang_jan@mail.hust.edu.cn)

**Abstract:** Firefly algorithm is widely used in the tackling of optimization problems. This paper proposed a global path planning algorithm based on the modified firefly algorithm (PPMFA) in order to find an optimal path under multiple objective functions. Owing to the low convergence speed and inaccurate local search ability of the standard firefly algorithm (SFA), the Gaussian random walk is proposed to replace the fixed step size of the SFA so as to improve the random search ability. Incorporating a double check method during the iteration process, the success rate of the movement of each firefly is increased. In order to measure the distance between any two fireflies (which represent two paths), the conception and calculation method of Path Center (PC) is proposed. Simulation results show that compared with the particle swarm optimization (PSO) and SFA, the proposed algorithm outperforms both of the algorithms in terms of convergence speed and accuracy.

### 1. INTRODUCTION

Path planning (PP) problem is a hot topic in many kinds of research and application field. Traditional path planning algorithms mainly focus on finding a feasible path from the start point to the destination. However, with the increasing requirements of energy saving, time-saving and other requirements, an optimal path is needed to satisfy some certain objective such as the shortest length, smoothness of the path, the safety of the path and so on.

There are plenty of path planning algorithms. Artificial potential field [1], fuzzy logic [2], simulated annealing (SA) [3] algorithm are the most widely used traditional path planning algorithms. Meanwhile, with the development of swarm intelligence algorithms, genetic algorithm (GA) [4] inspired by the Darwin's Law, ant colony algorithm (AC) [5] inspired by the food hunting behavior of ant, particle swarm optimization (PSO) [6] algorithm deduced from the hunting behavior of fishes or birds become more and more popular. As for the construction of the solution space, the grid method is a common approach to model the working environment owing to its ease of implementation.

Firefly algorithm (FA) proposed by Xin-she Yang at Cambridge is a novel metaheuristic algorithm which is inspired by the mating behavior of the fireflies in the summer sky [7]. Unlike other kinds of

optimization methods, FA can converge into all of the global and local optima simultaneously and is easy to implement. Research shows that the FA can outperform both PSO and random search [7]. Since its implementation, FA becomes more and more popular and is widely used in the tackling of the almost every kind of optimization problems such as image retrieval [8], economic load dispatch [9], variable structure optimization [10] and so on. However, there are still some disadvantages of these algorithms when they are applied in the path planning problems. For GA, it is difficult to encode the path if there are too many path points [4]; AC suffers a heavy burden of calculation; simulated annealing algorithm (SA) [3] obtains a slow convergence speed caused by the deterministic property of the iteration process; fuzzy logic is dependent on the human experience [2]; PSO is easily got to premature which will lead to an inaccurate solution and SFA obtains a slow convergence speed and inaccurate solution [7]. In this paper, we proposed a global path planning algorithm based on the modified firefly algorithm (PPMFA) to find an optimal path in terms of certain objective function. In order to measure the distance between any two fireflies which represent two paths in the solution space respectively, the conception and calculation process of Path Center (PC) is proposed. Then we assume that the distance between two PC points is the distance between the corresponding two paths. Meanwhile, the modified firefly algorithm maintains two features to improve the performance of the SFA. Firstly, the Gaussian random walk is proposed to replace the fixed random step size of SFA so as to improve the random search performance of the firefly algorithm. Secondly, the so-called double check method is proposed in order to increase the success rate of the movement of the fireflies. Throughout these improvements, the PPMFA can outperform both PSO and SFA. The rest of the paper is organized as follows: it outlines the preliminary of the path planning task in section 2. The modified firefly algorithm is introduced in section 3. Simulation results and comparison studies among PPMFA, SFA and PSO are presented in section 4. Section 5 concludes the paper.

## 2. PRELIMINARY OF THE PATH PLANNING PROBLEM

### 2.1 Encode of the fireflies and distance calculation

In this paper we assume that each firefly corresponding to a unique candidate path in the solution space. Suppose each path has  $n$  main path points in the  $(x, y)$  plane, each path point is denoted as  $p_i = (x_i, y_i)$ , so we can encode each firefly (path) by:

$$x_i = (x_1, y_1, \dots, x_n, y_n) \quad (2-1)$$

where  $i$  represents the  $i$ th firefly. In order to smooth the path, the B-spline technique is used to increase the path points to  $N$  ( $N \gg n$ ). Among the  $n$  main path points,  $m$  new path points are interpolated between two adjacent path points using the above mentioned B-spline technique. In the FA, parameter  $r_{ij}$  denotes the distance between  $i$ th firefly and  $j$ th firefly. In the path planning task,  $r_{ij}$  should be the distance between path  $i$  and path  $j$  which is not the same as the Euclidean distance in the two or three-dimensional space. Therefore, a criterion is needed to quantitate the distance between any two paths. Considering the situation in the plane

geometry, the geometry center of a plane figure can be calculated as [11]:

$$\text{Geo} = (x, y) = \left( \frac{\int xf(x)dx}{\int f(x)dx}, \frac{\int yf(y)dy}{\int f(y)dy} \right) \quad (2-2)$$

where  $f(x)$ ,  $f(y)$  are the function value at  $x$  and  $y$  respectively,  $\text{Geo}$  is the geometric center of the plane figure. Considering each path as an unclosed plane figure (UPF) and each path segment corresponding to an edge of the UPF, then the UPF has  $n - 1$  edges (Path Segment) which can be denoted as:  $L_i\{p_i, p_{i+1}\}$  and it is obvious that the geometry center of each edge is the midpoint, then the geometry center of  $i$ th segment can be defined as follows:

$$\text{Geo}_i = \left( \frac{p_{xi} + p_{x(i+1)}}{2}, \frac{p_{yi} + p_{y(i+1)}}{2} \right) \quad (2-3)$$

Connect each geometric center for each edge in turn until the last one, then we have  $n-2$  edges, calculate the geometry center for these  $n-2$  edges and connect them in turn, then we get  $n-3$  edges, repeat the above process until there is only one geometry center left, then we call this point as the Path Center (PC) of the path which can be represented as:

$$\text{PC}(i) = (p_{cxi}, p_{cyi}) = \text{Geo}(p_1, \dots, p_n) \quad (2-4)$$

where  $\text{PC}(i)$  denotes the PC of  $i$ th path,  $p_i (i = 1, 2, \dots, n)$  denote the coordinates of the  $n$  path points,  $\text{Geo}(\cdot)$  denote the operator of calculating the PC of a path.

We assume that the distance between any two fireflies is the distance between the corresponding PCs, then we have:

$$r_{ij} = \text{dis}[\text{PC}(i), \text{PC}(j)] \quad (2-5)$$

## 2.2 Cost Function

In the real application, the path length always associated with energy and time needed to move from the start point to the destination. The shorter the path is, the less energy and time needed for the action subject such as the unmanned vehicle or autonomous robot. Therefore, the path length is considered as one of the main requirements and we have:

$$\text{PL}(i) = \sum_{i=1}^{n-1} \text{dis}(i, i+1) \quad (2-6)$$

where  $\text{PL}(i)$  is the length of  $i$ th path,  $\text{dis}(i, i+1)$  is the distance between consecutive path points  $i$  and  $i+1$ .

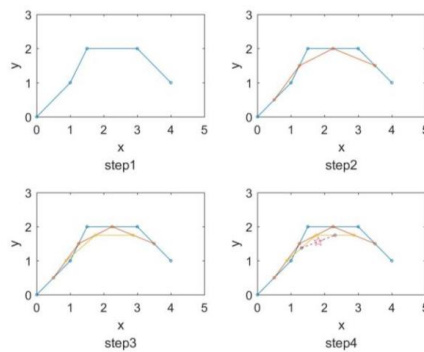


Figure 1 The calculation process of the Path Center (PC).

The distance between a path point and the center of the obstacle is:

$$d_{ij} = \sqrt{(x_j - p_{xi})^2 + (y_j - p_{yi})^2} \quad (2-7)$$

Suppose there are M obstacles in the working space, for each path we define the variable of safety as:

$$\text{colli} = \sum_{i=1}^M \text{mean} \left( \sum_{j=1}^n \max \left( 1 - \frac{\text{dis}_{ij}}{R_i}, 0 \right) \right) \quad (2-8)$$

where  $\max \left( 1 - \frac{\text{dis}_{ij}}{R_i}, 0 \right)$  is the indicator of whether there is collision between obstacle i and path point j or not and the quantitative description of how badly the collision is :

$$\begin{cases} \max \left( 1 - \frac{\text{dis}_{ij}}{R_i}, 0 \right) > 0 & \text{collision} \\ \max \left( 1 - \frac{\text{dis}_{ij}}{R_i}, 0 \right) = 0 & \text{free} \end{cases} \quad (2-9)$$

$\text{colli}(\cdot)$  represents the cost value generated from the collision between a path and all the obstacles in the working space. Then the total cost function is described as following:

$$\text{Cos t}(i) = \text{PL}(i) \cdot (1 + 9 \cdot \text{colli}(i)) \quad (2-10)$$

Where  $\text{Cos t}(i)$  represent the fitness value for firefly i (path i), - is the weight factor.

### 3. MODIFIED FIREFLY ALGORITHM

In this section, we proposed the modified firefly algorithm which obtains two features so as to improve the performance of the SFA. Firstly, a brief introduction of SFA will be given in 3.1.

#### 3.1 Standard Firefly Algorithm

For the reason of simplifying the construction of FA and ignore those features which are not related to our goal. There are three assumptions in the SFA [12]. Firstly, we assume that all fireflies have only one sex and are attracted to each other. Second, the attractiveness is associated with the intensity of the light being emitted by the fireflies which means that the brighter bug will attract the less capable bug to move towards her. Thirdly, the brightness of a firefly is associated or determined by the landscape of the objective function which in this paper is the cost function. There are two important issues in the FA, including the variation of the light intensity and the formulation of the attractiveness. Considering two fireflies i and j, the distance between them is denoted as above mentioned  $r_{ij}$ , the light intensity seen by the adjacent firefly can be practically approximated as the following Gaussian form:

$$I(r) = I_0 e^{-\gamma r^2} \quad (3-1)$$

where  $\gamma$  is the light absorption coefficient which is a crucial factor of the convergence rate,  $I_0$  is the original light intensity.

As the attractiveness of a firefly is proportional to the light intensity seen by the adjacent fireflies, we can now define the attractiveness  $\beta$  of a firefly by:

$$\beta = \beta_0 e^{-\gamma r^2} \quad (3-2)$$

Where  $\beta_0$  is the attractiveness at  $\gamma = 0$ , for most of our implementations, we can take  $\beta_0 = 1$ .

When the less bright firefly  $i$  is attracted by the brighter firefly  $j$ , the movement can be determined by:

$$x_i = x_i + \beta_0 e^{-\gamma r^2} (x_j - x_i) + \alpha \varepsilon_i \quad (3-3)$$

In (3-3), the second term of the right hand is generated from the attractiveness of firefly  $j$  to firefly  $i$ . The third term is the randomization with  $D$  being the step size which is  $\alpha$  fixed value in SFA,  $\varepsilon_i$  is a vector of random numbers drawn from a Gaussian distribution or uniform distribution. For example, the simplest form of  $\varepsilon_i$  is  $\varepsilon_i = rand - 0.5$  rand, where rand is a random number generator uniformly distributed in  $[0, 1]$ .

In this paper, the one which has a smaller Cost function value is brighter than the one with a larger Cost function value.

As for metaheuristic methods, the diversification is crucial to their performance. However, the parameter  $D$  in the SFA is a fixed value which is not good for the diversity of the firefly population.

### 3.2 Gaussian Random Walk

As for metaheuristic methods, the diversification is crucial to their performance. However, the parameter  $D$  in the SFA is a fixed value which is not good for the diversity of the firefly population.

As mentioned in [7], the Gaussian distribution is widely used in the description of random variables of the actual physical world. Therefore, instead of choosing the step size as a fixed value similar to SFA, we proposed a Gaussian random walk scheme during the search process in order to maintain the diversification of the firefly population. In this application, the dimension of the firefly is  $2n$  which means a tremendous number of candidate paths in the searching space. Therefore, our primary goal is not to converge all the fireflies into one point but find a better solution with the Cost function value as small as possible.

In this paper, we can calculate  $\alpha$  by:

$$\alpha = \alpha_0 \theta \quad (3-4)$$

where  $\theta$  is a Gaussian random number generated by [13]:

$$\begin{cases} \theta = \sqrt{-\frac{\sigma^2 \ln 4 \rho^2}{2}} + \mu & \rho < 0.5 \\ \theta = \sqrt{-\frac{\sigma^2 \ln 4 (1 - \rho)^2}{2}} + \mu & \rho > 0.5 \end{cases} \quad (3-5)$$

In (3-5),  $\sigma$ ,  $\mu$  represent the standard deviation and the mathematical mean of the Gaussian distribution,  $\rho$  is the uniformly distributed random number. In the real application, we should choose the random value which is less than 1. Through the Gaussian random walk, the diversification of the firefly algorithm can be guaranteed.

### 3.3 Double Check Method

Let  $I(f(x_i))$ ,  $I(f(x_j))$  denote the light intensity of firefly  $i$  and  $j$  respectively. If  $I(f(x_i)) < I(f(x_j))$ , which means firefly  $j$  is brighter than firefly  $i$ , then  $i$  should fly towards  $j$  ( $x_i$  turns to be  $x'_i$ ). However, there is no guarantee that firefly  $i$  will actually get closer to firefly  $j$  because the movement of firefly is not deterministic but has a certain level of uncertainty due to the random walk of metaheuristic methods. Therefore, in this paper, we proposed a method called double check to improve the success rate of the firefly's movement towards the brighter firefly.

So if we have  $I(f(x'_i)) < I(f(x_i))$ , which means that the firefly failed to move close to the brighter firefly, then another chance will be given to the less bright firefly, the second movement can be described as:

$$x_i = x'_i + \beta_0 e^{-\gamma r^2} (x_j - x'_i) + \alpha \varepsilon_i \quad (3-6)$$

By doing so, we can improve the success rate of the movement of each firefly. Based on these features, the basic steps of the PPMFA can be summarized as the pseudo code shown in Figure 2.

Throughout the two features, the performance of FA can be improved.

---

```

Object function:  $cost(x)$ ,  $x = (x_1, y_1, \dots, x_n, y_n)^T$ 
Generate initial population of firefly  $x_i$  ( $i = 1, 2, \dots, N$ )
Light intensity  $I_i$  at  $x_i$  is determined by  $cost(x_i)$ 
While  $t < MaxIteration$  do
  for  $i = 1 : N$  all  $N$  fireflies do
    for  $j = 1 : N$  all  $N$  fireflies do
      if  $cost(x_i) > cost(x_j)$  then
        move firefly  $x_i$  to firefly  $x_j$  ( $x_i$  turns to be  $x'_i$ )
         $x_i = x'_i + \beta_0 e^{-\gamma r^2} (x_j - x'_i) + \alpha \varepsilon_i$ 
         $\alpha = \alpha_0 \theta$ 
        where  $\theta$  is the Gaussian random number
      end if
      if  $cost(x'_i) > cost(x_i)$  then
        move firefly  $x'_i$  to firefly  $x_j$ 
         $x_i = x'_i + \beta_0 e^{-\gamma r^2} (x_j - x'_i) + \alpha \varepsilon_i$ 
      end if
      vary attractiveness with distance  $r$  via  $exp(-\gamma r^2)$ 
      evaluate new solution and update light intensity
    end for
  end for
end while
Post process results and visualization

```

---

**Figure 2 Pseudo code of the PPMFA algorithm**

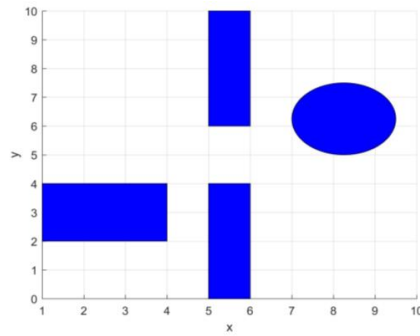
#### 4 SIMULATIONS

In order to test the effectiveness and improvements of the proposed PPMFA, a large number of simulations in terms of different destinations and iteration times are conducted. For the reason that there is no mathematical proof of the convergence of most metaheuristic algorithms, therefore the Monte Carlo method is used in the validation of the effectiveness of these algorithms.

##### 4.1 Preliminary of the simulation

###### 1) Modeling of the environment

In the real world, obstacles maintain all kinds of shape and size which make it is hard to model them, therefore, in order to simplify the task of modeling of the obstacles, plane circles are used. As shown in Figure 3, the rectangle obstacle is the modeled as a set of circles.

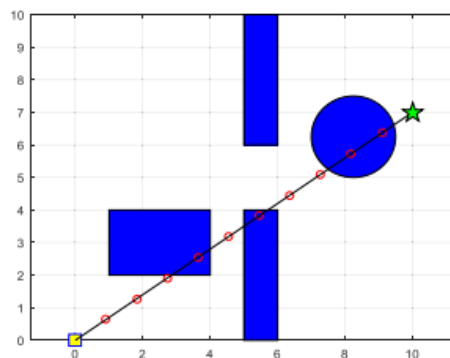


**Figure 3 Modeling of the simulation environment**

## 2) Initialization of the fireflies

Initialization plays the crucial importance of the metaheuristic methods. In the path planning task, our goal is to find a path with the cost function value as small as possible. However, in the search space, a candidate path is just a point in the  $2n$ -dimension space which makes it hard to find the optimal point (path) using a few fireflies compared with the tremendous candidates in the search space. Thus, if we initialize the whole population by totally randomization, it will make the search process in the dark and ignore much useful prior information of the task.

In order to solve this problem, we initial one of the firefly using the specific path which directly from the start point to the destination as the initial reference path (IRP) as shown in Figure 4. It is obvious that the IRP has the shortest path length and it can be a guidance for the algorithm. The neighborhood of IRP could be one of the local search area which may help the algorithm find the optimal path with a faster speed. The red circles are the main path points, among these main path points, the B-spline technique is utilized to smooth the path and increase the number of path points. As for the rest of the firefly population, random initialization is applied.



**Figure 4 Initial reference path**

## 4.2 Initialization of the Simulation

Comparison studies among the PPMFA, PSO and SFA are conducted so as to test the performance of the three algorithms and validate the improvements of the proposed algorithm. The three algorithms share the same Cost function.

### 1) Particle Swarm Optimization initialization

PSO is a widely used optimization algorithm which is easy to implement. During the iteration process, the movement of each particle can be defined as [6]:

$$v_{id}^{t+1} = w \times v_{id}^t + c_1 r_1 (p_{id} - x_{id}^t) + c_2 r_2 (p_{gd} - x_{id}^t) \quad (4-1)$$

$$x_{id}^{t+1} = x_{id}^t + v_{id}^{t+1} \quad (4-2)$$

where  $v_{id}^t \in [v_{min}, v_{max}]$ ,  $w$  is the weighting factor and is set to be  $w = 0.98^t$ ,  $c_1 = c_2 = 1.5$  is the learning factors,  $r_1, r_2$  are the random numbers drawn from uniform distribution  $p_{id}$ ,  $p_{gd}$  represent the historical best position of  $i$ th particle and the current global best particle respectively. At the right hand of the equation, the first term is the weighted velocity inheritance, the second term is the self-cognition of the particle, and the third term represents the information sharing among the particles. As for the initialization of the particles (paths), the above mentioned method in section 4.1 is applied.

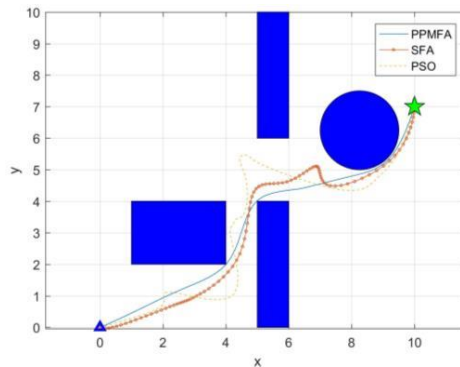
## 2) SFA and PPMFA initialization

In SFA and PPMFA, they share the basic parameters. Meanwhile, the PPMFA obtain the two features mentioned above during the iteration process. As for the basic parameters, the light absorption coefficient is set to be  $\gamma = 0.01$ , randomization parameter is set to be  $\beta\alpha_0 = 0.2$ . During the generation of Gaussian random number of the proposed PPMFA algorithm, we choose  $\sigma = 1$ ,  $\mu = 0$ .

## 4.3 Comparison studies

As mentioned in section 1, both SFA and PSO have some disadvantages, the proposed algorithm is aimed at tackling these disadvantages. Therefore, in order to validate these improvements several simulations with different conditions are conducted. Metaheuristic methods are non-deterministic algorithms, the simulation result for each simulation will be slightly different from each other. 50 successful simulation results for each algorithm are used to calculate the average Cost function value to the corresponding algorithm.

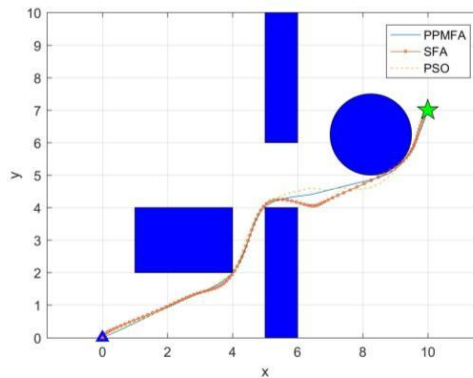
The start point of all of the simulation is set to be (0, 0), the destination 1 and destination 2 are set to be (10, 6) and (10, 7) respectively. The comparison results for the three algorithms after 20 iterations are given by Figure 5 and Figure 6 shows the comparison results after 50 iterations with the destination settled as (10, 7). The average Cost function value of 50 successful simulations for different conditions are given in Table 1 and Table 2.



**Figure 5 Path Planning results after 20 iterations**

It is obviously shown in Figure 5 that the PPMFA can obtain a much better solution compared with the SFA and PSO after 20 iterations and we can draw the conclusion that the proposed algorithm is faster than the other two algorithms in terms of convergence speed.





**Figure 6 Path Planning results after 50 iterations**

Compare Figure 5 with Figure 6, it is clear that the planning results of PPMFA are close to each other after 20 and 50 iterations which means that the PPMFA can find the optimal solution within 20 iterations. Meanwhile, the SFA and PSO are still trying to converge to the optimal solution. Therefore, the improvements of PPMFA can be validated.

**Table 1 Results of the three algorithms after 20 iterations**

star t	goal	Cost [PSO]	Cost [SFA]	Cost [PPM FA]	Short est distan ce
(0,	(10,	19.24	15.40	13.18	12.20
0)	7)	71	89	17	66
(0,	(10,	14.89	15.67	12.18	11.66
0)	6)	34	53	15	19

1) Shortest distance means the direct distance between original and the destination without considering the obstacles, similarly hereinafter 2) Cost[\*] means the average Cost of 50 successful path planning for each algorithm, \*={PSO, PPMFA, SFA}, similarly hereinafter

The Monte Carlo experiment results for the three algorithms shown in Table 1 and 2 shows that under the same condition, the proposed algorithm can find the optimal path in a faster speed, Therefore, we can clearly draw the conclusion that the proposed algorithm outperforms both PSO and SFA in terms of convergence speed and accuracy, then the improvements of the proposed algorithm are validated.

## 5 CONCLUSION

In this paper, we proposed a global path planning method based on the modified firefly algorithm (PPMFA) to obtain an optimal path from the start point to the destination with the shortest path length and without colliding with any obstacles in the working space. In order to obtain the distance between two fireflies (paths), the conception and calculation method of Path Center (PC) was proposed. Two features are applied to the standard firefly algorithm in order to improve the performance of the firefly algorithm. Firstly, the fixed step size of SFA is replaced by the Gaussian random walk so as to diversify the population

which will improve the local search ability of the FA; incorporating with the double check method, the success rate of the movement for each firefly is increased. Comparison studies show that the proposed algorithm outperforms both the PSO and SFA in terms of the accuracy and convergence speed. The proposed algorithm can find the feasible solution which is close to the shortest distance less than 20 iteration times. The effectiveness and improvements are validated. In future work, we are going to apply the proposed algorithm in the path planning work of the unmanned autonomous vehicle which is used to the autonomous power patrol system.

#### ACKNOWLEDGEMENT

This work was supported by the special program for technology innovation of Hubei Province (2016AAA039), the National Natural Science Foundation of China (61473130) and outstanding young talents project of Hubei Natural Science Foundation of China (2015CFA047).

译文如下:

#### 使用改进萤火虫算法的全局路径规划

*Xiaochao Chen<sup>a</sup>, Ming Zhou<sup>b\*</sup>, Jian Huang<sup>a</sup>, Zhiwei Luo<sup>b</sup>*

<sup>a</sup> Laboratory of Image Processing and Intelligent Control, School of Automation, Huazhong University of Science and Technology, No. 1037, Hongshan District, Wuhan, Hubei, 430074, China

<sup>b</sup> Advanced Research Organization, Kobe University, Kobe 657-8501, Japan  
huang\_jan@mail.hust.edu.cn

**摘要:** 萤火虫算法被广泛应用于优化问题的处理。本文提出了一种基于改进萤火虫算法 (PPMFA) 的全局路径规划算法, 以便在多个目标函数下寻找最优路径。针对标准萤火虫算法(SFA)收敛速度慢、局部搜索能力不准确的问题, 提出高斯随机游走来代替 SFA 的固定步长, 以提高随机搜索能力。在迭代过程中加入双重检查方法, 提高了每只萤火虫运动的成功率。为了测量任意两只萤火虫 (代表两条路径) 之间的距离, 提出了路径中心 (PC) 的概念和计算方法。仿真结果表明, 与粒子群优化 (PSO) 和 SFA 相比, 所提算法在收敛速度和精度方面均优于这两种算法。

#### 1 简介

路径规划 (PP) 问题是许多研究和应用领域的热门话题。传统的路径规划算法主要专注于寻找一条从起点到终点的可行路径。但是, 随着节能、省时等要求的不断提高, 需要一条最优路径来满足一定的目标, 如最短路径、路径平滑度、路径安全性等。

目前有很多路径规划算法。人工势场[1]、模糊逻辑[2]、模拟退火(SA)[3] 算法是应用最广泛的传统路径规划算法。同时, 随着群体智能算法的发展, 受达尔文定律启发的遗传算法 (GA) [4], 受蚂蚁觅食行为启发的蚁群算法 (AC) [5], 从鱼类或鸟类的狩猎行为推导出的粒子群优化算法 (PSO) [6] 越来越流行。对于解决方案空间的构建, 网格法因其易于实现而成为建模工作环境的常用方法。

由剑桥大学的 Xin-she Yang 提出的萤火虫算法 (FA) 是一种新颖的元启发式算法, 其灵感来自夏季天空中萤火虫的交配行为[7]。与其他类型的优化方法不同, FA 可以同时收敛到所有全局和局部最优, 并且易于实现。研究表明, FA 的性能优于 粒子群优化算法 PSO 和随机搜索 [7]。自实施以来, FA 变得越来越流行, 并被广泛应用于解决几乎所有类型的优化问题, 如图像检索 [8]、经济负荷调度 [9]、可变结构优化 [10] 等。

然而, 这些算法在应用于路径规划问题时仍然存在一些缺点。对于遗传算法 GA, 如果路径点太多, 则很难对路径进行编码 [4]; 蚁群算法 AC 的计算负担很重; 模拟退火算法 SA [3] 由于迭代过程的确定性导致收敛速度慢; 模糊逻辑依赖于人类经验[2]; PSO 很容易过早收敛, 这将导致不准确的解决方案, SFA 导致缓慢的收敛速度和不准确的解决方案 [7]。

在本文中, 我们提出了一种基于改进萤火虫算法的全局路径规划算法 (PPMFA), 以寻找特定目标下的最佳路径。为了测量分别代表解空间中两条路径的任意两只萤火虫之间的距离, 提出了路径中心 (PC) 的概念和计算过程。假设两个 PC 点之间的距离就是对应的两条路径之间的距离。同时, 改进的萤火虫算法保留了两个特征来提高 SFA 的性能。首先提出高斯随机游走来代替 SFA 的固定随机步长, 以提高萤火虫算法的随机搜索性能。其次, 为了提高萤火虫运动的成功率, 提出了所谓的双重检查法。在这些改进过程中, PPMFA 能够表现出优于 PSO 和 SFA 的性能。

论文的其余部分安排如下: 第 2 节概述了路径规划任务的初步研究。第 3 节介绍了改进的萤火虫算法 PPMFA 中的仿真结果和比较研究, 第 4 节介绍了 SFA 和 PSO 算法。第 5 节对本文进行总结。

## 2 对路径规划问题的初步研究

### 2.1 萤火虫的编码和距离计算

在本文中, 我们假设每个萤火虫对应于解空间中的一个唯一候选路径。假设每条路径在  $(x, y)$  平面上有  $n$  个主要路径点, 每条路径点表示为  $p_i = (x_i, y_i)$ , 所以我们可以以下列方式对每个萤火虫 (路径) 进行编码:

$$x_i = (x_1, y_1, \dots, x_n, y_n) \quad (2-1)$$

其中  $i$  代表第  $i$  只萤火虫, 为了使路径平滑, 使用 B-spline 技术将路径点增加至  $N$  ( $N \gg n$ )。在  $n$  个主要路径点中, 使用上述 B-spline 技术在两条相邻路径之间插入新的路径点。在 FA 算法中,  $r_{ij}$  参数表示第  $i$  个路径与第  $j$  个路径之间的距离, 而与二维或三维空间里的欧几里得距离不同。因此需要一个标准来量化任意两条路径之间的距离。

考虑到平面几何的情况, 平面图形的中心可以用如下方式计算[11]:

$$Geo = (x, y) = \left( \frac{\int xf(x)dx}{\int f(x)dx}, \frac{\int yf(y)dy}{\int f(y)dy} \right) \quad (2-2)$$

其中  $f(x)$ ,  $f(y)$  分别为  $x$  和  $y$  处的函数值, Geo 为平面图形的几何中心。将每条路径视为一个未闭合的平面图形 (UPF), 那么每个路径段对应于 UPF 的一条边, 那么 UPF 有  $n-1$  条边 (路径段), 可以表示为:  $L_i\{p_i, p_{i+1}\}$ , 显示每条边的几何中心是中点, 那么第  $i$  段的几何中心可以定义

如下：

$$Geo_i = \left( \frac{p_{xi} + p_{x(i+1)}}{2}, \frac{p_{yi} + p_{y(i+1)}}{2} \right) \quad (2-3)$$

依次连接每条边的每个几何中心，直到最后一个，那么我们有  $n-2$  条边，计算这  $n-2$  条边的几何中心并依次连接他们，然后我们可以得到  $n-3$  边，重复上述过程直到只有一个几何中心，我们将这一点成为路径的中心（PC），可以表示为：

$$PC(i) = (p_{cxi}, p_{cyi}) = Geo(p_1, \dots, p_n) \quad (2-4)$$

其中  $PC(i)$  表示第  $i$  个路径的  $PC$ ， $p_i (i = 1, 2, \dots, n)$  表示第  $n$  个路径点的坐标  $Geo(\cdot)$  表示计算路径  $PC$  的算子。

我们假设，任意两只萤火虫之间的距离是对应的  $PC$  之间的距离，那么我们有：

$$r_{ij} = dis[PC(i), PC(j)] \quad (2-5)$$

图 1 给出了计算过程的示例。

## 2.2 成本函数

在实际应用中，路径长度总是与从起点移动到目的地所需的能量和时间相关。路径越短，无人机或自动机器人等动作主体所需的时间和能量就越少，因此，路径长度被视为主要要求之一，从而我们有：

$$PL(i) = \sum_{i=1}^{n-1} dis(i, i+1) \quad (2-6)$$

其中  $PL(i)$  是第  $i$  条路径的长度， $dis(i, i+1)$  是连续路径点  $i$  和  $i+1$  之间的距离。

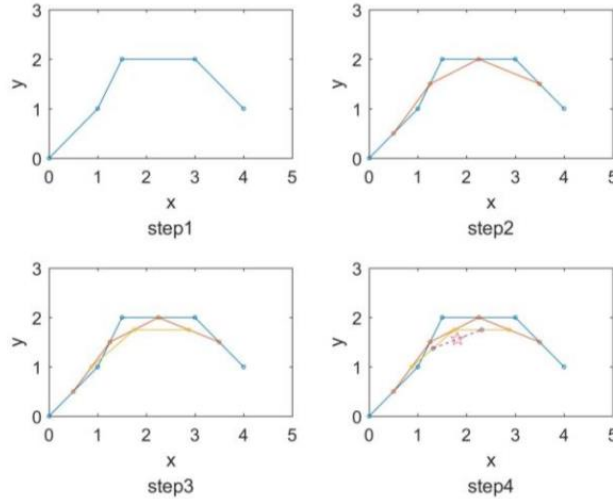


图 1 路径中心（PC）的计算过程

路径点与障碍中心点之间的距离为：

$$d_{ij} = \sqrt{(x_j - p_{xi})^2 + (y_j - p_{yi})^2} \quad (2-7)$$

假设工作空间有  $M$  个障碍物，对于每条路径我们定义安全变量为：

$$colli = \sum_{i=1}^M \text{mean} \left( \sum_{j=1}^H \max \left( 1 - \frac{dis_{ij}}{R_i}, 0 \right) \right) \quad (2-8)$$

其中 $\max(1 - \frac{dis_{ij}}{R_i}, 0)$ 是障碍物  $i$  与路径点  $j$  是否会发生碰撞以及碰撞严重程度的定量描述。

$$\begin{cases} \max\left(1 - \frac{dis_{ij}}{R_i}, 0\right) > 0 & \text{collision} \\ \max\left(1 - \frac{dis_{ij}}{R_i}, 0\right) = 0 & \text{free} \end{cases} \quad (2-9)$$

$colli(\cdot)$ 表示从路径与工作中所有障碍物之间的碰撞得出的成本值，总成本函数描述如下：

$$Cos t(i) = PL(i) \cdot (1 + 9 \cdot colli(i)) \quad (2-10)$$

其中 $Cos t(i)$ 表示萤火虫  $i$  (路径  $i$ )。

### 3 改进萤火虫算法

在本节中，我们提出了改进算法，该算法具有两个特征，以提升 SFA 的表现。首先，SFA 的简介将在 3.1 中给出。

#### 3.1 标准萤火虫算法 (SFA)

为了简化 FA 的构造并忽略那些与我们的目标无关的特征。在 SFA 中有三个假设[12]。

首先，我们假设所有萤火虫只有一种性别，并且彼此吸引。吸引力与萤火虫的亮度有关，这意味着明亮的虫子会吸引发光能力较差的虫子对着他进行移动。

萤火虫的亮度是与目标函数的分布相关或由其决定的，在本文中即为成本函数。

在 FA 中有两个重要的问题，分别是光强度的变化和吸引力的构成。

考虑到两只萤火虫  $i$  和  $j$ ，他们之间的距离被表述为 $r_{ij}$ ，被相邻的萤火虫看到的光强度实际上可以近似为如下的高斯形式：

$$I(r) = I_0 e^{-\gamma r^2} \quad (3-1)$$

其中 $\gamma$ 是光吸收系数，这是一个对于收敛速度至关重要的因子， $I_0$ 是原始光强度。

由于萤火虫的吸引力与相邻萤火虫看到的光强度成比例，我们现在可以通过以下方式定义萤火虫的吸引力 $\beta$ ：

$$\beta = \beta_0 e^{-\gamma r^2} \quad (3-2)$$

其中 $\beta_0$ 是 $\gamma = 0$ 时的吸引力，对于我们的多数情况来说，我们可以取 $\beta_0 = 1$ 。

当较暗的萤火虫被较亮的萤火虫  $j$  吸引时，其动作可以通过如下方式计算：

$$x_i = x_i + \beta_0 e^{-\gamma r^2} (x_j - x_i) + \alpha \varepsilon_i \quad (3-3)$$

在式 3-3 中，等式右侧的第二项来源于萤火虫  $j$  对萤火虫  $i$  的吸引力，第三项是随即项，其中 $\alpha$ 是步长，是 SFA 中的一个固定值， $\varepsilon_i$ 是从高斯分布或均匀分布中提取的一个随机数向量。例如， $\varepsilon_i$ 最简单的形式是 $\varepsilon_i = rand - 0.5$ ，其中  $rand$  是统一分布在 $[0,1]$ 中的随即生成数。

在本文中，具有较小成本函数值的一方相较于具有较大成本函数值的一方更加明亮。

#### 3.2 高斯随机游走

对于元启发式算法，多样化对他们的表现至关重要。然而，参数 $\alpha$ 是 SFA 中的一个固定值，不利于萤火虫种群的多样性。

正如[7]中所提到的，高斯分布被广泛用于描述实际物理世界中的随机变量。因此，代替选择步长作为类似于 SFA 的固定值，我们提出了一个高斯搜索过程中的随机游走方案，从而保持萤火虫种群的多样性。在这个过程中，萤火虫的维度为  $2n$ ，这意味着搜索空间中有大量的候选路径。因此我们的首要目标不是将所有萤火虫汇聚成一个点，而是找到一个成本函数尽可能小的较好的解。

在本文中，我们可以通过如下方式计算：

$$\alpha = \alpha_0 \theta \quad (3-4)$$

其中 $\theta$ 是通过如下方式生成的随机高斯数[13]：

$$\begin{cases} \theta = \sqrt{-\frac{\sigma^2 \ln 4\rho^2}{2}} + \mu & \rho < 0.5 \\ \theta = \sqrt{-\frac{\sigma^2 \ln 4(1-\rho)^2}{2}} + \mu & \rho < 0.5 \end{cases} \quad (3-5)$$

在式 3-5 中， $\sigma$ ， $\mu$ 代表标准偏差和高斯分布的数学平均值， $\rho$ 是均匀分布的随机数。在实际应用中，我们应该选择小于 1 的随机值。通过 **Gaussian Random Walk**，可以保证萤火虫算法的多样化。

### 3.3 双重检查法

分别用 $I(f(x_i))$ ， $I(f(x_j))$ 表示萤火虫*i*和萤火虫*j*的光照强度。如果 $I(f(x_i)) < I(f(x_j))$ ，也就是说萤火虫*j*比萤火虫*i*更明亮，那么萤火虫*i*应飞向萤火虫*j*（ $x_i$ 变为' $x_i$ '）。但是，不能保证萤火虫*i*实际上会更接近萤火虫*j*，因为萤火虫的运动不是确定的，而是具有源于元启发式随机游走的一定程度上的不确定性。因此在本文中，我们提出了一种称为双重检查法的方法提高萤火虫向更亮的萤火虫移动的成功率。

所以，如果 $I(f(x'_i)) < I(f(x_i))$ ，也就是说萤火虫没能靠近更亮的萤火虫，那么将给较暗的萤火虫第二次机会，第二次移动可用如下方式描述：

$$x_i = x'_i + \beta_0 e^{-\gamma r^2} (x_j - x'_i) + \alpha \varepsilon_i \quad (3-6)$$

通过这样做，我们可以提高每只萤火虫动作的成功率。基于这些特点，PPMFA 的步骤可以概括为如图 2 所示的伪代码。

```

对象函数:  $\text{Cost}(i)$ ,  $x = (x_1, y_1, \dots, x_n, y_n)^T$ 
生成萤火虫的初始种群:  $x_i (i = 1, 2, \dots, n)$ 
 $x_i$  处的发光强度  $I_i$  取决于  $\text{Cost}(x_i)$ 
While  $t < \text{最大迭代}$  do
    for  $i=1:N$  all N 萤火虫 do
        for  $j=1:N$  all N 萤火虫 do
            if  $\text{Cost}(x_i) > \text{Cost}(x_j)$  then
                将萤火虫  $x_i$  向萤火虫  $x_j$  移动 ( $x_i$  变为  $x'_i$ )

                
$$x_i = x_i + \beta_0 e^{-\gamma r^2} (x_j - x_i) + \alpha \varepsilon_i$$


                
$$\alpha = \alpha_0 \theta$$

                其中  $\theta$  是高斯随机数
            end if
            if  $\text{Cost}(x'_i) > \text{Cost}(x_i)$  then
                将萤火虫  $x'_i$  向萤火虫  $x_j$  移动

                
$$x_i = x'_i + \beta_0 e^{-\gamma r^2} (x_j - x'_i) + \alpha \varepsilon_i$$

            end if
            通过  $e^{-\gamma r^2}$  随距离  $r$  改变吸引力
            评估新解决方案并更新光照强度
        end for
    end for
end while
发布处理结果并使其可视化
    
```

图 2 PPMFA 算法的伪代码

## 4 仿真

为了测试提出的 PPMFA 的有效性并为其提出改进, 进行了大量不同目的与迭代次数下的仿真。由于对于多数元启发式算法收敛没有数学证明, 所以使用了蒙特卡洛方法来验证算法的有效性。

### 4.1 初步仿真

#### 1) 初步仿真

在现实世界中, 障碍物保持着各种形状和大小, 因此很难对它们进行建模。为了简化障碍物建模的任务, 使用了平面圆。如图 3 所示, 矩形障碍物被建模为一组圈。

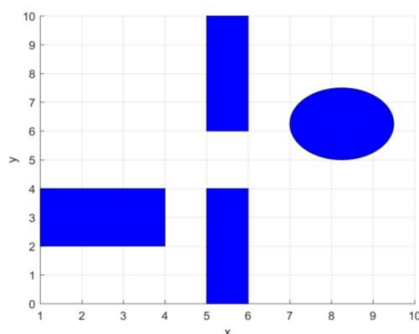


图 3 模拟环境建模

#### 2) 萤火虫初始化

初始化在元启发式算法起着重要的作用。在路径规划任务中, 我们的目标是找到一条代价函数值尽可能小的路径。然而, 在搜索空间中, 候选路径只是  $2n$  空间中的一个点, 这使在巨量的候选路径中使用几只萤火虫找到最佳点 (路径) 十分困难。因此, 如果我们通过随机处理完全初

始化整个种群，那么搜索过程会处于黑暗中且忽略任务的许多有用的先验信息。

为了解决这个问题，我们使用图 4 中所示的一条从起点指向目的地的特殊路径作为初始参考路径（IRP），从而初始化了一只萤火虫。很明显，IRP 具有最短的路径长度，它可以作为算法的指导。这条 IRP 的邻居可能处于本地搜索区域，这可以帮助算法更快找到最优路径。

红圈是主要路径点，在这些主要路径点中，B 样条技术被用于平滑路径并增加路径点数。随机初始化被应用于萤火虫种群的其余部分。

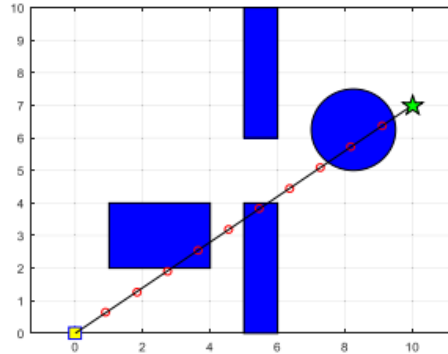


图 4 初始参考路径

#### 4.2 仿真初始化

为了测试 PPMFA、PSO 和 SFA 的表现并验证所提出的改进算法，进行了三者的比较研究。三种算法使用相同的成本函数。

##### 1) Particle Swarm Optimization 初始化

PSO 是一种广泛使用、易于实施的优化算法。在迭代过程中，每个粒子的运动可以定义为[6]:

$$v_{id}^{t+1} = w \times v_{id}^t + c_1 r_1 (p_{id} - x_{id}^t) + c_2 r_2 (p_{gd} - x_{id}^t) \quad (4-1)$$

$$x_{id}^{t+1} = x_{id}^t + v_{id}^{t+1} \quad (4-2)$$

其中  $v_{id}^t \in [v_{min}, v_{max}]$ ,  $w$  是权重因子，被设置为  $w = 0.98^t$ ,  $c_1 = c_2 = 1.5$  是学习因子,  $r_1$ ,  $r_2$  是从均匀分布中抽取的随机数,  $p_{id}$ ,  $p_{gd}$  代表第  $i$  个粒子的历史最佳位置和当前的全局最佳粒子。在方程的右侧，第一项是加权速度继承，第二项是自我认知粒子，第三项代表信息粒子之间的共享。4.1 节中提到的方法则被应用于初始化粒子（路径）。

##### 2) SFA 和 PPMFA 初始化

在 SFA 和 PPMFA 中，他们共享基本的参数，同时 PPMFA 在以上迭代过程中获得了上述提到的两个特征。至于基本参数，光吸收系数设为  $\gamma = 0.01$ ，随机化参数被设为  $\alpha_0 = 0.2$ ，在提出的 PPMFA 算法的高斯随机数生成过程中，我们选择  $\sigma = 1$ ,  $\mu = 0$ 。

#### 4.3 比较研究

如第 1 节所述，SFA 和 PSO 都有一些缺点，本文所提出的算法旨在解决这些缺点。因此，为了验证这些改进，进行了不同条件下的几种仿真。元启发式算法具有非确定性，每次仿真的结果彼此将略有不同。每个算法的 50 次成功仿真结果用于计算对应于每种算法的成本函数平均值。

所有仿真的起点都设置为 (0, 0)，目的地 1 和目的地 2 分别设置为 (10, 6) 和 (10, 7)。



在目的地设置为（10，7）的情况下，图 4 和图 5 分别展示了三种算法在 20 次迭代和 50 次迭代后的对比结果。表 1 和表 2 展示了不同条件下的 50 次成功仿真的平均成本函数值。

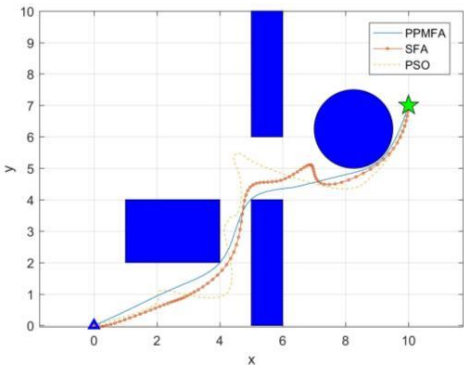


图 5 20 次迭代后的路径规划结果

从图 5 可以明显看出，在 20 次迭代以后，PPMFA 可以获得比 SFA、PSO 更好的解决方案，由此我们可以得出结论：本文提出的算法比其他两个算法具有更快的收敛速度。

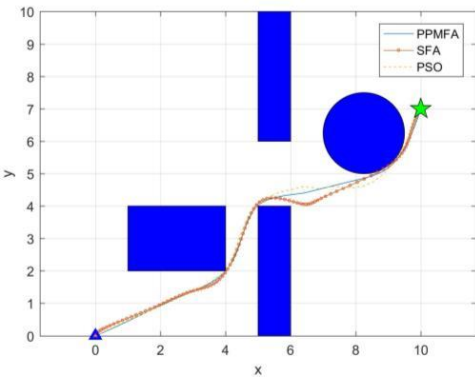


图 6 50 次迭代后的路径规划结果

比较图 5 和图 6，很明显 PPMFA 在 20 次和 50 次迭代后的规划结果彼此接近，也就是说 PPMFA 可以在 20 次迭代以内找到最优解决方案。同时 FSFA 和 PSO 仍然在尝试收敛到最优解。因此，PPMFA 的优化效果可以被验证。

表 1 3 种算法 20 次迭代后的结果

star	goal	Cost	Cost	Cost	Short
t		[PSO]	[SFA]	[PPMFA]	est
				FA]	distan
					ce
(0,	(10,	19.24	15.40	13.18	12.20
0)	7)	71	89	17	66
(0,	(10,	14.89	15.67	12.18	11.66
0)	6)	34	53	15	19

最短距离是指起点于目的地之间不考虑障碍物的直线距离，下同

Cost(\*)表示每条路径规划成功 50 次的平均成本，\*={PSO, PPMFA, SFA}，

表 1 中所示的三种算法的 Monte Carlo 实验结果表明, 在相同条件下, 本文提出的算法可以以更快的速度找到最优路径。因此, 我们可以清楚地得出结论: 本文所提算法在收敛速度与精度方面优于 PSO 和 SFA, 故验证了本文所提算法的改进。

## 5 总结

本文提出了一种基于改进的萤火虫算法 (PPMFA) 的全局路径规划方法, 获得从起点到终点的最优路径 (以最短路径长度, 且不与工作空间中的障碍物冲突为目标)。为了获得两只萤火虫之间的距离, 提出了路径中心 (PC) 的概念与计算方式。为了提高萤火虫算法的性能, 将两个特征应用到标准萤火虫算法中。首先, SFA 中的固定步长被高斯随机游走代替从而使种群多样化以提高 FA 的局部搜索能力; 结合双重检查法, 提高每只萤火虫动作的成功率。对比研究的结果表明, 本文提出的算法在收敛速度与精度方面都优于 PSO 和 SFA。本文提出的算法可以在 20 次迭代以内找到接近于最短距离的可行解。该方法的有效性与改进效果都得以验证。

在今后的工作中, 我们所提出的算法将被应用于自动巡逻系统中的无人驾驶自动汽车的路径规划工作中。

## 致谢

这项工作得到了以下特殊项目的支持: 湖北省科技创新特殊项目 (2016AAA039) 中国国家自然科学基金 (61473130) 和湖北省优秀青年人才项目 (2015CFA047)。