

文章编号: 2096-1618(2017)06-0613-05

# 一种融合 ORB 特征和 Mean Shift 的视觉跟踪算法

闫文龙, 朱毅, 卢军

(成都信息工程大学软件工程学院, 四川 成都 610225)

**摘要:** 由于嵌入式系统对于运算的实时性要求较高, 一些较为成熟的计算机视觉跟踪算法不能很好地满足这一要求。为解决这一问题, 介绍一种结合 ORB(oriented fast and rotated brief)特征点和 Mean Shift 的算法用以视觉跟踪。传统的 Mean Shift 算法运行速度较快, 但在目标被遮挡下容易失效, 导致跟踪结果不理想, 所以决定采用融合 ORB 特征检测算法的 Mean Shift 算法来实现目标跟踪。该算法通过检测目标的初始位置, 并根据与模板匹配的特征点计算出目标在当前帧的尺度以及旋转角度, 从而提高搜索窗口的精度。通过对比实验验证了: 与传统的 Mean Shift 及其改进算法相比, 文中介绍的算法在跟踪物体部分被遮盖时也显示出良好的健壮性, 并且在跟踪的实时性上有良好的表现。

**关键词:** 视觉跟踪; Mean Shift; ORB 特征; 嵌入式平台; 实时性

**中图分类号:** TP391.4

**文献标志码:** A

**doi:** 10.16836/j.cnki.jcuit.2017.06.008

## 0 引言

对于动态物体运用计算机视觉跟踪, 近几十年来一直都是热门问题。一代代学者针对这一问题进行了研究与思考。K. Fukunaga 等<sup>[1]</sup>提出 Mean Shift 算法, 然而 Mean Shift 算法最初只是作为一种聚类分析的方法出现。科马尼丘等<sup>[2]</sup>发现可以将 Mean Shift 算法应用于视频跟踪范畴中, 并在文献[3-4]介绍了它可以应用的领域。继而 Mean Shift 算法运算简单、实时性好等优点被众多学者发现。

当使用 Mean Shift 跟踪动态物体时, 会对采集到的每一帧图像进行色彩的处理, 由于目标的色彩是固定的, 故可以根据感兴趣区域中目标的色彩占比来跟踪目标。目标的每一次移动都会使 Mean Shift 向量得到更新, 通过不断重复更新 Mean Shift 向量, 最终就会逼近目标在视频中的位置, 持续输出所得到的坐标即实现跟踪过程。然而 Mean Shift 算法也存在着一些缺点: 它的跟踪特征缺乏一些必要的更新方法; 在算法运行过程中, 当目标的大小产生改变时, 跟踪窗口的大小并不会随之变化, 这就会严重影响目标的跟踪效果; 并且因为该算法只采用目标的色彩特征作为跟踪特征, 所以当目标处于一种与自身色彩相近的背景中, 或者目标被部分遮挡时会造成算法失效。

针对 Mean Shift 的不足, 大批研究人员提出多种改进办法。由于 SIFT(scale-invariant feature transform)

特征<sup>[5]</sup>在目标发生旋转、目标尺度发生改变、目标被光照射的角度发生改变的情况下, 依旧能够很好地匹配目标<sup>[6]</sup>, 在复杂场景或目标被遮挡的情况下都具有良好的匹配效果, 诸多学者都选择通过引入 SIFT 特征使跟踪效果得到优化。CHEN 等<sup>[7]</sup>结合 SIFT 特征与 Mean Shift, 连续搜寻视场中与目标匹配的更好区域以达到追踪的目的。Dong 等<sup>[8]</sup>运用 SIFT 特征以更新 Mean Shift 的模板带宽和向量的方式来追踪运动朝向和尺度会发生变化的物体。文献[9]用 SIFT 验证 Mean Shift 追踪结果并更新 Mean Shift 所用模板的办法。此外文献[10-14]都提出各式各样对于 SIFT 或是 SURF(speeded up robust features)特征的改进方法。

与 SIFT 特征相似, ORB(oriented fast and rotated brief)特征<sup>[15]</sup>也具有旋转不变性、尺度不变性等特点。文献[16]融合 CAMShift 算法与 ORB 特征, 提出一种自适应的, 利用设定阈值分割目标颜色的算法来提高精确度。如果在算法运行过程中目标跟踪失败, 还会根据获取的目标 ORB 特征点重新检测目标。

由于嵌入式系统对于算法的实时性要求较高, 传统的计算机视觉跟踪技术运算量大, 往往难以达到既准确又快速的要求, 受上述文献的启发, 提出一种利用 ORB 特征以校正 Mean Shift 算法运行结果的新方法。

## 1 算法理论基础

### 1.1 Mean Shift 算法

Mean Shift 算法于 1975 年在文献[1]中首次提

出,其思想是通过计算概率密度的梯度来寻找局部最优方向,其实质是一个梯度下降算法。

随着 Mean Shift 理论的发展,其内容也发生了变化。科马尼丘等<sup>[2]</sup>在研究物体的视觉追踪时首次采用了 Mean Shift。他们先计算出当前帧中,感兴趣区域内的漂移均值向量,将漂移均值向量叠加前一帧中目标位置坐标,并且将这一坐标作为新的坐标起点,重复执行,直到最终结果趋于稳定后就结束。整个过程使用物体的颜色直方图作为要素循环计算,以求到感兴趣区域的“质心”。即用核函数将核带宽内的像素点进行加权以判断物体出现在感兴趣区域内的概率,其中权重由核函数计算得到,跟踪到的模型是不是目标模型需要利用巴氏距离系数判断,最后求取最优化的系数并使其最大化,最终迭代收敛到目标的新位置。从原理上讲,迭代最终是获取图像中目标的新坐标。该算法不需要进行全局性质的穷尽搜索,实时性强,在基于图像处理的目标跟踪中得到广泛应用,具体实现过程可以参考文献<sup>[2]</sup>。

## 1.2 ORB 特征介绍

Rublee 等<sup>[15]</sup>提出 ORB 算法,ORB 特征融合并改进了 FAST(features from accelerated segment test)特征点和 BRIEF(binary robust independent elementary features)特征描述子,与 SIFT 和 SURF 算法相比,ORB 运算速度更快,并且是免费的。

其中,FAST 负责找出感兴趣区域中的角点。首先,它会以某个像素为圆心画出一个圆,这个圆可能包含若干个像素,如果在这些像素中存在一定数量更亮或更暗的像素,则认为圆心是一个角点。FAST 执行速度很快,然而挑选出的角点并不都符合要求。所以需要利用 Harris 方法做进一步筛选。筛选这些点是否合格的标准被称为响应值,它的计算方式为

$$R = \det \mathbf{M} - \alpha (\text{trace} \mathbf{M})^2 \quad (1)$$

式(1)通过计算响应值的大小,并设定一个阈值来筛选角点。 $R$  为响应值, $\mathbf{M}$  为图像经过高斯加权的两个方向梯度的协方差矩阵。 $\det \mathbf{M}$  表示  $\mathbf{M}$  的行列式, $\text{trace} \mathbf{M}$  表示  $\mathbf{M}$  的迹。 $\alpha$  用以调整阈值,当  $R$  大于 0 时就认为是角点,反之则不是。

ORB 特征在尺度大小改变的情况下仍能较好匹配目标是通过搭建高斯金字塔完成的。每层金字塔都采用 FAST 算法采集角点,这样就在目标的各个尺度上都产生了角点,当目标尺度发生变化时就都可以匹配目标。

ORB 特征还具有方向不变性,即目标在视场中发生旋转它的特征点也依然能够匹配。为实现这一特

性,文献<sup>[9]</sup>采用灰度质心法。该方法假定特征点与其所做的邻域的灰度质心并不重合,那么就可以定义特征点到灰度质心的向量为该特征点的方向向量。假设  $K$  为某一个特征点, $K$  的相邻区块中的所有像素点的矩为

$$m_{pq} = \sum_{x,y} x^p y^q I(x,y) \quad (2)$$

其中  $m$  表示像素点的矩, $p, q$  表示矩的阶, $I(x,y)$  为点  $(x,y)$  处的灰度值。那么可以得到图像的质心为

$$C = \left( \frac{m_{10}}{m_{00}}, \frac{m_{01}}{m_{00}} \right) \quad (3)$$

于是可以求得从  $K$  指向质心的矢量的方向角为

$$\theta = \arctan(m_{01}, m_{10}) \quad (4)$$

BRIEF 描述符是一种运算非常快的特征点描述符,“Steer BRIEF”是 BRIEF 的一种改进,它很好地解决了 BRIEF 关于旋转不变性的描述问题。

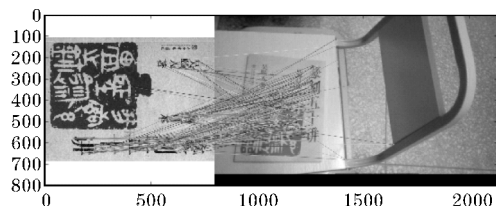


图1 利用 ORB 特征的检测结果

图1显示了在图片中采用 ORB 特征法检测书籍的结果,其中图的左侧为书籍模板。

## 2 融合 ORB 特征与 Mean Shift 的跟踪算法

### 2.1 颜色空间的选择

众多的颜色模型中,RGB 模型出现的频率最为频繁,但是缺点很明显,即在光照发生变化时分别代表着红色、绿色和蓝色的 RGB 色值都会跟着改变,这就导致色值相差很大的色彩看起来很接近。所以 RGB 模型在科学研究中不利于定量分析色彩。

HSV 颜色空间包括 3 个独立分量,它们是:色调(Hue)、对比度(Saturation)、亮度(Value)。在光照强度改变的情况下,Hue 分量几乎不会发生改变,所以用来构成 Mean Shift 算法中目标的颜色直方图对于光照的变化具有较强的鲁棒性。有时图像中一些特别亮或暗的点形成了图像分析中的噪声,这些噪声可以通过改变  $V$  或  $S$  分量来消除。于是在算法开始时,需要将 RGB 空间转换成 HSV 空间。其转换关系为

$$V = \max(R, G, B) \quad (5)$$

$$S = \begin{cases} \frac{V - \min(R, G, B)}{V}, & \text{if } V \neq 0 \\ 0, & \text{otherwise} \end{cases} \quad (6)$$

$$H = \begin{cases} \frac{60(G - B)}{V - \min(R, G, B)}, & \text{if } V = R \\ 120 + \frac{60(B - R)}{V - \min(R, G, B)}, & \text{if } V = G \\ 240 + \frac{60(R - G)}{V - \min(R, G, B)}, & \text{if } V = B \end{cases}$$

If  $H < 0$  then  $H = H + 360$

$$0 \leq V \leq 1, 0 \leq S \leq 1, 0 \leq H \leq 360 \quad (7)$$

式(5)、(6)、(7)中  $H$ 、 $S$ 、 $V$  分别代表像素点的色调分量、对比度分量、亮度分量的值,  $R$ 、 $G$ 、 $B$  代表像素点红色分量、绿色分量、蓝色分量的色值。

## 2.2 ORB 中目标尺度与特征点的方向计算

在目标运动过程中, ORB 特征关键点之间的相对位置只与尺度大小和旋转角度相关。因此可以通过比较模板中的目标与当前视场中目标所匹配上的特征点对, 得到当前视场中的目标的尺度大小与旋转角度, 继而调整跟踪窗口。

设在跟踪目标的模板中找出所有特征点构成的集合为

$$P_{\text{model}} = \{(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)\}$$

$P_{\text{cur}}$  为当前视场中找出的特征点集合,

$$P_{\text{cur}} = \{(x'_1, y'_1), (x'_2, y'_2), \dots, (x'_n, y'_n)\}$$

在两个集合中, 每个特征点都有一个方向, 设两个集合中特征点的主方向向量  $D_{\text{model}}$  和  $D_{\text{cur}}$  分别为:

$$D_{\text{model}} = (d_1, d_2, \dots, d_n),$$

$$D_{\text{cur}} = (d'_1, d'_2, \dots, d'_n),$$

特征点所在的尺度向量  $S_{\text{model}}$  和  $S_{\text{cur}}$  分别为:

$$S_{\text{model}} = (\sigma_1, \sigma_2, \dots, \sigma_n),$$

$$S_{\text{cur}} = \{\sigma'_1, \sigma'_2, \dots, \sigma'_n\},$$

则物体相对于模板在当前帧中的旋转角度  $\Delta\varphi$  由所有匹配上的特征点对的方向差分别取平均得到:

$$\Delta\varphi = \frac{1}{n} \sum_{i=1}^n (D_{\text{cur}} - D_{\text{model}}) \quad (8)$$

$\varphi_{\text{cur}}$  为目标主方向,  $\varphi_{\text{model}}$  为模板主方向, 则有:

$$\varphi_{\text{cur}} = \varphi_{\text{model}} + \Delta\varphi \quad (9)$$

同样地,  $\sigma_{\text{cur}}$  为当前帧的目标尺度,  $\sigma_{\text{model}}$  为模板中目标的尺度:

$$\sigma_{\text{cur}} = \sigma_{\text{model}} \times \frac{1}{n} \sum_{i=1}^n \frac{S_{\text{cur}}}{S_{\text{model}}} \quad (10)$$

## 2.3 跟踪窗口的确定

在目标识别与跟踪系统中, 跟踪窗口不需要完全

贴合目标的轮廓, 而往往采用一个矩形框表示。这样做简化计算, 也增强了跟踪的容错率。

在计算出当前视场中目标的大小和主方向后, 窗口的大小与方向也就确定了。

窗口的初始位置需要根据特征点的匹配情况确定, 只要找出在当前视场中矩形的中心再结合目标的大小就可以确定窗口的绘制。而当前视场中矩形中心的确定, 则需要参照在目标模板中矩形窗口的中心与特征点的位置关系。

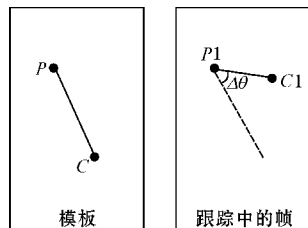


图2 确定跟踪窗口的位置

图2展示了点  $P$  和点  $C$  为模板中特征点与矩形窗口的中心点,  $P1$  为当前视场中与  $P$  相匹配的特征点,  $C1$  为未知的矩形中心点。可以看出在追踪过程中, 物体的方向和大小都产生改变。

由几何知识可知: 通过建立坐标系,  $\Delta\theta$  可表示为

$$\Delta\theta = \arctan \frac{y_C - y_P}{x_C - x_P} - \arctan \frac{y_{C1} - y_{P1}}{x_{C1} - x_{P1}} \quad (11)$$

$\Delta\theta$  相当于当前帧中的目标相对于模板的主方向  $\Delta\varphi$ ,  $\Delta\varphi$  可以由公式(8)计算出。式(11)、(12)中的  $x, y$  均分别为点  $P, C, P1, C1$  的横坐标与纵坐标。特征点  $P1$  到  $C1$  的距离与  $P$  到  $C$  的距离的比值可以近似地看作是模板与当前帧中目标尺度的比值, 即

$$\frac{\sigma_{\text{cur}}}{\sigma_{\text{model}}} = \frac{\sqrt{(x_{C1} - x_{P1})^2 + (y_{C1} - y_{P1})^2}}{\sqrt{(x_C - x_P)^2 + (y_C - y_P)^2}} \quad (12)$$

$\sigma_{\text{cur}}$  与  $\sigma_{\text{model}}$  的比值可由公式(10)求得, 联立式(11)、(12)即可求得  $C1$  的坐标  $(x_{C1}, y_{C1})$ 。

确定矩形中心后, 结合当前帧目标尺度就能绘制出跟踪窗口了。

## 2.4 融合 ORB 特征与 Mean Shift 的跟踪算法

在嵌入式领域中, 运算的实时性通常是最为关键的指标之一, 一些传统的视频跟踪算法往往难以满足这一需求。为了解决该问题, 决定采用运算速度很快的 ORB 特征结合 Mean Shift 实施追踪。这样也同时修正了由于目标与背景色彩相似导致的 Mean Shift 算法失效问题。

算法的主要流程如图3所示。



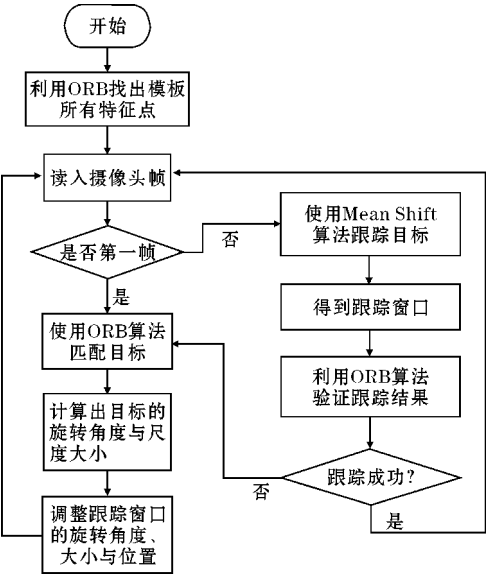


图3 融合 ORB 特征与 Mean Shift 的流程图

算法步骤:

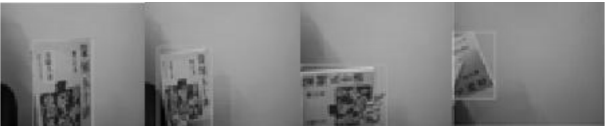
- 步骤 1 利用摄像头捕获视频帧,判断当前帧是否为第一帧。如果是第一帧,则利用 ORB 特征算法匹配目标,并将匹配上的目标作为 Mean Shift 跟踪算法的新的模板,并读取下一帧。如果不是,则执行步骤 2。
- 步骤 2 运用 Mean Shift 实施追踪,在当前视场中绘制出标示物体的窗口。同时利用 ORB 特征检测跟踪的效果。
- 步骤 3 为 ORB 特征点检测设置阈值,当特征点的个数小于阈值时认为对目标跟踪失败了,此时执行步骤 4。若特征点数大于所设定的阈值则认为目标没有丢失,此时读取下一帧继续跟踪。
- 步骤 4 根据 ORB 特征算法重新寻找目标,利用特征点分布确定目标位置,并计算出目标的旋转角度与尺度大小调整跟踪窗口。
- 步骤 5 在新的追踪窗口上继续在下一帧使用 Mean Shift。
- 步骤 6 重复以上步骤,持续跟踪下去。

3 实验结果与分析

为了验证算法的正确性与有效性,设计了直接利用摄像头捕获视频流进行目标跟踪的实验,将本算法的结果与采用 Mean Shift 以及采用基于 SIFT 的 Mean Shift 改进算法的结果相对比。实验的硬件采用 Broadcom BCM2837 64Bit Quad Core Processor 1.2 GHz CPU, 1 GB 400 MHz SDRAM, 软件环境为 python 2.7 和 OpenCV 3.0.0 运行库。测试用摄像设备的分辨率为 640×480 像素。



(a)传统 Mean Shift 算法跟踪效果



(b)融合了 SIFT 的 Mean Shift 改进算法跟踪效果



(c)文中算法的跟踪效果

图4 三种算法的跟踪效果对比

为验证算法,在室内拍摄了三段视频,分别采用传统的 Mean Shift 算法,结合 SIFT 改进的 Mean Shift 算法,以及本文介绍的算法,三段视频中的目标相同,均为书本。图 4 为三段视频的图像截取。图 4 展示了当使用传统的 Mean Shift 时,窗口的大小一直保持不变,这说明它不能反映目标的尺度变化,当目标由近处移动至远处(或相反)时会导致结果不精确。并且,在目标受到遮挡的情况下出现跟踪失败的结果。另外,采用了融合 SIFT 特征的 Mean Shift 改进算法以及本文所介绍的跟踪算法的跟踪效果更加准确,并对于目标的尺度变化和旋转都具有一定的鲁棒性,在物体未被完全遮蔽的情况下仍然能做到出色的识别。

要将对于目标的视频跟踪应用于嵌入式平台,由于嵌入式平台计算性能有限,对于图像的处理速度慢,所以跟踪的实时性和准确性都是需要考虑的因素。从实验结果的直观感受上来说,Mean Shift 运行起来最为流畅,处理速度最快,但目标未被完全遮蔽时会导致失误;融合 SIFT 特征的 Mean Shift 改进算法虽然跟踪的准确率很高,但是跟踪视频会有很明显的卡顿,出现明显的丢帧现象;而本文所介绍的算法在处理速度上明显优于结合 SIFT 特征的 Mean Shift 改进算法,实验效果也非常出色。

表 1 不同算法实时性与特征点匹配情况对比			
算法	平均单帧处理用时/s	总特征点个数	无筛选特征点配对个数
Mean Shift	0.004	—	—
SIFT	2.083	717	367
ORB	0.119	500	273

表 1 展示了上述 3 种算法单帧处理时间对比以及对于特征点的匹配情况对比。从中可以看出本文方法的实时性要远好于融合 SIFT 特征的 Mean Shift 改进算法。

## 4 结束语

为了解决嵌入式平台下的视频跟踪实时性和正确性问题,介绍了结合 ORB 特征的 Mean Shift 改进算法,不仅能利用特征点检测目标,还可以对 Mean Shift 的追踪结果进行检查和修正。与以前的方法比较,本文介绍的算法更为快速和稳定,本文设计的实验部分验证了这一点。本文介绍的算法不能实现自适应地对运动物体跟踪,离不开对目标模板的检测,还需要进一步的研究和改进。另外,受限于硬件运算速度和算法的效率,本文仅考虑了对速度较慢的目标,在对高速物体进行跟踪时效果不理想。

## 参考文献:

- [1] FUKUNAGA, K, HOSTETLER L. The estimation of the gradient of a density function, with applications in pattern recognition [J]. IEEE Transactions on Information Theory, 1975, 21(1): 32-40.
- [2] Comaniciu D, RAMESH V, MEER P. Real-time tracking of nonrigid object using mean shift [C]. Proc of IEEE Conf on Computer Vision and Pattern Recognition. Hilton Head Island: [s. n.], 2000: 142-149.
- [3] Comaniciu D, MEER P. Mean shift analysis and applications [C]. The Proceedings of the Seventh IEEE International Conference on Computer Vision. Kerkyra Greece 1999.
- [4] CHENG Yi zong. Mean shift mode seeking and clustering [J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 1995, 17(8).
- [5] LOWE D G. Distinctive image features from scale-invariant keypoints [J]. International Journal of Computer Vision, 2004, 60(2): 91-110.
- [6] Ke Y, Sukthankar R. PCA-SIFT: A more distinctive representation for local image descriptors Computer Vision and Pattern Recognition [C]. Proceedings of the 2004 IEEE Computer Society Conference on. [s. l.]: IEEE, 2004: 506-513.
- [7] CHEN A H, ZHU M, WANG Y H. Mean shift tracking combining SIFT [C]. ICSP2008 Proceedings. Leipzig, Germany, Springer LNCS, 2008, 1532-1535.
- [8] Dong R, Li B, Chen Q M. Multi-degree-of-freedom mean-shift tracking algorithm based on SIFT feature [J]. Control and Decision, 2012, 27(3).
- [9] 王智军, 王建华. 基于 SIFT 验证的 Mean Shift 跟踪运动目标新算法 [J]. 电光与控制, 2016, 23(11): 93-96.
- [10] DI N, LI G J, CHEN CH N. Image matching technology research based on normalized DOG features [J]. Journal of Electronic Measurement and Instrumentation, 2014, 28(6): 585-560.
- [11] YANG J H, HANG Y Q, XU J F, et al. Quick realization of CUDA-based registration of high-resolution digital video images [J]. Chinese Journal of Scientific Instrument, 2014, 35(2): 380-386.
- [12] WEI X, FAN SH SH, GAO SH. Rapid matching and transform algorithm of images based on the BRISK [J]. Electronic Measurement Technology, 2015, 38(3): 43-47.
- [13] LU M M, ZHAO F J, LI N. Feature extraction of high-resolution SAR images based on bag-of-word model [J]. Foreign Electronic Measurement Technology, 2015, 34(6): 62-69.
- [14] LIU S S, ZHANG H, MAO ZH, et al. Target detection method based on HRM feature extraction and SVM [J]. Foreign Electronic Measurement Technology, 2014, 33(10): 38-41.
- [15] Rublee E, Rabaud V, Konolige K. ORB: an efficient alternative to SIFT or SURF [M]. ICCV, 2011: 2564-2571.
- [16] 钟华民, 王伟, 张慧华, 等. 结合 ORB 特征和色彩模型的视觉跟踪算法 [J]. 模式识别与人工智能, 2015, 28(1): 91-96.

## A Visual Tracking Algorithm Combining ORB Feature and Mean Shift

YAN Wen-long, ZHU Yi, LU Jun

(College of Software, Chengdu University of Information Technology, Chengdu 610225, China)

**Abstract:** As the real-time requirements of the embedded system for computing are critical, some of the traditional vision tracking algorithm can hardly meet them. In order to solve this problem, this paper proposes a computer vision tracking algorithm combining ORB (oriented fast and rotated brief) feature points and Mean Shift algorithm. The traditional Mean Shift algorithm runs faster, but it is easy to fail while the target is blocked, which results in the tracking result and it is not satisfactory. Therefore, it is decided to use the Mean Shift algorithm which combines the ORB feature detection algorithm to achieve the target tracking. The algorithm improves the accuracy of the search window by detecting the initial position of the target and calculating the scale and rotation angle of the target in the current frame according to the feature points matched with the template. Compared with the traditional algorithm, the algorithm proposed in this paper shows a good robustness in solving the problem of obscuring the target, and it has a good performance in real-time tracking.

**Keywords:** visual tracking; Mean Shift; ORB feature; embedded platform; real-time