

基于通道注意力和特征切片的图像快速匹配算法

盖绍彦^{1,2*}, 黄妍妍^{1,2}, 达飞鹏^{1,2**}¹东南大学自动化学院, 江苏 南京 210096;²东南大学复杂工程系统测量与控制教育部重点实验室, 江苏 南京 210096

摘要 针对现有图像匹配算法追求匹配精度,而一定程度上忽略匹配速度下降的问题,设计了通道注意力和特征切片描述网络(CAFSD),并将其与快速特征点检测算法(ORB)结合,进而提出一种精度和速度均衡的图像快速匹配算法。首先利用通道注意力对ORB获取的图像块进行初步全局特征学习,接着将初步特征描述子切片,再对每个特征切片进行特征学习和二进制化,最后重新拼接得到二进制特征描述子,减少了传统二进制编码的冗余问题。此外,设计了复合损失函数进一步减少描述子二进制化造成的信息损失。实验结果表明,所提算法的匹配性能和处理速度优于大部分其他二进制匹配算法。

关键词 图像处理; 图像匹配; 注意力; 特征描述

中图分类号 TP391.4 **文献标志码** A

DOI: 10.3788/AOS231054

1 引言

图像匹配是对多张图像寻找相同或相似目标物体的空间对准关系的过程^[1]。图像匹配是光学测量领域的研究热点之一,广泛应用于图像拼接、产品光学测量、视频防抖、三维重建等领域。在实际场景中,不仅要保证图像匹配技术具有较高的匹配准确率,还对匹配技术的实时性有一定的要求,所以加快图像匹配的速度,实现图像快速匹配至关重要。在当前的研究工作中,加快图像匹配的速度主要从加快对特征点的检测速度和加快对特征的匹配速度这两方面出发。

加快特征点检测的方法中,最具有代表性的是由Rosten等^[2]、Calonder等^[3]、Rublee等^[4]提出的 oriented fast and rotated brief(ORB)算子,通过对比不同点之间的像素差来检测特征点,运算过程较为简单,处理速度快。除此之外,一些算法通过筛选特征点来提高对图像的匹配速度。刘自金等^[5]将原图像的各个区域划分为高信息熵区域和低信息熵区域,并直接在高信息熵区域上检测特征点,从而提高图像匹配的速度。Chu等^[6]在卷积神经网络中通过引入注意力模块来筛选特征点。陈世伟等^[7]在合成的人工光学图像上进行特征点检测,加快了匹配算法对特征点的检测速度。于子雯等^[8]在尺度空间设置权重系数并结合响应强度图来筛选均匀分布且稳定的特征点。齐向明等^[9]设计了一种子窗口注意力的方法来保护图像尺度空间的边缘信

息和角点信息。但是,筛选特征点的过程本身就占用一定的时间,并且特征点蕴含着丰富的特征信息,如果直接筛选掉一些特征点可能会增加图像匹配的误差。

一些算法通过加快对特征匹配的速度来减少图像匹配时间,将特征描述算法输出的浮点型特征描述子转换成二进制型特征描述子来减少特征匹配时的运算量,从而减少特征匹配时间。Ye等^[10]设计了 compact discriminative binary(CDbin)描述子,以较少的训练参数量获得二进制特征描述子。Fan等^[11]对特征描述网络 HardNet^[12]的训练数据中的正样本进行特征增强,并在输出层使用 sign 函数来进行二进制化。Yan等^[13]提出了一种二进制多视角抗噪特征描述网络。Gu等^[14]对深度卷积神经网络 AlexNet^[15]结构进行修改,将输出的描述子元素值映射为-1或1。Yang等^[16]使用多位二进制描述符对图像块进行描述,减少了直接将实值的浮点型描述符转化为二进制描述符造成的信息丢失。Soleimani等^[17]提出了一种循环移位二进制描述符,减少了用于计算描述符的参数数量,从而提高了匹配速度。

上述研究工作通常只考虑在算法的输出层做一些二进制化的操作,对特征描述子进行二值映射,但较少考虑到特征描述子中不同位置元素之间的相关性,若该相关性太高,在二进制编码时会造成编码冗余,从而影响整个图像快速匹配算法的准确率。特征表示深度神经网络 SFLHC^[18]对每一个元素单独使用 Sigmoid

收稿日期: 2023-05-26; 修回日期: 2023-07-19; 录用日期: 2023-09-06; 网络首发日期: 2023-09-22

基金项目: 江苏省前沿引领技术基础研究专项(BK20192004C)、江苏省高校优势学科建设工程资助课题

通信作者: *qxxyymm@163.com; **dafp@seu.edu.cn

函数和分段阈值函数并进行二进制化操作,本文受此启发,并结合通道注意力的思想^[19],对二进制特征描述网络 CDbin^[10]进行改进,设计出通道注意力和特征切片描述网络(CAFSD),并将其与快速特征点检测算法 ORB 结合,进而提出一种基于通道注意力机制和特征切片的图像快速匹配算法,该算法在提高二进制描述精度的同时可大幅度提高图像的匹配速度。此外,在三元组损失函数的基础上,引入量化损失函数、均匀分布损失函数和相关性损失函数以形成复合损失函数优化网络训练,进一步减小浮点型描述子转换成二进制描述子造成的误差。

2 基于通道注意力和特征切片的快速图像匹配算法设计

所提基于通道注意力和特征切片的快速图像匹配算法的处理流程如图 1 所示,自左向右分别是第一部分特征检测与图像裁剪、第二部分特征描述和第三部分特征匹配。在第一部分,主要使用 ORB 算子

来检测输入图像的特征点,并以特征点为中心将原图像旋转与特征点相同的角度,裁剪 64×64 的图像块并降采样为 32×32 尺寸图像。再将 32×32 尺寸图像输入第二部分进行特征描述,特征描述的工作由 CAFSD 完成,该网络由全局特征描述模块(GFDM)和切片与编码模块(SEM)组成。输入第二部分的图像块先进入 GFDM 进行全局初步特征学习得到初步特征描述子,接着将初步特征描述子输入 SEM,得到多个特征切片,并将它们分别输入到对应的卷积层对局部信息进行进一步学习,使用符号函数(sign 函数)将卷积层的输出值映射为 -1 或 1 ,将映射之后的特征切片重新拼接为二进制特征描述子。将两张输入图像所对应的二进制特征描述子输入第三部分进行特征匹配,计算两张图像中所有最近特征描述子对之间的 Hamming 距离,依据设定的 Hamming 距离阈值筛选这些最近特征描述子对,得到两张图像中匹配的特征描述子对,最后可得到两张图像的匹配结果。

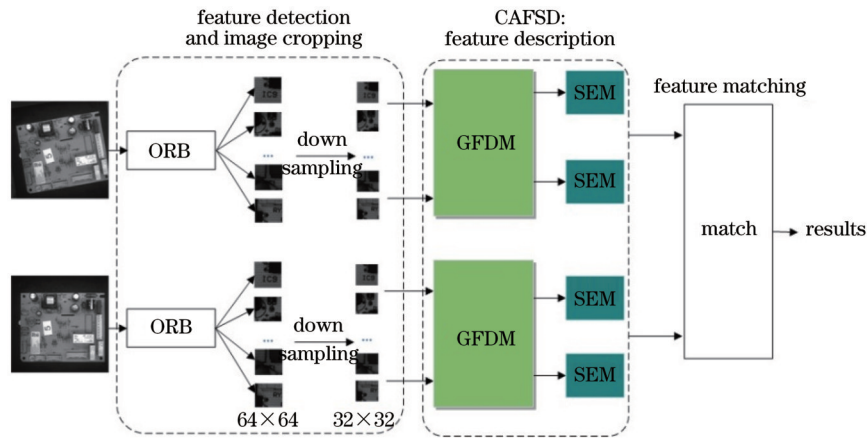


图 1 基于通道注意力和特征切片的快速图像匹配算法的处理流程

Fig. 1 Processing flow of fast image matching algorithm based on channel attention and feature slicing

2.1 GFDM

GFDM 是 CAFSD 的第一部分,也是图 1 第一部分输出的 32×32 尺寸的图像块输入 CAFSD 之后首先进入的模块。GFDM 的结构如图 2 所示,主要由对二进制特征描述网络 CDbin^[10]进行简化操作得到的网络结构(SCDbin)和挤压激励分支(SEB)构成。CDbin 网络的输入图像尺寸、卷积核尺寸和 padding 填充值均偏大,这些因素导致了 CDbin 网络具有参数量偏多、训练时间过长、过拟合等问题,所以对 CDbin 进行了合理简化,使其在网络复杂度和训练时间下降的同时能保证一定的性能。

CDbin 简化之后的结构如图 2 的 SCDbin 部分所示。为了减少参数量,将卷积层减少为 3 层,第一层卷积层的移动步长为 1,剩下两层的移动步长为 2,每层均使用批归一化(BN)和 ReLU 激活函数来增加参数的非线性。假设输入 SCDbin 的图像块 x , x 经过第一

个卷积层后的输出可以表示为

$$x' = \text{ReLU} \left\{ \text{BN} \left[\text{Conv2d}(x) \right] \right\} = \text{RBC}(x), \quad (1)$$

式中:RBC 表示式(1)的计算过程;Conv2d 为二维特征卷积操作,其卷积核尺寸为 3×3 ,padding 填充值为 1。 x' 经过第二层卷积层后的输出可以表示为

$$x'' = \text{ReLU} \left\{ \text{BN} \left[\text{Conv2d}(x') \right] \right\} = \text{RBC}(x'). \quad (2)$$

SEB 是基于压缩与激励网络(SENNet)^[19]设计的。SENNet 对输入的特征图进行挤压激励操作,得到通道信息的重要程度,并对其与原输入的特征图进行张量点乘运算,这样的操作会增加原神经网络的深度,可能会导致网络退化,从而影响整个神经网络的性能。所以,所提算法将挤压激励操作作为一个分支,从侧面插入到 SCDbin 中,如图 2 的 SEB 部分所示。在 SCDbin 从空间角度对输入的特征图进行更深层学习的同时,SEB 分支从通道的角度学习输入特征图不同通道信

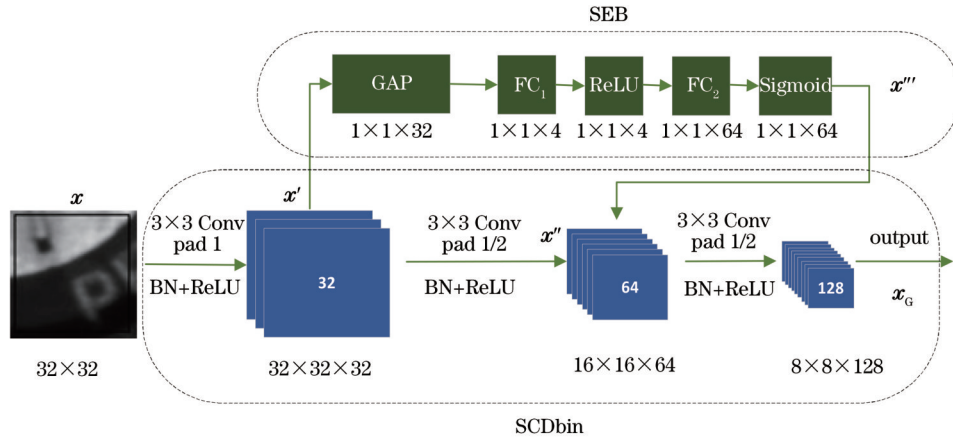


图 2 GFDM 的结构
Fig. 2 Structure of GFDM

息的重要程度,双分支的网络结构提高了GFDM的训练效率。根据图2可得,SEB的输入为 x' ,则SEB的输出可以表示为

$$x''' = \text{Sigmoid} \left\{ \text{FC}_2 \left\{ \text{ReLU} \left\{ \text{FC}_1 \left[\text{GAP} (x') \right] \right\} \right\} \right\} = \text{SEB} (x'), \quad (3)$$

式中:GAP(\cdot)为全局平均池化运算,计算 x' 各通道参数的平均值,并将 x' 的维度由 $32 \times 32 \times 32$ 缩减到 $1 \times 1 \times 32$;FC₁和FC₂分别表示SEB的第一层和第二层全连接层。最后信号经过Sigmoid激活函数,将FC₂输出特征图的参数“挤压”到(0,1)区间,Sigmoid输出的特征图不同通道的参数值表示不同通道信息的重要程度。SEB的输出 x''' 与SCDbin第二层卷积层的输出 x'' 进行张量的点乘运算,强化64个通道中相对重要信息的学习。结合式(1)~(3)和图2,GFDM的输出可以表示为

$$x_G = x'' \cdot x''' = \text{ReLU} \left\{ \text{BN} \left[\text{Conv2d} (x') \right] \right\} \cdot \text{SEB} (x') = \text{RBC} \left[\text{RBC} (x) \right] \cdot \text{SEB} \left[\text{RBC} (x) \right], \quad (4)$$

如上,GFDM得到初步特征描述子 x_G ,其维度为 $8 \times$

8×128 。

2.2 SEM

CDbin网络^[10]得到输入图像块的浮点型特征描述子后,直接使用sign函数将整个浮点型描述子映射为二进制描述子,这样的编码方式可以抽象为图3右图的直接编码方式。图3中,transform表示对输入特征图所作的一系列计算操作,sign则是符号函数,将transform变换输出的浮点型描述子转为二进制描述子。一般情况下,获取二进制特性描述子时,希望特征描述子中不同位置的元素之间的相关性尽可能小,这样可以减小编码冗余的程度,提高匹配准确率。减小相关性的思路如图3左图的切片编码方式所示,将输入的特征分割为多个特征切片,再针对每一个特征切片进行transform和sign的操作,最后将二进制特征切片拼接为二进制特征描述子。

基于上述切片的思路,设计了SEM,对初步特征描述子进行进一步学习和编码。如图4所示,SEM结构涉及3个步骤,即切片(slice)、编码(encode)和拼接(concat)。具体处理过程如下。

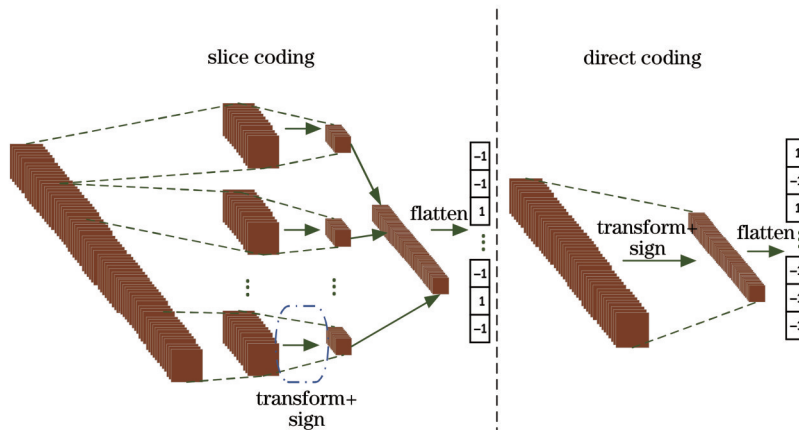


图 3 切片编码方式(左)与直接编码方式(右)
Fig. 3 Slice coding (left) and direct coding (right)

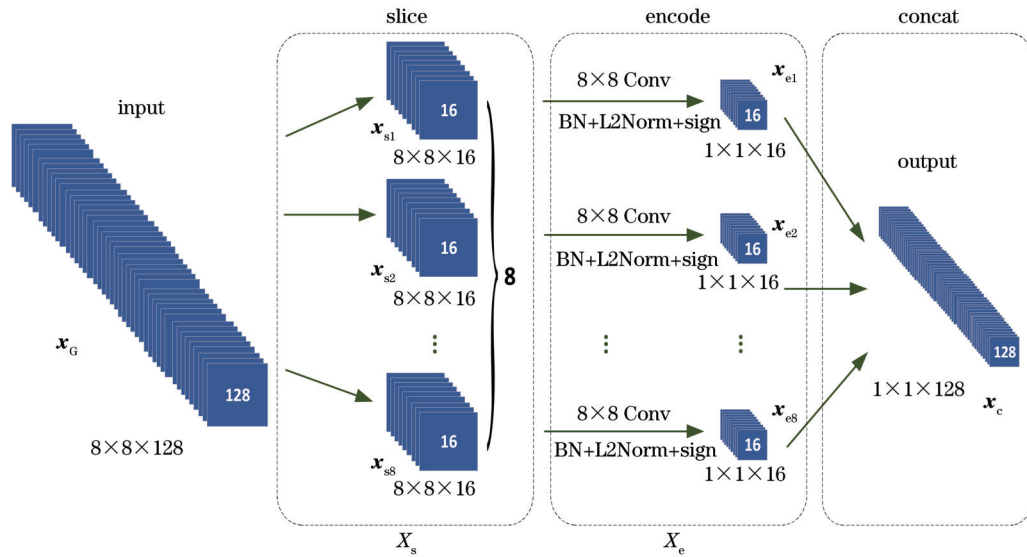


图 4 SEM 的结构

Fig. 4 Structure of SEM

由式(4)得到的初步特征描述子 x_G 维度是 $8 \times 8 \times 128$ 。在进行 slice 处理操作时,在通道维度以 16 为切片长度,将 x_G 分割为 8 个 $8 \times 8 \times 16$ 维度的特征切片,以 X_s 表示进行 slice 操作之后的特征切片的集合,表示为

$$X_s = \text{slice}(x_G) = \{x_{s_i} | 1 \leq i \leq 8\} = \{x_{s_1}, x_{s_2}, \dots, x_{s_8}\}, \quad (5)$$

式中: x_{s_i} 表示 X_s 中第 i 个特征切片。在 encode 步骤中,将每一个特征切片输入到一个卷积层,如图 4 的 encode 部分所示,最后使用 sign 函数将特征切片的元素值映射为 -1 或 1 。使用 x_{e_i} 表示特征切片 x_{s_i} 经过 encode 之后的输出,表达式为

$$x_{e_i} = \text{sign} \left\{ \text{L2Norm} \left\{ \text{BN} \left[\text{Conv2d}(x_{s_i}) \right] \right\} \right\} = \text{SLBC}(x_{s_i}), \quad (6)$$

式中: L2Norm 表示 L2 规范化; SLBC 表示式(6)的运算过程。用 X_e 表示所有特征切片经过 encode 操作之后的二进制特征切片集合,表达式为

$$X_e = \text{encode}(X_s) = \{x_{e_i} | 1 \leq i \leq 8\} = \{x_{e_1}, x_{e_2}, \dots, x_{e_8}\}, \quad (7)$$

式中: x_{e_i} 是 X_e 中第 i 个二进制特征切片, $1 \leq i \leq 8$ 。在 concat 步骤,将所有的二进制特征切片拼接为 $1 \times 1 \times 128$ 的二进制特征描述子。使用 x_c 表示 concat 操作的输出,表达式为

$$x_c = \text{concat}(X_e). \quad (8)$$

综上,整个 CAFSD 的输出可以表示为

$$x_c = \text{concat}(X_e) = \text{concat}[\text{encode}(X_s)] = \text{concat}\{\text{encode}[\text{slice}(x_G)]\} = \text{CES}(x_G). \quad (9)$$

2.3 损失函数

为了进一步减小算法中浮点型描述子二进制化造成的信息损失,在三元组损失函数 $L_T^{[11]}$ 的基础上,引

入量化损失函数 L_Q 、均匀分布损失函数 L_E 和相关性损失函数 L_C , 将它们组成复合损失函数 L_{all} 来优化 CAFSD 的训练过程。

1) 量化损失函数 L_Q

SEB 使用 sign 函数将浮点型元素值映射为 -1 或 1 , 这样会造成一定程度的信息损失。为了最大程度地减小信息损失,引入量化损失函数 L_Q 。假设 CAFSD 的 batch size 为 N , 用维度为 l 的二进制描述子 $B_n = [b_{n,1}, b_{n,2}, \dots, b_{n,l}]$, $n = \{1, 2, \dots, N\}$ 表示任意一个图像块输入 CAFSD 之后的输出 x_c , 使用维度为 l 的浮点型描述子 $F_n = [f_{n,1}, f_{n,2}, \dots, f_{n,l}]$, $n = \{1, 2, \dots, N\}$ 表示 x_c 进行二进制化前的浮点型特征描述子。 L_Q 的表达式为

$$L_Q = \frac{1}{l \times N} \sum_{n=1}^N \sum_{j=1}^l (f_{n,j} - b_{n,j})^2, \quad (10)$$

式中: L_Q 为 B_n 与 F_n 的对应元素之差的平方和的平均值。CAFSD 往最小化 L_Q 的方向进行训练,使 F_n 中每一个元素值尽可能接近 -1 或 1 。

2) 均匀分布损失函数 L_E

使用大型数据集训练二进制特征描述网络,若网络输出的不同描述子在相同位置的元素值的总和接近 0, 则说明输出的特征描述子均匀分布,依此来设计 L_E , 表达式为

$$L_E = \frac{1}{2l} \sum_{j=1}^l \left(\frac{\sum_{n=1}^N f_{n,j}}{N} \right)^2, \quad (11)$$

L_E 就是对不同浮点型描述子在相同位置的元素值的平均值的平方和求平均,再除以 2 得到的。加入 L_E , 可以使 CAFSD 输出的特征描述子往均匀分布的方向得到训练。

3) 相关性损失函数 L_C

为了进一步减小二进制描述子不同位置元素之间的相关性,避免编码冗余,引入相关性损失函数 L_C 。假设一个 batch size 内网络输出 N 个 l 维的二进制描述子,对应存在 N 个 l 维的浮点型描述子,将其组合成 $N \times l$ 的矩阵 $F_{N \times l} = [y_{c1}, y_{c2}, \dots, y_{cl}]$,则相关性矩阵 $R_{l \times l} = [r_{p,q}]_{l \times l}$ 的元素计算公式为

$$r_{p,q} = \frac{(y_{cp} - \overline{y_{cp}})^T (y_{cq} - \overline{y_{cq}})}{\sqrt{(y_{cp} - \overline{y_{cp}})^T (y_{cp} - \overline{y_{cp}})} \sqrt{(y_{cq} - \overline{y_{cq}})^T (y_{cq} - \overline{y_{cq}})}}, \quad (12)$$

式中: $\overline{y_{cp}}$ 和 $\overline{y_{cq}}$ 表示 $F_{N \times l}$ 中第 p 个列向量 y_{cp} 和第 q 个列向量 y_{cq} 的平均值。 L_C 的表达式为

$$L_C = \frac{1}{2(l^2 - l)} \sum_{p \neq q} r_{p,q}^2, \quad (13)$$

L_C 的值即为 $R_{l \times l}$ 非对角线元素平方和的平均值,再除以 2。综上,构建 CAFSD 的复合损失函数 L_{all} , 表达式为

$$L_{all} = L_T + \alpha L_Q + \beta L_E + \gamma L_C, \quad (14)$$

式中: α 、 β 和 γ 均是权重系数。

2.4 CAFSD 的整体结构

结合图 2 和图 4, 整个 CAFSD 的结构可表示为图 5, 输入的图像块 x 经过 GFDM 和 SEM 得到二进制特征描述子 x_c , 公式为

$$x_c = \text{CES}(x_G) = \text{CES}(x'' \cdot x''') = \text{CES}\{\text{RBC}[\text{RBC}(x)] \cdot \text{SEB}[\text{RBC}(x)]\}. \quad (15)$$

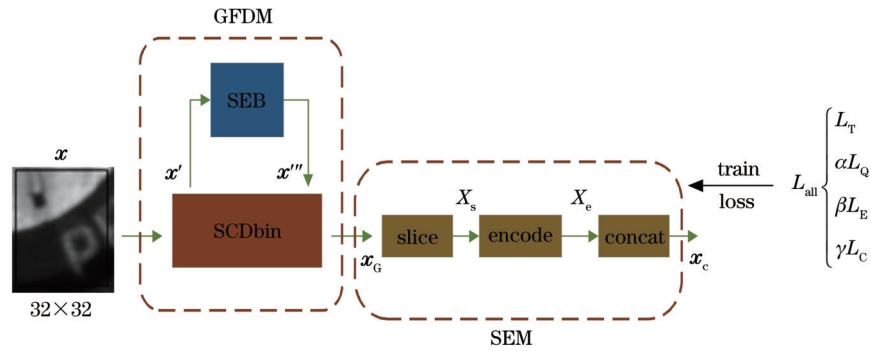


图 5 CAFSD 的结构

Fig. 5 Structure of CAFSD

在进行前向传播的同时,计算一个 batch size 的 L_{all} 并回传,使 CAFSD 往最小化 L_{all} 的方向训练。

3 实验结果与分析

3.1 实验数据及参数设置

所提 CAFSD 使用 UBC Phototour 数据集进行训练和测试^[20]。UBC Phototour 数据集包括 Liberty、Notre Dame 和 Yosemite 3 个子数据集,通常用其中一个数据集对网络进行训练,另两个对网络进行测试^[20],取平均值作为最终结果。此外,对于整个图像匹配算法(图 1)的测试,使用 20 张电路板数据,其中包括对模板图像进行旋转、缩小等空间变换的图像。

实验所使用的 GPU 为 NVIDIA GeForce GTX 1060,训练框架为 PyTorch1.9.0, batch size 设置为 1000,优化器选择 SGD,初始学习率为 0.1。

3.2 评价指标

采用常用的评价指标评价算法的性能,包括 FPR95^[20]、匹配精度、匹配分数和匹配时间^[9]。其中 FPR95 是 UBC Phototour 数据集所用的衡量特征描述算法质量的评价指标。其他指标用于对图像匹配算法进行整体测试,其中匹配时间是指从特征检测时开始到特征匹配结束算法所花费的时间。

3.3 损失函数权重系数验证实验

对 L_{all} 中的权重系数进行探索。分别使用 L_T 与 L_Q 、 L_T 与 L_E 、 L_T 与 L_C 这三种组合的损失函数参与 CAFSD 训练,使用 UBC Phototour 数据集进行测试,结果如表 1 所示。第一列为 L_T 与另一损失函数的权重系数之比,第 2~4 列是另一损失函数分别为 L_Q 、 L_E 、 L_C 时的 FPR95 结果。由第 2~4 列的最优结果,可得 L_Q 、 L_E 和 L_C 最优的系数为 1、0.5、0.5。

表 1 在 UBC Phototour 数据集上不同权重系数下的 FPR95
Table 1 FPR95 under different coefficients on UBC Phototour dataset unit: %

Weight	L_Q	L_E	L_C
1:1	2.42	2.66	2.95
1:0.5	2.50	2.55	2.89
1:0.1	2.58	2.63	3.01

使用 L_{all} 训练 CAFSD 时,将其中的 L_Q 、 L_E 和 L_C 分别乘以最优权重系数进行训练,与修改系数前的 1:1:1 进行对比,结果如表 2 所示, FPR95 比修改前的降低 2.57%。

使用电路板数据集对修改权重系数前后的性能指标进行对比,如表 3 所示,修改后的匹配精度和匹配分

表2 在 UBC Phototour 数据集上修改权重系数前后的 FPR95
Table 2 FPR95 before and after modifying weight coefficient on UBC Phototour dataset

$1:\alpha:\beta:\gamma$	FPR95 / %
1:1:1:1	2.72
1:1:0.5:0.5	2.65

数均明显得到提升。匹配时间差异不大,原因是权重系数未对算法的复杂度产生较大影响。

表3 修改权重系数前后在电路板数据集的结果

Table 3 Results before and after modifying weight coefficient on circuit board dataset

$1:\alpha:\beta:\gamma$	Matching accuracy	Score	Time /s
1:1:1:1	0.9113	0.6161	2.57
1:1:0.5:0.5	0.9396	0.6230	2.58

根据表1~3,可将式(14)中的 α 、 β 和 γ 分别确定为1、0.5和0.5。

3.4 消融实验

为了验证 GFDM、SEM 和 L_{all} 对算法性能提升的有效性,与改进前的 CDbin 网络^[7]做消融实验。在 UBC Phototour 数据集下的结果如表4所示,其中 both+ L_T 、both+ L_{all} 分别表示同时使用 GFDM 和 SEM 并使用 L_T 、 L_{all} 的两种情况。与改进前的 CDbin 相比, FPR95 下降,特别是 both+ L_{all} 情况下的 FPR95 是最低的。特别要说明的是,单独使用 GFDM 比单独使用 SEM 的结果好,这是因为 GFDM 提供了优质的初步特征描述子。

表4 应用不同模块在 UBC Phototour 数据集上的 FPR95

Table 4 FPR95 of different modules on UBC Phototour dataset

Module	FPR95 / %
CDbin ^[7]	3.89
GFDM	2.78
SEM	2.89
both+ L_T	2.76
both+ L_{all}	2.65

电路板数据集上的测试结果如表5所示, both+ L_{all} 的匹配精度和匹配分数是最好的,匹配精度比改进前的 CDbin 高 4.98%。both+ L_{all} 的匹配时间稍有增加,可能是因为存在循环操作,但相比 CDbin 也仅高 0.407 s,在可接受的范围内。

3.5 对比实验

首先,在 UBC Phototour 数据集上对 CAFSD 和其他二进制特征描述网络进行对比,结果如表6所示,可以看出 CAFSD 的 FPR95 优于其他经典算法。

将算法应用到整体的图像匹配,在电路板数据集上的结果如表7所示。CAFSD 的匹配精度和匹配分数排名第一,而匹配时间也低于大部分算法,排名第三。

表5 应用不同模块在电路板数据集上的结果

Table 5 Results of different modules on circuit board dataset

Module	Matching accuracy	Score	Time /s
CDbin ^[7]	0.8950	0.6097	2.173
GFDM	0.9085	0.6155	2.12
SEM	0.9080	0.6127	2.39
both+ L_T	0.9101	0.6149	2.58
both+ L_{all}	0.9396	0.6230	2.58

表6 各算法在 UBC Phototour 数据集上的 FPR95

Table 6 FPR95 of different algorithms on UBC Phototour dataset

Algorithm	FPR95 / %
ORB ^[4]	57.54
CDbin ^[7]	3.89
BLCD ^[12]	8.52
DBLD ^[15]	2.69
DCBMQ ^[21]	31.51
CFBD ^[17]	2.97
LDBD ^[22]	19.83
CAFSD	2.65

表7 各算法在电路板数据集的结果

Table 7 Results of different algorithms on circuit board dataset

Algorithm	Matching accuracy	Score	Time/s
ORB ^[4]	0.8160	0.6051	0.226
CDbin ^[7]	0.8950	0.6097	2.173
BLCD ^[12]	0.9018	0.6130	2.67
DBLD ^[15]	0.9215	0.6011	3.89
DCBMQ ^[21]	0.8520	0.6013	2.60
CFBD ^[17]	0.9173	0.6163	2.81
LDBD ^[22]	0.8991	0.6107	3.68
CAFSD	0.9396	0.6230	2.58

为了进一步验证所提算法的泛化性,分别在公共数据集 DAISY 和 RetinaMnist 进行测试^[23]。在 DAISY 数据集随机抽取 30% 的图像作为模板图像,生成对应的旋转、缩小等图像,将变换后的图像与模板图像作为一组输入图像对进行算法测试,结果如表8所示,CAFSD 的匹配精度和分数是最优的,匹配时间也是可以接受的。

在 RetinaMnist 数据集上的结果如表9所示, CAFSD 优于其他匹配算法。其中在匹配时间上, ORB 算法属于传统算法,所以时间上远远少于其他算法。测试结果证明了所提 GFDM、SEM 和 L_{all} 对算法性能提升的有效性。

3.6 可视化展示

使用改进前的 CDbin 网络替代图1的 CAFSD 部分,与所提算法在同一组输入图像上进行可视化展示,

表 8 DAISY 数据集上的结果

Table 8 Results on DAISY dataset

Algorithm	Matching accuracy	Score	Time /s
ORB ^[4]	0.8310	0.5146	0.115
CDbin ^[7]	0.8874	0.6081	1.258
BLCD ^[12]	0.9071	0.5960	1.498
DBLD ^[15]	0.9184	0.6081	2.096
DCBMQ ^[21]	0.8962	0.5973	1.870
CFBD ^[17]	0.9050	0.6099	1.711
LDBD ^[22]	0.8873	0.6001	2.582
CAFSD	0.9199	0.6129	1.880

表 9 在 RetinaMnist 数据集上的结果

Table 9 Results on RetinaMnist dataset

Algorithm	Matching accuracy	Score	Time /s
ORB ^[4]	0.8709	0.5830	0.099
CDbin ^[7]	0.9131	0.6291	1.145
BLCD ^[12]	0.9189	0.6302	1.335
DBLD ^[15]	0.9210	0.6289	1.995
DCBMQ ^[21]	0.9037	0.6129	1.617
CFBD ^[17]	0.9150	0.6219	1.610
LDBD ^[22]	0.9047	0.6121	1.895
CAFSD	0.9385	0.6316	1.615

结果如图 6 所示。图 6(a)~(d)为图像旋转以及缩小的各种变换情况。将特征点用线连接起来表示匹配。在图 6(a)~(c),左边的 CAFSD 的线明显多于右边的

CDbin。而图 6(d)空间变换较复杂,CAFSD 匹配的特征点略多。总体上看,CAFSD 匹配的特征点多于 CDbin。

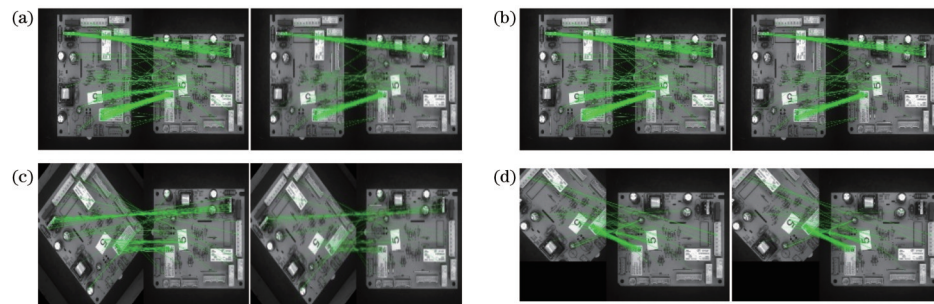


图 6 CAFSD(左)和 CDbin(右)的可视化结果。(a)旋转 45°;(b)旋转 90°;(c)旋转 135°;(d)缩小 0.3+旋转 135°

Fig. 6 Visualization results of CAFSD (left) and CDbin (right). (a) 45° rotation; (b) 90° rotation; (c) 135° rotation; (d) 0.3 diminution + 135° rotation

4 结 论

提出了一种基于通道注意力和特征切片的图像快速匹配算法,核心为 CAFSD。CAFSD 输出的二进制特征描述子可在保证算法一定匹配性能的同时,加快算法的匹配速度。所提算法的测试效果优于其他二进制图像匹配算法,但是在匹配时间上,仍有进一步提升的空间。

参 考 文 献

- [1] 孙琨, 刘李漫, 陶文兵. 基于子空间映射和一致性约束的匹配传播算法[J]. 计算机学报, 2017, 40(11): 2546-2558.
Sun K, Liu L M, Tao W B. A match propagation method based on subspace mapping and coherent constraint[J]. Chinese Journal of Computers, 2017, 40(11): 2546-2558.
- [2] Rosten E, Porter R, Drummond T. Faster and better: a machine learning approach to corner detection[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2010, 32(1): 105-119.
- [3] Calonder M, Lepetit V, Strecha C, et al. BRIEF: binary robust independent elementary features[M]//Daniilidis K, Maragos P, Paragios N. Computer vision - ECCV 2010. Lecture notes in computer science. Heidelberg: Springer, 2010, 6314: 778-792.
- [4] Rublee E, Rabaud V, Konolige K, et al. ORB: an efficient alternative to SIFT or SURF[C]//2011 International Conference on Computer Vision, November 6-13, 2011, Barcelona, Spain. New York: IEEE Press, 2012: 2564-2571.
- [5] 刘自金, 石玉英. 基于信息熵的改进 SIFT 图像快速匹配算法[J]. 激光杂志, 2021, 42(12): 129-135.
Liu Z J, Shi Y Y. Improved fast SIFT image matching algorithm based on information entropy[J]. Laser Journal, 2021, 42(12): 129-135.
- [6] Chu T Y, Chen Y M, Huang L H, et al. A grid feature-point selection method for large-scale street view image retrieval based on deep local features[J]. Remote Sensing, 2020, 12(23): 3978.
- [7] 陈世伟, 夏海, 杨小冈, 等. 基于风格迁移不变特征的 SAR 与光学图像配准算法[J]. 系统工程与电子技术, 2022, 44(5): 1536-1542.
Chen S W, Xia H, Yang X G, et al. SAR and optical image registration algorithm based on style transfer invariable features [J]. Systems Engineering and Electronics, 2022, 44(5): 1536-1542.
- [8] 于子雯, 张宁, 潘越, 等. 基于改进的 SIFT 算法的异源图像匹配[J]. 激光与光电子学进展, 2022, 59(12): 1211002.
Yu Z W, Zhang N, Pan Y, et al. Heterogeneous image matching based on improved SIFT algorithm[J]. Laser & Optoelectronics Progress, 2022, 59(12): 1211002.
- [9] 齐向明, 冯一帆. Sub-Window 尺度空间的 Attention-HardNet 特征匹配算法[J]. 激光与光电子学进展, 2021, 58(22): 2210006.
Qi X M, Feng Y F. Attention-HardNet feature-matching algorithm in sub-Window scale space[J]. Laser & Optoelectronics Progress, 2021, 58(22): 2210006.
- [10] Ye J M, Zhang S L, Huang T J, et al. CDbin: compact discriminative binary descriptor learned with efficient neural network[J]. IEEE Transactions on Circuits and Systems for

- Video Technology, 2020, 30(3): 862-874.
- [11] Fan B, Liu H M, Zeng H, et al. Deep unsupervised binary descriptor learning through locality consistency and self distinctiveness[J]. IEEE Transactions on Multimedia, 2021, 23: 2770-2781.
- [12] Mishchuk A, Mishkin D, Radenovic F, et al. Working hard to know your neighbor's margins: local descriptor learning loss [C]//Advances in Neural Information Processing Systems, December 4-6, 2017, Red Hook, NY, USA. New York: Curran Associates, 2017: 4826-4837.
- [13] Yan C G, Gong B, Wei Y X, et al. Deep multi-view enhancement hashing for image retrieval[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2021, 43(4): 1445-1451.
- [14] Gu X F, Wang Y F, Ma T Y. DBLD-SLAM: a deep-learning visual SLAM system based on deep binary local descriptor[C]//2021 International Conference on Control, Automation and Information Sciences (ICCAIS), October 14-17, 2021, Xi'an, China. New York: IEEE Press, 2021: 325-330.
- [15] Krizhevsky A, Sutskever I, Hinton G E. ImageNet classification with deep convolutional neural networks[J]. Communications of the ACM, 2017, 60(6): 84-90.
- [16] Yang H, Gong C T, Huang K J, et al. Weighted feature histogram of multi-scale local patch using multi-bit binary descriptor for face recognition[J]. IEEE Transactions on Image Processing, 2021, 30: 3858-3871.
- [17] Soleimani P, Li K F, Capson D W. A circular shifting binary descriptor for efficient rotation invariant image matching[C]//2022 26th International Conference on Pattern Recognition (ICPR), August 21-25, 2022, Montreal, QC, Canada. New York: IEEE Press, 2022: 393-399.
- [18] Lai H J, Pan Y, Ye L, et al. Simultaneous feature learning and hash coding with deep neural networks[C]//2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), June 7-12, 2015, Boston, MA, USA. New York: IEEE Press, 2015: 3270-3278.
- [19] Hu J, Shen L, Sun G. Squeeze-and-excitation networks[C]//2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, June 18-23, 2018, Salt Lake City, UT, USA. New York: IEEE Press, 2018: 7132-7141.
- [20] Winder S A J, Brown M. Learning local image descriptors[C]//2007 IEEE Conference on Computer Vision and Pattern Recognition, June 17-22, 2007, Minneapolis, MN, USA. New York: IEEE Press, 2007.
- [21] Duan Y Q, Lu J W, Wang Z W, et al. Learning deep binary descriptor with multi-quantization[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2019, 41(8): 1924-1938.
- [22] Wang Z W, Xiao H, Duan Y Q, et al. Learning deep binary descriptors via bitwise interaction mining[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2023, 45(2): 1919-1933.
- [23] Yang J C, Shi R, Wei D L, et al. MedMNIST v2-a large-scale lightweight benchmark for 2D and 3D biomedical image classification[J]. Scientific Data, 2023, 10: 41.

Fast Image Matching Based on Channel Attention and Feature Slicing

Gai Shaoyan^{1,2*}, Huang Yanyan^{1,2}, Da Feipeng^{1,2**}

¹School of Automation, Southeast University, Nanjing 210096, Jiangsu, China;

²Key Laboratory of Measurement and Control of Complex Systems of Engineering, Ministry of Education, Southeast University, Nanjing 210096, Jiangsu, China

Abstract

Objective Image matching is the process of finding spatial alignment relationships between identical or similar target objects in multiple images. Image matching is one of the research hotspots in the field of optical measurement, which is widely used in image mosaic, product optical measurement, video anti-shake, iterative reconstruction, and other fields. In current research work, accelerating image matching mainly starts from two aspects: accelerating feature point detection speed and accelerating feature matching speed. The most representative method for accelerating feature point detection is the oriented fast and rotated brief (ORB) operator, which detects feature points by comparing the pixel differences between different points. The operation process is relatively simple, and the processing speed is fast. In addition, some algorithms speed up image matching by filtering feature points. Some algorithms reduce image matching time by accelerating the speed of feature matching. Ye *et al.* designed a compact discriminative binary descriptor (CDBin) to obtain binary feature descriptors with smaller number of training parameters. The existing image matching algorithms tend to focus on matching accuracy while neglecting the decrease in matching speed to some extent. In order to solve this problem, an algorithm is proposed, which outperforms most other binary matching algorithms in terms of matching performance and time.

Methods The floating-point feature descriptors output by the feature description algorithm are converted into binary feature descriptors to reduce the computational complexity during feature matching and thus reduce feature matching time. This article is inspired by the classical methods. Gu *et al.* modified the AlexNet structure based on the depth convolutional neural network and mapped the output descriptive sub element value to -1 or 1 . Yang *et al.* used multi-bit binary descriptors to describe image blocks, reducing information loss caused by directly converting real-valued floating-point descriptors into binary descriptors. Soleimani *et al.* proposed a cyclic shift binary descriptor, which reduced the number of

parameters used for calculating descriptors and thus improved matching speed. This article is inspired by the feature representation deep neural network SFLHC, which uses Sigmoid functions and segmented threshold functions separately for each element, binarizes them, and combines the idea of channel attention to improve the binary feature description network. A channel attention and feature slicing description network (CAFSD) is designed, which is combined with the fast feature point detection algorithm, namely ORB. Furthermore, a fast image matching algorithm based on channel attention mechanism and feature slicing is proposed, which can significantly improve the matching speed of images while improving the accuracy of binary description. In addition, based on the triplet loss function, the quantization loss function, uniform distribution loss function, and correlation loss function are introduced to form a composite loss function to optimize network training, further reducing the error of converting floating point descriptors to binary descriptors.

Results and Discussions The core CAFSD feature description network of this algorithm is trained and tested using the UBC Phototour dataset. The UBC Phototour dataset includes three sub datasets: Liberty, NotreDame, and Yosemite. Usually, one dataset is used to train the network, while the other two datasets are used to test the network, and the average value is taken as the final result. In addition, for the testing of the entire image matching algorithm (Fig. 1), 20 circuit board data are used. Commonly used evaluation indicators include FPR95, matching accuracy, matching score, and matching time. FPR95 is an evaluation indicator used in the UBC Phototour dataset to measure the quality of feature description algorithms. Other indicators are used for the overall testing of image matching algorithms, where matching time refers to the time taken by the algorithm from feature detection to the end of feature matching. The results are shown in Table 1. Loss functions of L_T and L_Q , L_T and L_E , as well as L_T and L_C are used to participate in CAFSD training, and UBC Phototour dataset is used for testing. The optimal coefficients for L_Q , L_E , and L_C results are 1, 0.5, and 0.5. In combination with the ORB detection algorithm, the CAFSD has been developed. It can be seen that the speed and accuracy of matching images can be improved obviously.

Conclusions This article proposes a fast image matching algorithm based on channel attention and feature slicing, with the core of the algorithm being CAFSD. Compact discriminative binary descriptor obtains binary feature descriptors with smaller number of training parameters. The existing image matching algorithms tend to focus on matching accuracy while neglecting the decrease in matching speed to some extent. In order to solve this problem, the CAFSD algorithm is proposed in this article. The inspiration for this article comes from the special power of the SFLHC deep neural network, which uses binary Sigmoid functions and piecewise finite functions for each element and combines the idea of channel focusing to improve the binary part network for complex recognition of binary scenes. In combination with the ORB detection algorithm, the CAFSD has been developed. In addition, based on channel attention and additional offloading, a fast image algorithm has been presented and proved.

Key words image processing; image matching; attention; feature description