计算机系统应用 ISSN 1003-3254, CODEN CSAOBN
Computer Systems & Applications,2021,30(11):310−316 [doi: 10.15888/j.cnki.csa.008191]

E-mail: csa@iscas.ac.cn
http://www.c-s-a.org.cn
Tel: +86-10-62661041

基于 SuperPoint 的轻量级特征点及描述子提取网络①

李志强, 朱　明

(中国科学技术大学 信息科学技术学院, 合肥 230027)
通讯作者: 李志强, E-mail: lzqustc@mail.ustc.edu.cn

摘　要: 图像特征点及描述子提取是 SLAM、SFM 和 3D 重建等任务的基础, 较好的图像特征点及描述子提取算法会对这些任务的进步产生十分重要的作用. 本文聚焦于提取特征点和描述子算法中鲁棒性较高、性能较好的 SuperPoint 网络, 对该网络进行了一定程度的改进. 针对其计算量和参数较大的问题, 首先将普通卷积改成深度可分离卷积, 改变卷积层数和下采样方式, 之后改进通道剪枝算法, 使其可以应用于深度可分离卷积, 对网络进行剪枝. 实验结果显示, 在轻微损失特征点检测和匹配效果的情况下, 将网络参数量压缩为原来网络的 15%, 运算量压缩为原来网络的 5%, FPS 提升 6.64 倍, 取得了较好的实验效果.

关键词: SuperPoint; 特征点; 描述子; 深度可分离卷积; 通道剪枝

引用格式: 李志强,朱明.基于 SuperPoint 的轻量级特征点及描述子提取网络.计算机系统应用,2021,30(11):310–316. http://www.c-s-a.org.cn/1003-3254/8191.html

Lightweight Feature Point and Descriptor Extraction Network Based on SuperPoint

LI Zhi-Qiang, ZHU Ming

(School of Information Science and Technology, University of Science and Technology of China, Hefei 230027, China)

Abstract: The extraction of image feature points and descriptors is the foundation of some tasks such as SLAM, SFM, and 3D reconstruction. Preeminent algorithms for image feature point and descriptor extraction play a significant role in processing these tasks. This study accomplishes some improvements in the SuperPoint network with high robustness and good performance in the extraction of feature points and descriptors. Considering the flaws of the heavy calculation burden and massive parameters, the authors first change the ordinary convolution to depthwise separable convolution, the number of layers, and the down-sampling method. Afterward, the channel pruning algorithm is perfected so that it can be applied to depthwise separable convolution and prune the network. Experiments have proved that this study reduces the network parameter number and calculation burden respectively to 15% and 5% those of the original SuperPoint network, and the FPS is increased by 6.64 times under the condition of a slight loss of feature point detection and matching effects. Thus, good experimental results are achieved.

Key words: SuperPoint; feature point; descriptor; depthwise separable convolution; channel pruning

图像特征点及描述子提取是计算机视觉方向研究核心问题之一, 提取图像特征点可以将大量的图像信息稀疏化, 从而更好地进行信息压缩, 而提取描述子则是对特征点周围信息进行描述, 生成特征向量, 用以和

① 基金项目: 安徽省 2019 年重点研究与开发计划 (201904a05020035)
Foundation item: Key Research Plan of Anhui Province in 2019 (201904a05020035)
收稿时间: 2021-01-30; 修改时间: 2021-02-26; 采用时间: 2021-03-24; csa 在线出版时间: 2021-10-22

其他区域进行区分. 图像特征点及描述子提取正被广泛应用于图像匹配[1], 3D 重建[2] 和 SLAM[3] 等诸多领域, 优秀的图像特征点及描述子提取算法可以对计算机视觉的发展发挥巨大的作用.

自 2012 年以来, 深度学习[4] 以其提取特征更丰富、鲁棒性更强、精度更高和不需要手工设计特征等优点在图像分类[5]、目标检测[6] 和图像分割[7] 等领域有了飞速发展. 近些年来, 深度学习与特征点及描述子提取的结合也是该领域研究的一大热点问题.

1 概述

目前的特征点及描述子提取算法主要有以下两类: 完全手工设计特征和通过深度学习提取特征.

完全手工设计特征指的是特征点及描述子提取的算法完全通过手工设计获得, 图像信息获取完全依靠手工设计的算法获取, 如 SIFT 和 SURF 算法. Lowe[1] 提出的 SIFT 算法是通过构建图像高斯金字塔进而构建 DOG 金字塔, 在 DOG 金字塔中寻找满足条件的极值点作为特征点. SIFT 算法是通过统计特征点邻域梯度分布信息提取出特征点主方向, 将坐标轴旋转到主方向, 计算以特征点为中心 16×16 窗口内像素梯度的幅度和方向, 归一化后形成 128 维特征向量作为描述子. 而 Bay 等人[8] 针对 SIFT 算法计算量较大的缺点进行改进提出了 SURF 算法, 利用 Hessian 矩阵提取特征点, 减少了图像下采样的时间消耗. SURF 算法通过统计特征点邻域内的 Harr 小波特征获取特征点主方向, 在沿着特征点主方向的邻域内, 提取 Harr 特征形成 64 维特征向量作为描述子. 完全手工设计的特征点和描述子提取算法是通过数学公式对图片进行进化和抽象来提取信息, 其鲁棒性和泛化性较大规模数据集驱动的深度学习具有天然劣势.

通过深度学习提取特征点或描述子的过程不依靠手工设计的算法获得, 是通过卷积神经网络训练和推理获得. 由于深度学习鲁棒性和泛化性较好, 因此这种方式获得的特征点或描述子也较手工设计的方式泛化性能更好. Tian 等人[9] 提出的 L2-Net 利用卷积神经网络在欧几里得空间学习图像块描述子的特征. 网络输入是图像块, 输出是 128 维向量, 网络以 L2 范数描述图像特征之间的距离, 用损失函数约束匹配上的图像块对的描述子距离尽可能近, 不匹配上的图像块对的描述子距离尽可能远. Mishchuk 等人[10] 提出的 HardNet 在 L2-Net 基础上改进了损失函数. HardNet 参照 SIFT 算法, 使用损失函数最大化最近邻正样本和负样本之间的距离. Barroso-Laguna 等人[11] 提出的 Key.Net 则是将手工设计和卷积神经网络结合形成新的多尺度特征点检测网络. DeTone 等人[12] 的 SuperPoint、Ono 等人[13] 的 LF-Net 和 Dusmanu 等人[14] 的 D2-Net 等均是端到端学习特征, 输入一张图片, 输出特征点和描述子. 其中 SuperPoint 提出一种自监督方式训练网络提取特征点和描述子. SuperPoint 网络有两个分支分别进行特征点和描述子的提取, 共享相同的编码器. LF-Net 由两部分组成: 全卷积网络生成特征点和通过可微采样器在获取特征点附近图像生图像块并将图像块输入描述子提取网络生成描述子. D2-Net 与之前完全手工设计特征算法的先生成特征点再提取描述子过程和 SuperPoint 算法的特征点与描述子并行生成过程均不相同, D2-Net 只生成描述子, 再根据当前点的描述子在最大响应通道中是否为局部最大值判断当前的是否为特征点.

大数据驱动的深度学习能够比完全手工设计的算法提取更深层次的图像特征, 鲁棒性和泛化能力更强. 而端到端学习特征的网络减少了工程的复杂度也减少了误差的累计. 在端到端提取特征点和描述子网络中 SuperPoint 是其中效果较好的网络, SuperPoint 网络输出的特征点和描述子精度较高, 网络结构较其他网络更为简单, 因此本文选择在 SuperPoint 网络上进行进一步优化. SuperPoint 网络虽然网络结构较为简单, 但是对于计算能力较低的设备, 如嵌入式设备, SuperPoint 网络参数量和运算量仍较大, 不能实现实时运行, 因此本文在 SuperPoint 网络基础上进行优化, 构造效果接近 SuperPoint 网络, 但参数量和运算量更低的轻量型网络, 实现在计算能力较低的设备上实时运行的目标.

2 轻量级特征点及描述子提取网络实现

为了实现 SuperPoint 网络的轻量化, 本文首先更改 SuperPoint 网络的卷积方式、卷积层数和下采样方式, 将普通卷积改成深度可分离卷积[15] 并且减少了网络层数, 然后对网络进行进一步剪枝, 进一步减少网络参数和运算量.

2.1 基于深度可分离卷积的 SuperPoint 网络

SuperPoint 是端到端的特征点及描述子提取网络, 网络结构是类似语义分割网络的编码器-解码器结构, 输入一张完整的图片, 经过共享的编码器提取图像深

层特征, 再分别经过特征点和描述子两个解码器, 分别输出特征点和描述子, 与完全手工设计算法的先检测特征点, 再计算描述子不同, 特征点和描述子并行生成. SuperPoint 网络结构如表 1 所示, 表中每一行为一个卷积通道, 第一个数字是输入通道, 中间两个数字是卷积核大小, 最后一个数字是卷积核数目, “+池化”是指在卷积后进行最大池化操作. 共享编码器结构类似与 VGG 网络[16] 的卷积结构, 前 6 层每经过两次 3×3 卷积后紧跟着进行 2×2 最大池化, 共享编码器经过卷积池化等操作后, 进行了图片降维, 提取了深层特征, 减少了后续的计算量. 经过特征点解码器和描述子解码器输出的特征图大小为原图的大小的 1/8, 为了输出了原图一样大小的特征图, 特征点解码器输出的特征图进行 8 倍的子像素卷积, 描述子解码器输出特征图进行 8 倍上采样.

表 1 SuperPoint 网络结构

共享编码器	特征点解码器	描述子解码器
1×3×3×64	128×3×3×256	128×3×3×256
64×3×3×64+池化	256×1×1×65	256×1×1×256
64×3×3×64	—	—
64×3×3×64+池化	—	—
64×3×3×128	—	—
128×3×3×128+池化	—	—
128×3×3×128	—	—
128×3×3×128	—	—

SuperPoint 网络是使用自监督方式进行训练, 训练过程如下所示: (1) 构建包含基础图形的虚拟图片, 如线、多边形和立方体等组成的图片. 已知虚拟图片的角点, 训练编码器和特征点解码器提取特征点. (2) 使用训练好的编码器和特征点解码器输出真实图片及其 N 个经过随机单应性变换图片的特征点, 将 N 个经过随机单应性变换图片的特征点通过逆向单应性变换还原到原图上, 与原图的特征点合并为增强的特征点数据集. (3) 将真实图片及其经过单应性变换后的图片输入 SuperPoint 网络中, 根据特征点的位置和特征点的对应关系训练网络生成特征点及描述子.

本文使用的损失函数与 SuperPoint 网络的损失函数保持一致. 损失函数由特征点损失和描述子损失两部分组成, 如式 (1) 所示:

$$\begin{aligned} &L(X,X',D,D',Y,Y',S)\\ &=L_p(X,Y)+L_p(X',Y')+\lambda L_d(D,D',S) \end{aligned} \tag{1}$$

其中, X, D 分别为原图输入网络后输出的特征点特征图和描述子特征图, Y 为原图特征点的的标签值, X', D'和 Y'对应输入图片为原图经过单应性变换后的图片, 其余含义与 X, D 和 Y 相同, S 由式 (5) 说明. L_p 和 L_d 分别表示特征点损失和描述子损失, 超参数 λ 用来平衡特征点检测损失和描述子损失. L_p 具体公式如下所示:

$$L_p(X,Y)=\frac{1}{H_cW_c}\sum_{\substack{h=1,\\w=1}}^{H_c,W_c} l_p(x_{hw};y_{hw}) \tag{2}$$

其中, H_c, W_c 分别表示特征点特征图的高和宽. x_{hw}, y_{hw} 分别表示 X, Y 在 (h, w) 处的值. l_p 具体公式如下所示:

$$l_p(x_{hw};y)=-\ln\left(\frac{\exp\left(x_{hwy}\right)}{\sum_{k=1}^{65}\exp(x_{hwk})}\right) \tag{3}$$

其中, x_{hwk} 表示为 x_{hw} 在第 k 个通道的值. l_p 使得 x_{hw} 在标签值 y 对应的通道上尽可能大.

$$L_d(D,D',S)=\frac{1}{(H_cW_c)^2}\sum_{\substack{h=1,\\w=1}}^{H_c,W_c}\sum_{\substack{h'=1,\\w'=1}}^{H_c,W_c} l_d\left(d_{hw};d'_{h'w'};s_{hwh'w'}\right) \tag{4}$$

其中, d_{hw}, $d_{h'w'}$分别表示 D, D'在 (h, w), (h', w') 处的值. 由于共享编码器经过 8 倍下采样, 因此输出的描述子特征图中的点对应输入图片中一个 8×8 像素点的图片单元. $s_{hwh'w'}$用来判断 d_{hw} 对应输入图片单元的中心位置经过与原图一致的单应性变换后, 是否在 $d'_{h'w'}$对应输入图片单元的中心位置的邻域内, $s_{hwh'w'}$是用来判断 d_{hw}, $d'_{h'w'}$在原图中对应位置是否相近. $s_{hwh'w'}$=1 表示在原图中对应位置相近, 为正向对应, 反之为反向对应. $s_{hwh'w'}$和 l_d 具体公式如下所示:

$$s_{hwh'w'}=\begin{cases}1, \text{ if } \|Hp_{hw}-p_{h'w'}\|<8\\ 0, \text{ else}\end{cases} \tag{5}$$

其中, p_{hw}, $p_{h'w'}$分别表示 d_{hw}, $d'_{h'w'}$对应的输入图片单元的位置中心. Hp_{hw} 是对 p_{hw}, 进行与原图相同的单应性变换.

$$\begin{aligned} l_d(d;d';s)=&\lambda_d\cdot s\cdot\max\left(0,m_p-d^{\mathrm{T}}d'\right)\\ &+(1-s)\cdot\max\left(0,d^{\mathrm{T}}d'-m_n\right) \end{aligned} \tag{6}$$

其中, 超参数 λ_d 用来平衡描述子内部正向对应损失和负向对应损失值, 超参数 m_p 为正向对应阈值, m_n 为负向对应阈值.

为了降低运算量和参数量, 本文将 SuperPoint 网

络中除共享编码器第一层以外的其余卷积更换成深度可分离卷积. 深度可分离卷积将传统的卷积方式分成逐层卷积和逐点卷积两部分, 如图 1 所示, 左侧为逐层卷积过程, 右侧为逐点卷积过程. 假设输入特征图大小为 $H\times W$, 通道数是 C_{in}, 逐层卷积是对每一个通道使用 $S\times S$ 大小的 1 个通道的卷积核进行卷积, 逐层卷积对通道内的特征信息进行处理. 逐点卷积是使用 1×1 大小的卷积核进行传统方式卷积, 处理通道间的特征信息. 假设输入特征图大小为 $H\times W$, 通道数是 C_{in}, 卷积核大小是 $S\times S$, 卷积核数目是 C_{out}, 对传统卷积来说参数量为 $C_{\text{in}}\times S\times S\times C_{\text{out}}$, 运算量为 $H\times W\times C_{\text{in}}\times S\times S\times C_{\text{out}}$, 对逐层卷积来说, 参数量为 $C_{\text{in}}\times S\times S\times 1$, 为传统卷积的 $1/C_{\text{out}}$, 运算量为 $H\times W\times C_{\text{in}}\times S\times S\times 1$, 为传统卷积的 $1/C_{\text{out}}$. 对逐点卷积来说, 参数量为 $C_{\text{in}}\times 1\times 1\times C_{\text{out}}$, 为传统卷积的 $1/S^2$, 运算量为 $H\times W\times C_{\text{in}}\times 1\times 1\times C_{\text{out}}$, 为传统卷积的 $1/S^2$. 因此深度可分离卷积的参数量和运算量均为传统卷积的 $1/C_{\text{out}}+1/S^2$. 通常来说, 卷积核大小 S 为 3, 卷积核数目 C_{out} 远远大于 9, 因此深度可分离卷积的参数量和运算量大概是传统卷积的 1/9. 深度可分离卷积既处理了通道内的特征信息也处理了通道间的特征信息, 可以替代传统卷积进行, 减少了网络的参数量和运算量.

为了进一步减少网络参数量和运算量, 本文将原始共享编码器的卷积层数和下采样方式进行更改. 具体操作如下所示: (1) 原始编码器的 8 层卷积改成 6 卷积; (2) 将卷积+最大池化的下采样方式更改为步长为 2 卷积, 这样卷积的运算量变为原来的 1/2, 并且省去了最大池化的计算; (3) 为了弥补这些操作带来的特征信息损失, 本文将共享编码器的输出维度设置成 256 维.

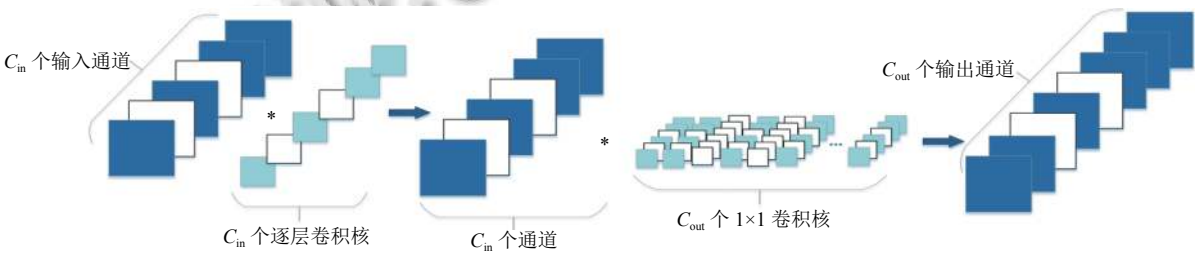


图 1 深度可分离卷积的卷积和剪枝过程

2.2 网络剪枝

为了进一步对网络结构进行优化, 本文进一步对网络进行剪枝, 寻找更优的网络结构. Liu 等人[17] 提出的通道剪枝算法是一种效果较好的算法, 该算法对 VGG 参数量压缩 20 倍, 运算量压缩 5 倍, 而没有影响精度. 但是算法是根据传统卷积设计的, 本文将其进行改进应用于深度可分离卷积中, 压缩本文的网络.

Liu 等人的通道剪枝算法是通过批归一化中 γ 参数衡量通道的重要程度, 删除低于某个阈值的通道, 进而删除与之关联的卷积核, 重新训练进行微调, 完成剪枝过程. 批归一化的过程如式 (7) 所示, 其中 x_i 是批归一化的输入的一个通道的特征, y_i 是批归一化的输出的一个通道的特征, μ_B 和 σ_B 分别为批特征的均值和方差, γ 和 β 为批归一化中的参数. 对于某个通道而言, 若其批归一化中的参数 γ 较小, 其归一化输出值也较小, 可认为该通道不重要, 则可以删除生成这个通道的卷积核与下一层卷积核中对应的该通道的通道. 具体衡量标准是将网络中所有的批归一化参数 γ 进行升序排序, 根据需要将前面一定比例较小的 γ 对应通道进行删除. 中间层的卷积核剪枝应该包括两部分: (1) 输入通道的删除导致卷积核中对应通道的删除; (2) 输出通道的删除导致对应卷积核的删除. 卷积核剪枝意味着寻找到一个较优的网络结构, 再进行训练进行微调提升其精度.

$$y_i=\gamma\frac{x_i-\mu_B}{\sqrt{\sigma_B+\varepsilon}}+\beta \tag{7}$$

Liu 等人的通道剪枝算法是删除所有批归一化中 γ 中较小的值对应的通道. 在深度可分离卷积中, 逐层卷积后续操作也是批归一化, 但是逐层卷积的输入输出通道数应该相同, 因此深度可分离卷积中只能通过逐点卷积中的批归一化衡量通道重要程度. 逐层卷积剪枝是通过输入通道的删除而进行剪枝, 具体流程参见算法 1.

算法 1. 深度可分离网络剪枝算法

1) 将所有非逐层卷积后的批归一化参数 γ 进行升序排序, 删除前 a% 对应的通道;

2) 如图 1 所示, 图中白色部分代表特征图和核卷积核被删除. 输入通道的删除导致逐层卷积核中对应通道的删除以及逐点卷积核中对应通道的删除;
3) 输出通道的删除导致逐点卷积中对应卷积核的删除;
4) 对剪枝后的网络重新训练进行微调.

本文中 a 设置为 20, 剪枝前共享编码器的输出通道为 [64, 64, 128, 128, 256], 特征点解码器输出通道为 [256, 65], 描述子解码器输出通道为 [256, 256], 剪枝后共享编码器的输出通道变为 [35, 47, 94, 86, 147], 特征点解码器输出通道变为 [256, 65], 描述子解码器输出通道变为 [256, 256]. 可以发现, 共享编码器相对于解码器结构更为复杂, 存在较多冗余信息, 被删除较多.

3 实验分析

本文实验过程中使用操作系统为 Ubuntu 18.04, 深度学习框架为 PyTorch 1.3. 在测试 FPS 时使用的硬件环境为 NVIDIA Jetson TX2 开发板和 AMD Ryzen 5 4600U CPU, 其他情况下硬件环境为 Intel Core i7-7800X CPU+ NVIDIA GeForce GTX 1080 Ti GPU. 本文实验过程中超参数设置与 SuperPoint 网络保持一致, 损失函数中 λ=0.000 1, λ_d=250, 正向对应阈值 m_p=1, 负向对应阈值 m_n=0.2. 训练过程中批处理大小为 32, 使用 ADAM 优化器, lr=0.001, β=(0.9, 0.999).

本文分析了改进后的网络和 SuperPoint 网络在参数量、运算量以及 HPatches 数据集[18] 评估效果对比.

3.1 参数量、运算量和 FPS 对比

参数量和运算量对比如表 2 所示, SuperPoint+2.1 代表使用 2.1 节中的优化方法, 即将传统卷积改成深度可分离卷积, 改变卷积层数和下采样方式, SuperPoint+2.1+2.2 代表在 2.1 节优化方法的基础上使用 2.2 节的优化方法, 即进一步进行网络剪枝. 浮点运算数代表运行网络所需要的浮点运算次数用来表示网络计算量. 实验结果表明, 使用 2.1 的优化方式后, 参数量被压缩为原始网络的 22%, 运算量被压缩为原始网络的 8%. 使用本文最终的优化方法 (2.1+2.2 节优化方式) 后, 参数量被压缩为原始网络的 15%, 运算量被压缩为原始网络的 5%, 大大降低了网络参数量和运算量.

表 2 参数量与运算量对比

方法	参数量 (千)	浮点运算数 (百万)
SuperPoint	1304	6557
SuperPoint+2.1	288	543
SuperPoint+2.1+2.2	193	328

FPS 表示网络每秒钟处理图片帧数, 反应网络运行速度. FPS 对比如图表 3 所示, 480×640_tx2 表示实验硬件环境为嵌入式 NVIDIA Jetson TX2 开发板, 使用其配置 GPU 进行推理, 图片分辨率为 480×640. 240×320_cpu 和 480×640_cpu 则表示实验硬件环境均为笔记本小新 PRO13 2020, 使用其配置的 CPU AMD Ryzen 5 4600U 进行推理, 图片分辨率分别为 240×320 和 480×640. 可以看出本文优化后网络的 FPS 在 3 次对比中分别提升 5.07, 7.19 和 7.68 倍, 均值为 6.65 倍, FPS 提升较大. 在 480×640_tx2 和 240×320_cpu 条件下, 本文网络近似实现了实时运行的目标.

表 3 FPS 对比

方法	480×640_tx2	240×320_cpu	480×640_cpu
SuperPoint	5.61	1.47	0.38
SuperPoint+2.1+2.2	28.46	10.57	2.92
提升倍数	5.07	7.19	7.68

3.2 Hatches 数据集评估效果对比

本文参照发布 SuperPoint 论文中的评估方式, 在 Hatches 数据集进行评估, Hatches 数据集是 2017 年发布的特征点及描述子评估数据集. Hatches 数据集内部包含属于 116 的 696 张照片, 其中 57 个场景属于大幅度的光照变化, 59 个场景属于大幅度的视角变化. 本文接下来分别对比 SuperPoint 和本文方法在特征点检测和特征点匹配效果的对比, 对比实现过程中网络使用的超参数设置均相同.

本文使用可重复率和定位误差来判断特征点检测效果. 可重复率指是: 在视角或者光照变化的两张图片中, 同时出现的特征点对占总的特征点数的比率. 定位误差指的是: 同时出现的特征点对的像素点距离的均值. 本文中同时出现的特征点对指的是在相同视角下特征点间距离小于 3 个像素点, 视角不同的特征点对需对其中一张照片经过逆向变换, 从而到相同视角. SuperPoint 和 SuperPoint+2.1 的对比显示: 经过将传统卷积改成深度可分离卷积、改变卷积层数和改变下采样方式后, 虽然使得网络模型变得更简单, 但是可重复率上表现更好, 增加了 1.12%, 定位误差也仅仅增加 0.11. SuperPoint+2.1 和 SuperPoint+2.1+2.2 的对比显示: 进行网络剪枝后, 可重复率降低 0.32%, 定位误差降低 0.02, 网络剪枝带来的特征点检测效果损失可以忽略不记. 表 4 证明, 本文的优化方法 (SuperPoint+

2.1+2.2) 在大幅度降低运算量和参数量的情况下, 并没有导致特征点检测效果大幅度下降, 甚至可重复率表现上更优.

表 4 特征点检测效果对比

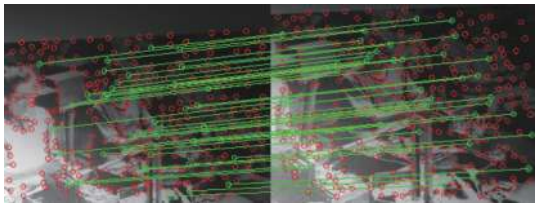
方法	可重复率 (%)	定位误差
SuperPoint	63.19	1.07
SuperPoint+2.1	64.31	1.18
SuperPoint+2.1+2.2	63.99	1.20

要实现特征点匹配效果对比, 首先要通过特征点和描述子获得两幅输入图片的单应性变化矩阵. 获取单应矩阵的过程如下所示: 首先图片 1 和图片 1 经过单应性变换产生的图片 2 分别送入网络中生成特征点和描述子, 描述子通过暴力方式进行最近邻匹配进行配对, 配对的特征点和描述子调用 OpenCV 中 findHomography() 函数, 方法选择 RANSAC 算法, 生成两个图片之间的估计的单应性变换矩阵. 表 5 中单应估计准确率指的是: 图片 1 经过真实的单应变换矩阵的图片边界角点和经过图 2 流程产生的估计的单应变换矩阵的图片边界角点的距离在一定的容忍距离差 e 下的数目占总的数量比例, 单应估计准确率可以反映图片间特征点匹配效果. 表 5 中对比 SuperPoint 和 SuperPoint+2.1, 可以发现: 本文使用的方法, 在 e=1 时下降 0.03, 在 e=3 时下降 0.02, 在 e=5 时下降 0.02, 仅用轻微的降幅, 在可以接受范围内. SuperPoint+2.1 和 SuperPoint+2.1+2.2 的对比表明: 剪枝算法几乎没有降低单应估计准确率, 仅在 e=1 时下降 0.02. 表 5 证明, 本文使用的方法并没有导致的特征点匹配效果大幅降低, 匹配精度仍然较高.

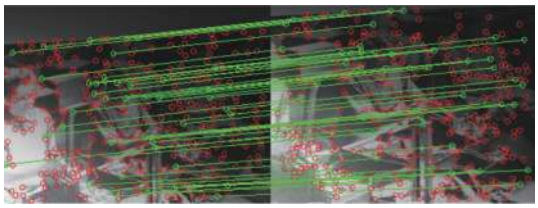
表 5 单应估计准确率对比

方法	e=1	e=3	e=5
SuperPoint	0.47	0.76	0.83
SuperPoint+2.1	0.44	0.74	0.81
SuperPoint+2.1+2.2	0.42	0.74	0.81

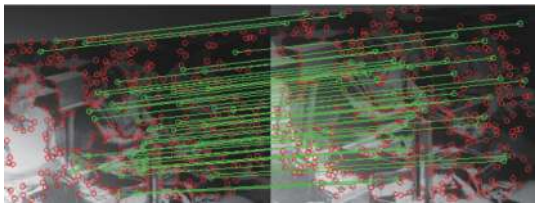
图 2 展示的为原始 SuperPoint 网络、使用 2.1 节优化后的网络和最终优化后的网络的在同一幅图片上特征点检测和匹配的效果图对比图, 红色点为检测到的特征点, 绿色线为匹配特征点. 对比 3 幅图片可以发现 3 个网络检测到的特征点数和匹配的特征点数都比较相近, 说明本文最终优化后的网络在参数量压缩为原来的 15%, 运算量运算量压缩为原来的 5% 和 FPS 提升 6.64 倍的同时, 特征点检测和匹配的效果几乎没有降低.



(a) SuperPoint



(b) SuperPoint+2.1



(c) SuperPoint+2.1+2.2

图 2 特征点检测与特征点匹配效果图

4 结论与展望

本文针对 SuperPoint 网络的参数量和运算量较大, 在嵌入式设备上不能实现实时运算的缺点, 对其网络结构进行精简和优化. 首先, 本文将深度可分离卷积应用于 SuperPoint 网络中并且改变了网络的层数和下采样方式. 然后本文将 Liu 的通道剪枝算法进行改进, 使其可以应用于深度可分离卷积中. 实验结果表明, 最终优化后的网络网络参数量被压缩为原始网络的 15%, 运算量被压缩为原始网络的 5%, 运行的 FPS 较原始网络提升 6.64 倍, 在计算资源有限的嵌入式和 CPU 上也能近似实现实时运行, 网络特征点检测和匹配效果较原始网络仅有轻微幅度下降.

下一步的研究工作在于将本文的网络与 SLAM 等算法进行结合, 用本文提出的特征点和描述子提取算法代替传统的特征点和描述子提取算法, 构建一个更鲁棒的 SLAM 算法.

参考文献

1 Lowe DG. Distinctive image features from scale-invariant keypoints. International Journal of Computer Vision, 2004, 60(2): 91–110. [doi: 10.1023/B:VISI.0000029664.99615.94]

2 Snavely N, Seitz SM, Szeliski R. Photo tourism: Exploring photo collections in 3D. ACM Transactions on Graphics, 2006, 25(3): 835–846. [doi: 10.1145/1141911.1141964]
3 Mur-Artal R, Montiel JMM, Tardos JD. ORB-SLAM: A versatile and accurate monocular SLAM system. IEEE Transactions on Robotics, 2015, 31(5): 1147–1163. [doi: 10.1109/TRO.2015.2463671]
4 LeCun Y, Bengio Y, Hinton G. Deep learning. Nature, 2015, 521(7553): 436–444. [doi: 10.1038/nature14539]
5 Krizhevsky A, Sutskever I, Hinton GE. ImageNet classification with deep convolutional neural networks. Communications of the ACM, 2017, 60(6): 84–90. [doi: 10.1145/3065386]
6 Redmon J, Divvala S, Girshick R, *et al*. You only look once: Unified, real-time object detection. Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition. Las Vegas: IEEE, 2016. 779–788.
7 Long J, Shelhamer E, Darrell T. Fully convolutional networks for semantic segmentation. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. Boston: IEEE, 2015. 3431–3440.
8 Bay H, Tuytelaars T, Van Gool L. Surf: Speeded up robust features. European Conference on Computer Vision. Graz: Springer, 2006. 404–417.
9 Tian YR, Fan B, Wu FC. L2-Net: Deep learning of discriminative patch descriptor in Euclidean space. 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). Honolulu: IEEE, 2017. 6128–6136.
10 Mishchuk A, Mishkin D, Radenovic F, *et al*. Working hard to know your neighbor's margins: Local descriptor learning loss. Proceedings of the 31st International Conference on Neural Information Processing Systems. Red Hook: Curran Associates Inc., 2017. 4826–4837.
11 Laguna A B, Riba E, Ponsa D, *et al*. Key.Net: Keypoint detection by handcrafted and learned CNN filters. Proceedings of the 2019 IEEE/CVF International Conference on Computer Vision. Seoul: IEEE, 2019. 5836–5844.
12 DeTone D, Malisiewicz T, Rabinovich A. Superpoint: Self-supervised interest point detection and description. Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops. Salt Lake City: IEEE, 2018. 224–236.
13 Ono Y, Trulls E, Fua P, *et al*. LF-Net: Learning local features from images. Proceedings of the 32nd International Conference on Neural Information Processing Systems. Red Hook: Curran Associates Inc., 2018. 6234–6244.
14 Dusmanu M, Rocco I, Pajdla T, *et al*. D2-net: A trainable CNN for joint description and detection of local features. Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. Long Beach: IEEE, 2019. 8092–8101.
15 Howard AG, Zhu ML, Chen B, *et al*. Mobilenets: Efficient convolutional neural networks for mobile vision applications. arXiv: 1704.04861, 2017.
16 Simonyan K, Zisserman A. Very deep convolutional networks for large-scale image recognition. arXiv: 1409.1556, 2014.
17 Liu Z, Li JG, Shen ZQ, *et al*. Learning efficient convolutional networks through network slimming. Proceedings of the 2017 IEEE International Conference on Computer Vision. Venice: IEEE, 2017. 2736–2744.
18 Balntas V, Lenc K, Vedaldi A, *et al*. HPatches: A benchmark and evaluation of handcrafted and learned local descriptors. 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). Honolulu: IEEE, 2017. 3852–3861.