International Journal of Computer Vision manuscript No.
(will be inserted by the editor)

GMS: 基于网格的运动统计，用于快速、超鲁棒的特征匹配

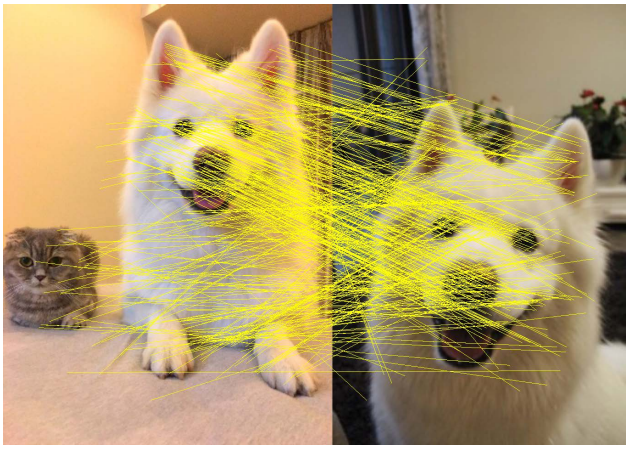
Jia-Wang Bian[1,2] · Wen-Yan Lin[3] · Yun Liu[4] · Le Zhang[5] · Sai-Kit Yeung[6] · Ming-Ming Cheng[4] · Ian Reid[1,2]

Received: date / Accepted: date

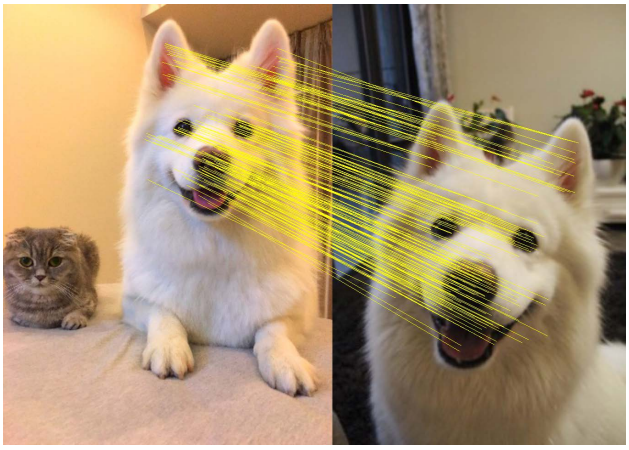
摘要 特征匹配旨在生成图像之间的匹配，并且在许多计算机视觉任务中得到了广泛使用。尽管在特征描述符和快速匹配方面已经取得了长足的进步，但从得到的初始匹配中选择出正确的匹配仍然非常困难，并且影响着整体性。更重要的是，现有选取正确匹配的方法通常需要较长的计算时间，从而限制了它们在实时应用程序中的使用。本文提出 GMS（Grid-based Motion Statistics）算法旨在将正确与错误的匹配进行高速分离。该方法利用平滑度约束和统计分析来判断正误，并使用基于网格的实现方式进行快速计算。GMS 在应对包括视点，尺度和旋转等各种图像变化时具有高鲁棒性。并且它的速度也很快，例如，即使当需要处理匹配数目达到 50k，在单个 CPU 线程中也仅需花费 1 或 2 毫秒。这对于实时的应用程序来说具有重要意义。此外，我们展示了 GMS 对经典的特征匹配和对极几何估计所带来的性能提升。最后，我们将 GMS 集成到著名的 ORB-SLAM 系统中来提升单眼初始化模块，获得了明显的改进。

Keywords 特征匹配, 对极几何, 视觉 SLAM, GMS

[1] The University of Adelaide, Australia
[2] Australian Centre for Robotic Vision, Australia
[3] Singapore Management University, Singapore
[4] TKLNDST, CS, Nankai University, China
[5] Agency for Science, Technology and Research, Singapore
[6] Hong Kong Univ. of Science and Technology, Hong Kong



(a) ORB-RT



(b) ORB-RT-GMS

图 1 尽管 Lowe 的比率测试（*ratio test*，RT）可以消除 ORB (Rublee et al. 2011) 特征在此处产生的许多错误匹配，但结果仍然很嘈杂（a）。我们提出了 GMS 算法进一步消除的运动不一致的错误匹配。

1 介绍

特征匹配是计算机视觉中最基本的问题之一，它旨在生成同一场景中不同视图之间的匹配，该匹配广泛

用于计算机视觉任务中，例如三维重建 (Schonberger & Frahm 2016) 和视觉 SLAM (Davison et al. 2007; Mur-Artal et al. 2015)。经典方法依赖于特征检测器 (Harris & Stephens 1988)，描述符 (Lowe 2004) 和用来生成初始匹配的算法 (Muja & Lowe 2009) 。通常初始匹配会经过基于 RANSAC (Fischler & Bolles 1981) 的估算器来拟合几何模型并同时去除异常值，然后保留下来的匹配会被用作更高级任务的输入。尽管在特征提取，匹配器和估计器方面已经取得了长足的进步，但整体性能仍然受到错误匹配的限制，即它们导致估计器无法找到正确的模型和正确的匹配。这个问题很关键，但是与上述其他问题相比，受到的关注相对较少。更重要的是，现有的匹配选取方法非常耗时 (Lin et al. 2017)，限制了它们在实时应用程序中的使用。为了解决这一问题距，我们提出了一种称为 GMS（Grid-based Motion Statistics）的新方法，用于高速分离正确和错误的匹配。

本文提出的方法依赖于*运动平滑度*约束，即我们假设一幅图像中的相邻像素会一起移动，因为它们通常会降落在一个刚性物体或结构中。尽管该假设并不总是正确的，如在图像边缘区域，但该假设对大多数常规像素都适用。这对于我们的目的已经足够了，因为我们的目标不是产生最终的匹配，而是为基于 RANSAC 的方法提供高质量的匹配。该假设会导致一个图像中的相邻的正确匹配在其他图像中也将距离较近，而错误的匹配则不然。这使得我们可以通过仅计算*相似邻居*（在两个图像中都接近目标匹配的匹配）数量的方法来将匹配分类为正确或错误。图 2 中展示了该假设的可视化，并在 Sec. 3.2进行了理论分析。

计算成本对于错误匹配删除方法至关重要，因为特征匹配通常被用于实时应用程序中，如视觉 SLAM (Mur-Artal et al. 2015)等。我们在 Sec. 3.3 中提出基于网格的快速计算框架，在该框架中，我们将图像划分为非重叠网格，并在网格级别而不是在单个匹配级别处理数据。这避免了不同匹配之间的距离比较，从而将整体复杂性从 $O(N^2)$ 降低到 $O(N)$。最终我们得到结果，即使当匹配数量达到 50K 时，GMS 在单个线程中也仅花费 1 或 2 毫秒的 CPU 时间来识别正确匹配，如 Fig. 11所示。

基础的网格框架不能很好的处理图像在尺度和旋转方面发生重大变化的情况。为了解决这个问题，我们提出了多尺度，多旋转的解决方案。具体来说，我们分别在图像对之间定义 5 个相对比例和 8 个相对旋转角。然后，我们在不同的设置下重复 GMS 算法，并保留最佳的匹配结果。由于在不同的重复中不存在数据依赖性，因此可以使用多线程编程来实现所提出的方法。从理论上讲，当 8（或 5）个 CPU 线程可用时，它们可以与基本 GMS 算法一样快。普通台式机或笔记本电脑能够负担得起这种资源需求。

我们对 GMS 进行了全面的评估，包括对常见图像变化的鲁棒性，以及在不同特征数量下的性能变化。此外，我们在最近的 FM-Bench (Bian et al. 2019) 上评估了所提出的方法来分析其对于对极几何估计的帮助。结果表明 GMS 相对于其他最新技术具有明显的优越性。最后，我们将提出的 GMS 合并到著名的 ORB-SLAM (Mur-Artal et al. 2015)中以进行单目初始化，并展示出了显著的改进。本文是我们初版 (Bian et al. 2017) 的扩展。我们从以下四个方面进行扩展：(a) 更直接，更清楚的演示；(b) 缩放和旋转的增强；(c) 更全面的评估；(d) 在实时应用的中使用。

2 相关工作

与 RANSAC 的关系. 本文所提出的方法旨在踢除错误匹配。尽管它与基于 RANSAC 的算法有关，但请注意 GMS 不能替代后者。不同点包括：(i) GMS 无法像 RANSAC 类似算法一样拟合模型；(ii) 模型的离群值在概念上不等同于错误匹配。例如，在静态场景假设下的三维重建问题中，如果某些正确的匹配落在运动物体中，则它们也将作为离群值被删除。GMS 的目标不是代替 RANSAC，而是向后者提供高质量的匹配以提高整体性能。

错误匹配移除. Lowe 的 *比率测试* (Lowe 2004), 也称为 RT, 是最广泛使用的一种方法。它通过比较两个最近候选点的距离以识别高质量的匹配。但是由于缺乏更有力的约束，在复杂场景仍然难以踢除许多错误匹配。Fig. 1 展示了一个例子，其中显示 GMS 可以进一步消除错误匹配。其他方法包括 KVLD (Liu & Marlet 2012)，它在图像和几何方面都加以约束；VFC (Ma et al. 2014)，它在两个点集之间插入矢量场并估计共识集。最近的 CODE (Lin et al. 2017)算法利用非线性优化对运动平滑度进行全局建模。我们的方法受此启发，但更简单和高效。LPM (Ma et al.

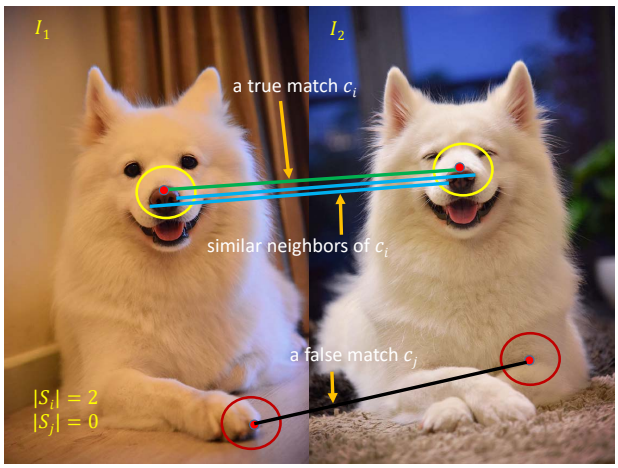


图 2 运动统计正确的匹配通常比错误匹配拥有更多的相似邻居，因此我们以相似邻居的数量来区分它们。

2019) 探索周围特征点的局部结构，这比 GMS 定义了更严格的假设。LC (Yi et al. 2018) 使用深度神经网络通过拟合基本矩阵来找正确匹配。它更像是 RANSAC (Fischler & Bolles 1981)的替代方案。但是作者表明该预测模型不比 RANSAC 更好。他们反而建议使用该方法提前查找正确匹配，然后继续使用 RANSAC 进行模型拟合。

使用 GMS 的应用. 会议版本的 GMS 发布后，我们注意到许多近期的工作使用了我们的方法并取得了卓越的性能。例如，(Causo et al. 2018) 在用于亚马逊机器人挑战赛的物品拣选系统中使用了 GMS，该系统可以在最短的时间内拣选所有目标物品。(Zhang et al. 2019) 使用并扩展了 GMS 来解决动态场景中的相机定位问题。最终的算法在 TUM 数据集 (Sturm et al. 2012) 上比 ORB-SLAM2 (Mur-Artal et al. 2015)计算的相机轨迹更精确了一个数量级。(Yoon et al. 2018) 将 GMS 用于 3D 轨迹重建系统中的点云三角化剖分。此外，GMS 已集成到 OpenCV 库 (Bradski 2000) 中，我们鼓励研究人员在更多的实时应用中使用和扩展该方法。

3 基于网格的运动统计

给定由特征检测器，描述符和匹配器生成的初始匹配，我们的目标是将正确匹配与错误匹配分开。

3.1 运动平滑假设

为了区分正确与错误的匹配，我们假设图像上坐标接近的像素会一起移动。这一点通常会成立，因为相邻像素落在同一个刚性物体上的可能性很高，因此它们在不同照片中的距离都很相近。注意该假设并不总是成立的，例如，当相邻像素落在不同的且会独立运动的物体上时，该假设不成立。但是，这种假设适用于大多数常规的像素，这些像素比边缘区域要多得多。此外，由于我们的目标是要为了得到一组高质量匹配来作为 RANSAC 的输入，并不是最终的匹配，因此该假设足以满足我们的目的。

3.2 运动统计

正确的匹配会受到平滑度约束的影响，而错误的匹配则不然。因此，如图 2 所示，正确的匹配通常比错误的匹配具有更多的相似邻居，其中相似邻居指的是在两个图像中都接近于目标匹配的匹配。我们使用相似邻居的数量来识别好的匹配。

令 C 为图像 I_1 和 I_2 上的所有匹配，c_i 是连接两个图像之间的点 p_i 和 q_i 的一个匹配。我们把 c_i 的邻居定义为：

$$N_i = \{c_j | c_j \in C, c_j \neq c_i, d(p_i, p_j) < r_1\}, \tag{1}$$

它的相似邻居为

$$S_i = \{c_j | c_j \in N_i, d(q_i, q_j) < r_2\}, \tag{2}$$

其中 $d(\cdot,\cdot)$ 表示两点之间的欧拉距离，r_1, r_2 为阈值。$|S_i|$ 指的是 S_i 中元素的数量，表示对 c_i 的运动支持。

运动支持可以作为区分正确与错误匹配的判别特征。为了区分真实和错误匹配对 S_i 建模，我们得到：

$$|S_i| \sim \begin{cases} B(|N_i|, t), & \text{if } c_i \text{ 是正确的} \\ B(|N_i|, \epsilon), & \text{if } c_i \text{ 是错误的} \end{cases} \tag{3}$$

其中 $B(\cdot,\cdot)$ 表示二项分布。$|N_i|$ 表示 c_i 的邻居数量。t 和 ϵ 分别是其邻居为的正确或错误匹配的概率。

在 Eqn. 3中，t 由特征质量决定，它接近正确的匹配的比率。ϵ 通常很小，因为错误匹配几乎是随机分布的。需要注意的是，在视觉上相似但位置不同的区域，例如重复结构中 (Kushnir & Shimshoni 2014)，

它可能会更大。在这里，我们假设由特征导致的匹配要好于随机分布造成的匹配，即 t 比 ϵ 大。

我们可以得出 $|S_i|$'s 的期望：

$$E_{|S_i|} = \begin{cases} E_t = |N_i| \cdot t, & \text{if } c_i \text{ 是正确的} \\ E_f = |N_i| \cdot \epsilon, & \text{if } c_i \text{ 是错误的} \end{cases} \tag{4}$$

和方差:

$$V_{|S_i|} = \begin{cases} V_t = |N_i| \cdot t \cdot (1-t), & \text{if } c_i \text{ 是正确的} \\ V_f = |N_i| \cdot \epsilon \cdot (1-\epsilon), & \text{if } c_i \text{ 是错误的} \end{cases} \tag{5}$$

因此我们可以定义正确和错误匹配之间的可分割性：

$$\begin{aligned} P &= \frac{|E_t - E_f|}{\sqrt{V_t} + \sqrt{V_f}} \\ &= \frac{|N_i| \cdot (t-\epsilon)}{\sqrt{|N_i| \cdot t \cdot (1-t)} + \sqrt{|N_i| \cdot \epsilon \cdot (1-\epsilon)}} \end{aligned} \tag{6}$$

其中 $P \propto \sqrt{|N_i|}$ 且当 $|N_i| \to \infty$, $P \to \infty$. 这意味着随着特征数足够大，基于 $|S_i|$ 的正确和错误匹配的可分离性变得越来越可靠。即使 t 仅略大于 ϵ，也会发生这种情况，从而可以通过简单地增加检测到的特征的数量的方式来获得在困难场景上的可靠的匹配。在 (Lin et al. 2018)中也展示了类似的结果，其中通过大量的独立试验将歧义分布分开此外，它表明提高特征质量（t）也可以提高可分离性。

这种独特的属性使我们可以通过简单地对 $|S_i|$ 进行阈值设定的方式来将 ci 分类为正确还是错误，从而得出：

$$c_i \in \begin{cases} \mathcal{T}, & \text{if } |S_i| > \tau_i \\ \mathcal{F}, & \text{其他情况} \end{cases} \tag{7}$$

其中 $\mathcal{T}$ 和 $\mathcal{F}$ 分别表示正确和错误匹配的集合。基于 Eqn. 6，我们设定 τ_i 为：

$$\tau_i = \alpha\sqrt{|N_i|}, \tag{8}$$

其中 α 为超参数。并且我们根据经验发现，当 α 的范围为 4 到 6 时，它会表现出良好的性能。

3.3 基于网格的框架

计算 S_i 的简单实现的复杂度为 $O(N)$，由于我们需要对 c_i 和所有其他的匹配进行对比，因此式中的

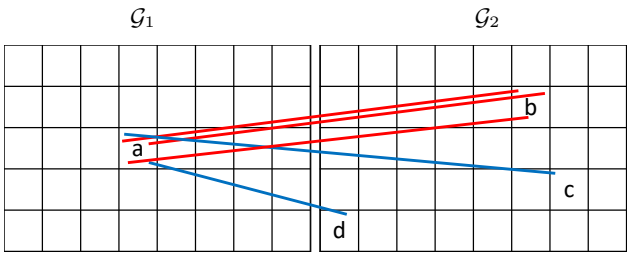


图 3 基于网格的框架。我们使用预先计算的网格查找相似邻居，而不是在点之间进行显式的距离比较。

$N = |C|$ 表示所有的匹配。因此，总的算法复杂度为 $O(N^2)$ 尽管采用近似最近邻算法，但可以将复杂度降低到 $O(Nlog(N))$, 但我们提出的基于网格的框架更快（$O(N)$）.

Fig. 3展示了该框架，其中我们将两个图像分别划分为非重叠单元 $\mathcal{G}_1$ 和 $\mathcal{G}_2$。假设 c_i 是分布在单元 G_a 和 G_b 上的匹配，就像 Fig. 3中的一条红色的匹配。c_i 的邻居可以重新定义为：

$$N_i = \{c_j | c_j \in C_a, c_i \neq c_j\}, \tag{9}$$

相似邻居可以重新定义为：

$$S_i = \{c_j | c_j \in C_{ab}, c_i \neq c_j\}, \tag{10}$$

其中 C_a 为落在 G_a 中的匹配，C_{ab} 为同时落在 G_a 和 G_b 中的匹配。换句话说，我们将落在同一个单元中的匹配称为 *neighbors*，把落在同一单元对的匹配称为相似邻居。这避免了匹配之间的显式比较。要获得所有匹配的运动支持，我们只需要将它们放在单元对中即可。用这种方式，总的复杂度被缩减成 $O(N)$。

注意落在一个单元对中的匹配共享相同的运动支持, 因此我们只需要对单元对进行区分，无需对单独的匹配进行区分。而且，不同于确定所有可能的单元对，我们仅检查包含第一张图像中包含最多匹配的一个最佳的单元对。例如，在 Fig. 3中，我们只检查 G_{ab} 并丢弃 G_{ac}, G_{ad}。此操作将大大减少单元对的数量，并尽早排除相当数量的错误匹配。

3.4 运动内核

如果单元大小比较小，考虑的邻居也很少。这会降低性能。然而，如果单元大小比较大，会将不准确的匹配包括进来。为解决此问题，我们将网格大小设置为较小以提高准确性，并提出运动内核以考虑更多

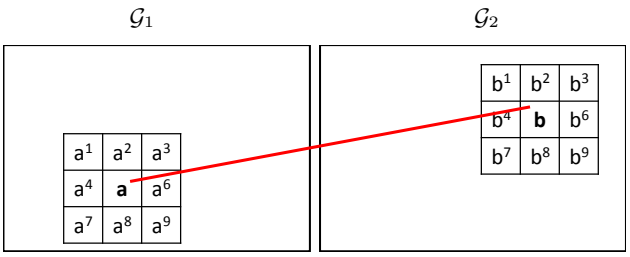


图 4 基础运动内核我们在计算原始单元对（C_{ab}）的运动支持时考虑周围的单元对（$C_{a^1b^1}$, ..., $C_{a^9b^9}$）。

邻居。Fig. 4展示了基础的运动内核，其中我们为了区分原本的单元对（C_{ab}），考虑了额外 8 个单元对（$C_{a^1b^1}$, ..., $C_{a^9b^9}$）。我们令 c_i 再次落在 C_{ab} 中。我们重定义它的邻居为：

$$N_i = \{c_j | c_j \in C_A, c_i \neq c_j\}, \tag{11}$$

其中

$$C_A = C_{a^1} \cup C_{a^2} \cup C_{a^3} ... \cup C_{a^9}. \tag{12}$$

我们重定义它的相似邻居为

$$S_i = \{c_j | c_j \in C_{AB}, c_i \neq c_j\}, \tag{13}$$

其中

$$C_{AB} = C_{a^1b^1} \cup C_{a^2b^2} \cup C_{a^3b^3} ... \cup C_{a^9b^9}. \tag{14}$$

基础内核假定两个图像之间的相对旋转很小。为了匹配具有明显旋转变化的图像对，我们可以旋转基础内核，如下一节所述。

3.5 多尺度与多旋转

为了处理两个图像之间的显著的尺度和旋转变化，我们在本节中提出了多尺度和多旋转解决方案。

多旋转方案 我们旋转了基础内核以模拟不同的相对旋转，得到了共计 8 个运动内核，如 Fig. 5所示。

在现实世界中，旋转通常是未知的，因此我们使用所有运动内核运行 GMS 算法并收集最佳结果，即，我们找到了最终产生大多数检索出的匹配的内核。多旋转方案的功效在 Sec. 4.2中得到了展示。

多尺度方案 在我们的网格框架中可以模拟两幅图像之间的相对比例，即我们固定一张图的单元大小（单元数），改变另一张图的单元大小。假设每张图像分

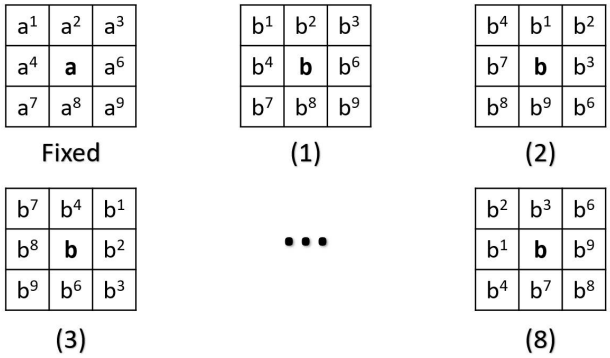


图 5 旋转运动内核。我们将第一幅图像中的图案固定，然后将第二幅图像中的图案沿顺时针方向旋转，总共生成了 8 个运动内核，用于模拟可能的相对旋转。

成了 $n \times n$ 个单元。我们改变第二张图像的单元数量为 $(n \cdot \alpha) \times (n \cdot \alpha)$。在这里，我们为 α 提供了 5 个候选值，包括 $\{\frac{1}{2}, \frac{\sqrt{2}}{2}, 1, \sqrt{2}, 2\}$。同样，我们使用所有预定义的相对尺度运行 GMS，并收集最佳结果。请注意，我们仅提供 5 个相对比例，以证明我们的解决方案对于解决比例问题是有效的，这在大多数场景下都是充分和高效的，在 Sec. 4.2 和 Sec. 4.4中展示。然而，对于更多显著的尺度变化，我们可以使用更多的候选值或增大 α 的值。

Algorithm 1 Grid-based Motion Statistics

Input: C, S, K {匹配, 尺度, 内核}
Output: X {选中的匹配}

```
G1, G2 = GenerateGrids(S); {Fig. 3}
for each G_a ∈ G1 do
    Find G_b from G2 with G_ab having most matches
    C_A, C_AB, C_ab = Search(K, G_ab); {Fig. 4, Fig. 5}
    τ = α√(|C_A| − 1); {Eqn. 8}
    s = |C_AB| − 1; {Eqn. 13}
    if s > τ then
        X = X ∪ C_ab;
    end if
end for
```

在水平，垂直和两个方向上将第一张图像的网格移动半个单元格宽度，然后重复该算法 3 次以上。

return X

3.6 算法和局限性

Alg. 1展示了 GMS 算法，将推定的匹配以及比例和旋转的设置作为输入并输出选中的匹配项。我们使用基础运动内核和单个等比例来匹配常规图像，例

如视频帧。多尺度和多旋转方案分别用于尺度和旋转显著变化的图像。

实现. 我们使用 C++ 中的 OpenCV 实现了该算法。(Bradski 2000) 当前使用单个 CPU 线程，但是可以使用多线程编程来加速多尺度和多旋转解决方案。我们在默认模式下使用 20×20 个单元，在激活多尺度解决方案时，我们会改变第二张图片中的单元数量。 Eqn. 8中的 $\alpha = 4$ 用作阈值。该代码已集成到 OpenCV 库中。

局限性. GMS 的局限性在于三个方面。首先，由于我们假设图像运动是逐段平滑的，因此在违反假设的区域中性能可能会下降，例如在图像边界处。这个问题并不严重，因为常规像素的数量远远超过边界。此外，由于我们的目标不是最终的通信解决方案，而是一组高质量的假说，因此该假设足以满足我们的目的。为了解决这个问题，我们考虑使用在未来的工作中使用边缘检测 (Liu et al. 2019)或图像分割 (Cheng et al. 2016; Liu et al. 2018)技术。

其次，在视觉上相似但位置上不同的图像区域中，性能会受到限制。在具有大量重复图案的场景中时常会发生此问题。由于仅局部视觉信息不足以解决该问题，因此我们将问题留给全局几何估计算法 (Kushnir & Shimshoni 2014)。第三，当我们在单元对级别处理数据时，将保留正确单元对中的不正确对应。这些匹配在许多对匹配精度不敏感的应用中很有用，例如对象检索 l (Philbin et al. 2007)。但是，对于精度敏感的任务，例如几何估计，应将其排除在外。因此，为了缓解此问题，我们建议在 RT 选出的匹配而不是所有假定的匹配上运行 GMS 算法。使用 RT 进行预处理的效果在 Fig. 7 和 Tab. 4上。

4 实验

我们从 4 个方面评估 GMS：

- 匹配选择的综合性能表征。
- 匹配具有显著的相对尺度和旋转变化的具有挑战性的图像对。
- 对特征匹配和对极几何估计总体性能的贡献。
- 集成在实时计算机视觉应用程序中。

4.1 GMS 用于选择匹配

为了全面评估 GMS 的性能，我们尝试使用不同的局部特征和不同的特征数量。我们还使用变化的错误阈值来检查检索到的匹配的准确性。两个从 VGG (Mikolajczyk & Schmid 2005)中选出的具有挑战性的数据集（*Graffiti* 和 *Wall*）被用于评估，它们以视点发生重大变化而闻名。每个数据集包含六张图像，其中提供了第一张图像与其他图像之间的 Ground Truth 单应性，从而导致五对图像的测试难度越来越大。

*召回率*和*准确率*被用做评估指标，我们认为到 Ground Truth 的距离小于 10 个像素的匹配是正确的，其他则是错误的。

4.1.1 不同特征上的结果

Fig. 6展示了 *Graffiti* 和 *Wall* 数据集, 其中 SIFT (Lowe 2004) 和 ASIFT (Morel & Yu 2009)分别被用于生成匹配。我们在所有匹配和由 Lowe 的*比率测试*选出的匹配上都测试了 GMS。

由于视点发生重大变化，因此 SIFT 无法在非常具有挑战性的关系对上提供足够的正确匹配，而 ASIFT 可以。结果表明，GMS 对 ASIFT 生成的匹配具有较高的查全率和查准率，而对 SIFT 生成的匹配则具有较低的查全率。原因是 ASIFT 提供了足够的匹配，而 GMS 可以将高特征编号转换为高匹配质量，如 Eqn. 6中所示，而 SIFT 匹配则很少。

但是，在 SIFT 匹配上，尽管 RT-GMS 很少，但可以实现高性能。注意 RT 的*召回率*是 RT-GMS 的上限。这是因为 GMS 的性能还与特征质量相关，如 Eqn. 6，并且 RT 选择的匹配质量在准确性上要高于初始匹配。与众所周知的最广泛使用的特征匹配方法 SIFT-RT 相比，所提出的 SIFT-RT-GMS 可以得到相似的召回率（即相似的对应编号），但精度更高。它对许多计算机视觉应用程序都具有重要意义。我们在 Tab. 2展示了两种方法的对比，在对极几何估计方面，具有挑战性的宽基线数据集的性能差距很大。

4.1.2 匹配的准确性

Fig. 7显示了在 *Graffiti* 上误差阈值变化的结果，其中我们使用 *Graffiti*（5 对）中的所有匹配进行评估。注意，关于正确对应的数量，ALL 是 GMS 的上限，

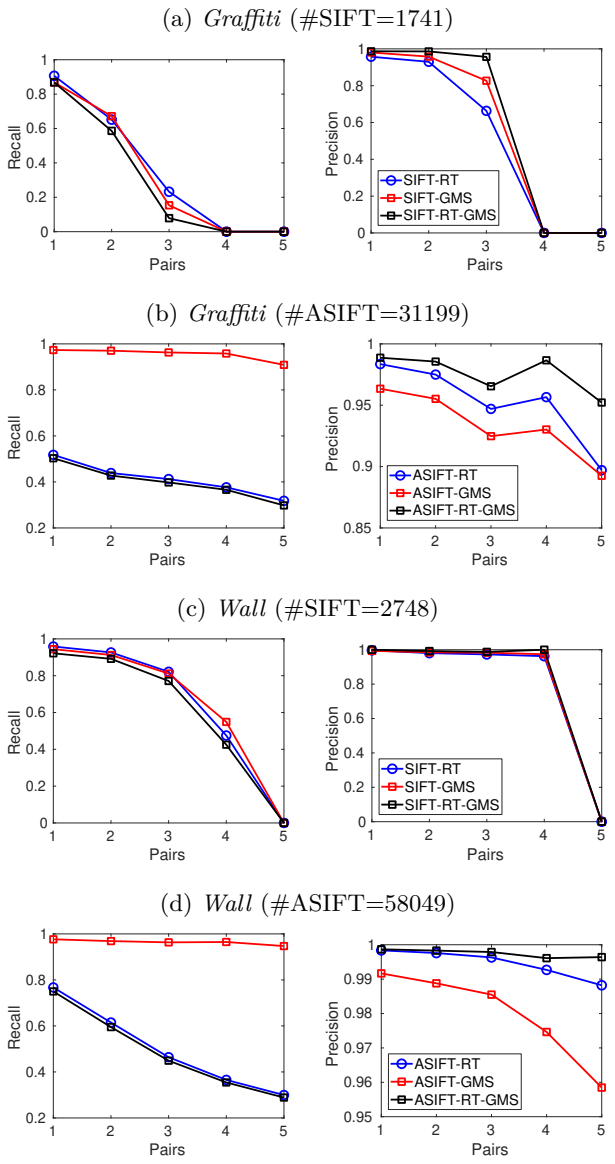


图 6 对视点变化逐渐增加的结果。“#”代表特征数量。RT 代表 Lowe 的比率测试 (Lowe 2004)。GMS 和 RT 将所有匹配作为输入，而 RT-GMS 将 RT 的结果作为输入。因此，RT 的召回率是 RT-GMS 的上限。

RT 是 RT-GMS 的上限。结果表明，ASIFT 能够比 SIFT 提供更好的匹配（请参阅 ALL 进行比较）。然而，当 RT 或 GMS 应用时，SIFT 匹配的精度高于 ASIFT 匹配的精度，特别是当误差阈值很小，如 1 或 2 个像素的时候。这是因为 ASIFT 生成了许多正确但不准确的匹配，并由 RT 和 GMS 选择了这些匹配。这些不准确的匹配在精度要求不严格的许多应用中很有用。例如对象检索 (Philbin et al. 2007)中。

但是，它们限制了对精度敏感的应用程序的性能，例如对极几何估计。因此，我们建议使用 SIFT-RT-GMS 解决方案以实现高精度的匹配。注意，由

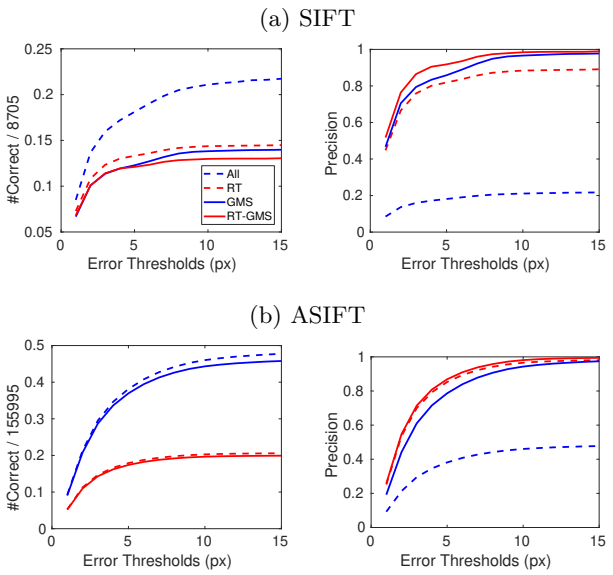


图 7 在 *Graffiti* 上具有不同错误阈值的结果。我们收集了总共 5 对中的所有匹配以进行评估。

于 ASIFT 匹配通常比 SIFT 匹配更好，因此使用 ASIFT 也是可能的，并且可能会有更高的性能。例如，(Lin et al. 2017)使用 ASIFT 特征并在相机姿态估计方面达到了最先进的性能。但是，它使用了高度复杂的回归方法，要消除不准确的匹配会消耗大量的计算成本。

4.1.3 不同特征数量下的性能

Eqn. 6 表示 GMS 的性能依赖于特征数量。为了了解特征数量如何对其产生影响，我们随机选择了具有不同最大特征数量的 ASIFT 特征子集进行评估。Fig. 8展示了 *Graffiti* 数据集上的结果。它表明 GMS 需要最多的特征，而 RT 的性能对特征数量的敏感性较低。RT-GMS 可以减少 GMS 的数量需求，并实现最精确的匹配。总体而言，我们建议用户在特征数量受限的情况下尝试使用 RT 和 RT-GMS。此外，请注意，GMS 的性能也与特征质量有关。我们证明，使用相同的特征检测器（即相同的特征数量），使用更好的描述符也可以提高性能，见 Sec. 4.4。

4.2 对尺度和旋转角度变化的鲁棒性

我们使用 *Semper* 和 *Venice* 数据集，用于分别测试 GMS 对旋转和缩放变化的鲁棒性。这两个数据集都选自 Heinly (Heinly et al. 2012)，是和 VGG (Mikolajczyk & Schmid 2005)具有相同数据组织形式的拓

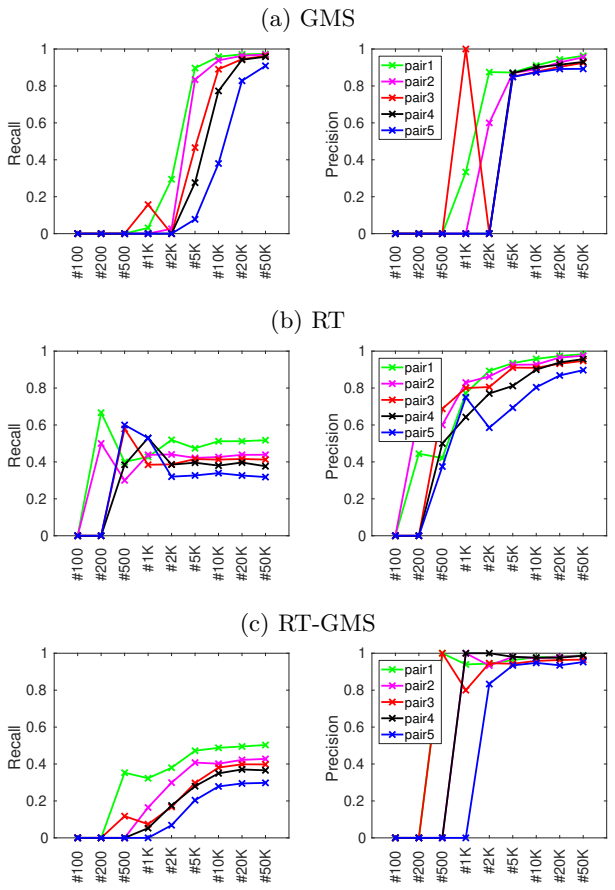


图 8 在 *Graffiti* 数据集上不同特征数量的结果。我们随机选择每个图像中检测到的 ASIFT 特征的子集进行评估。

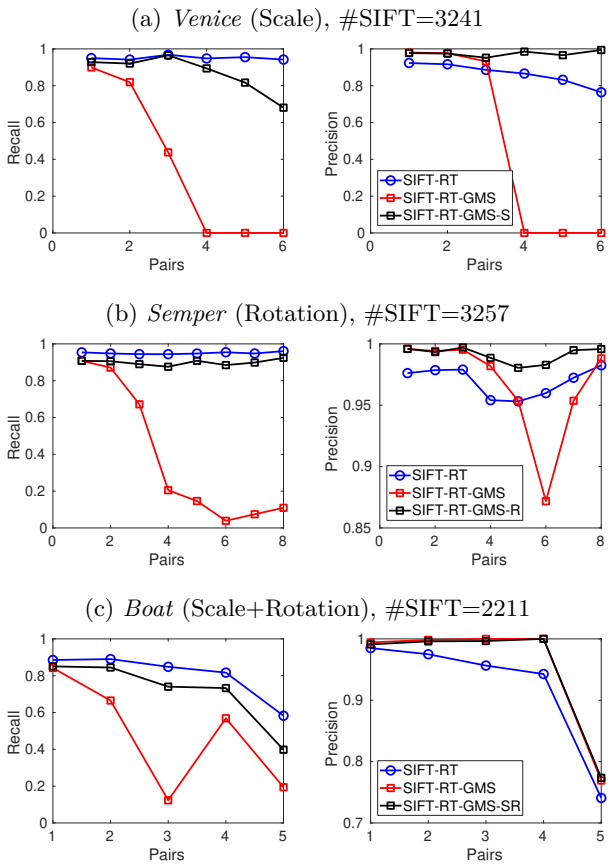


图 9 尺度和旋转角度变化的鲁棒性。GMS-S，GMS-R 和 GMS-SR 分别是指我们的具有多尺度，多旋转度和两者兼有的方法。

展。我们也使用 *Boat* 数据集进行测试，其中图像对在尺度和旋转角度具有明显的相对变化。Fig. 9 展示了我们使用 SIFT 功能生成推定的匹配，并在 RT 选择的匹配上运行 GMS 变体的结果。这表明基本的 GMS 对大尺度和旋转变化敏感，而使用多尺度和多旋转解决方案可以显着提高性能。与以前的结果相似，所提出的方法可以实现相似的召回率和更高的精度。我们在 Fig. 10中展示了 GMS 的可视化结果，其中展示了每个数据集上最具挑战性的结果。尺度不变性在非结构化环境中至关重要，因为图像的相对尺度未知。我们在 Tab. 4展示了使用提出的多尺度解决方案可以在有挑战的宽基线场景下有效提高性能。

4.3 GMS 的运行时间

我们使用 ASIFT 特征评估在 *Graffiti* 数据集上具有不同特征数量的 GMS 的运行时间。Fig. 11展示了结果，其中即使特征数量达到 50,000，GMS 在单个 CPU 线程中也只需约 2 毫秒。多刻度（GMS-S）和多旋转（GMS-R）变体分别以不同的预定义模式重复基础 GMS5 次和 8 次。因此，它们的计算成本线性增加。为了清楚起见，我们在图中未显示 GMS-SR。但是，它的耗时很明显，即比基础 GMS 高 40 倍。请注意，多尺度和多旋转解决方案在不同的重复中都不依赖于数据，因此可以通过使用多线程编程来加速它们。从理论上讲，当有 5（或 8）个 CPU 线程可用时，多尺度（或旋转）版本可以达到与基本 GMS 相同的速度。

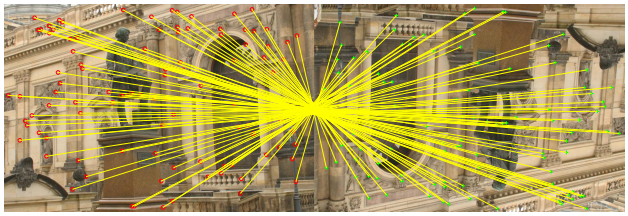
4.4 GMS 用于对极几何估计

我们在 FM-Bench (Bian et al. 2019)上评估了提出的方法，其中匹配的选择方法集成到了经典特征匹配和对极几何估计流水线中 (即 SIFT，RANSAC 和 8 点算法)，，并且对比了整体表现。基准数据集，指标和基准如下所述。

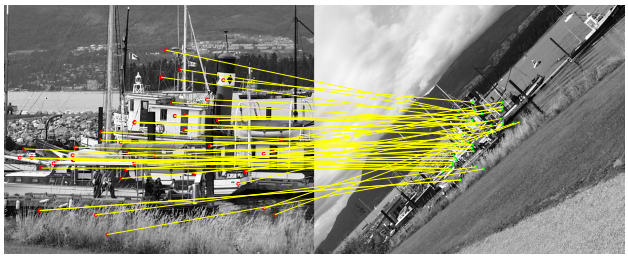
数据集. FM-Bench 由 4 个数据集，包括 TUM (Sturm et al. 2012), KITTI (Geiger et al. 2012), Tanks 和

(a) ASIFT-RT-GMS on *Graffiti*

(b) SIFT-RT-GMS-S on *Venice*



(c) SIFT-RT-GMS-R on *Semper*



(d) SIFT-RT-GMS-SR on *Boat*

图 10 GMS 的可视化结果。我们在每个数据集中最具挑战性的对上显示匹配结果，其中最多绘制了 100 个匹配以进行清晰的可视化表示。

Temples (T&T) (Knapitsch et al. 2017)，和一个社区照片集（CPC）组成。前两个数据集在视觉 SLAM 测试中很常见，它们分别提供了室内和室外视频。后两个数据集被广泛用于运动结构评估，它们提供了广泛的基线方案。特别的，CPC 数据集具有挑战性，因为其中图像是由游客捕获并从网上收集而来的。依照(Bian et al. 2019)随机从每个数据集中选出了 1000 个匹配图像对用于测试。Tab. 1总结了测试数据集，在 Fig. 12中展示了样本图像。

基准线. 我们对比了时下三个最先进的匹配选择方法，包括 CODE (Lin et al. 2017), LPM (Ma et al. 2019), LC (Yi et al. 2018)作为基准线。CODE 利用

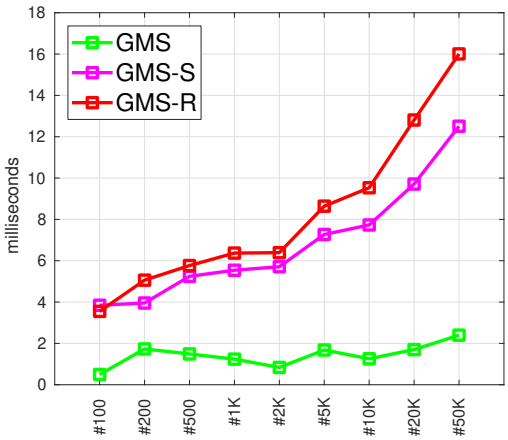
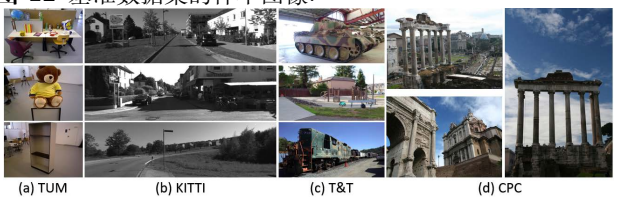


图 11 GMS 在单 GPU 上的运行时间。GMS-S 和 GMS-R 使用不同的设置分别重复基础 GMS 方法 5 次和 8 次，因此运行时间线性增加。GMS-SR（未在图中显示）消耗的计算成本是基础 GMS 的 40 倍。注意，由于在不同的重复中不存在数据依赖性，因此可以通过使用多阈值编程来加速多尺度和多旋转解决方案。

Table 1 基准数据集细节.

Datasets	#Seq	#Images	Resolution	Baseline	Property
TUM	3	5994	480 × 640	short	indoor scenes
KITTI	5	9065	370 × 1226	short	street views
T&T	3	922	1080 × 2048 1080 × 1920	wide	outdoor scenes
CPC	1	1615	varying	wide	internet images

图 12 基准数据集的样本图像.



复杂的非线性优化来查找正确的匹配，并且依赖于自行实现的 GPU-ASIFT 功能，该功能提取的功能是标准 ASIFT (Morel & Yu 2009)的几倍

LPM 探索邻居结构，LC 使用深度训练的神经网络。这两种方法都是特征独立的，并且在原始论文中使用了 SIFT (Lowe 2004)方法。请注意，这种比较对我们的方法不公平，因为 CODE 使用 ASIFT 功能和非常复杂的解决方案（比 GMS 算法慢了 100 倍）。

实现细节. 我们使用 DoG 检测器和 SIFT 描述符 (Lowe 2004)生成推定的匹配。该实现来自 VLFeat 库 (Vedaldi & Fulkerson 2010)。我们使用默认参数，在 4 个数据集上分别产生了平均 1082,1751，8133 和 7213 个

Table 2 % 基本矩阵估计的召回率.

Method	Dataset			
	TUM	KITTI	T&T	CPC
CODE	62.50	92.50	89.40	51.00
SIFT-RT	57.40	**91.70**	70.00	29.20
SIFT-RT-LPM	58.90	91.50	80.70	39.40
SIFT-RT-LC	54.10	89.70	76.60	39.40
SIFT-RT-GMS	**59.20**	**91.70**	**80.90**	**43.00**

Table 3 % 经过 RANSAC 后的正确匹配

Method	Dataset			
	TUM	KITTI	T&T	CPC
CODE	76.95	98.32	89.14	90.16
SIFT-RT	75.33	98.20	75.20	67.14
SIFT-RT-LPM	75.75	98.27	81.62	78.17
SIFT-RT-LC	75.96	**99.44**	84.01	83.99
SIFT-RT-GMS	**76.18**	98.58	**84.38**	**85.90**

Table 4 多尺度解决方案和 RT 预处理的消融研究结果 (FM %Recall)。

Method	Dataset			
	TUM	KITTI	T&T	CPC
SIFT-RT-GMS	**59.20**	**91.70**	**80.90**	**43.00**
without RT	51.9	90.6	73.4	31.4
without Multi-Scale	X	X	78.6	37.8

检测出的特征。使用比率测试（RT）预处理产生的匹配，阈值设置为 0.8。之后我们在选择出的好的匹配上应用评估方法 (LPM, LC, and GMS)。

根据经验，尽管在原始论文中使用 LC (Yi et al. 2018)的匹配，但使用 RT 的匹配比使用 LC 的所有匹配具有更好的性能。我们在前两个 SLAM 数据集中使用基本的 GMS，在后两个 SfM 数据集中使用多尺度解决方案，因为图像的结构更加不规则。由于这些数据集中没有明显的图像旋转，因此不使用多旋转解决方案。在 CODE (Lin et al. 2017)中。由于它是一个高度集成的匹配系统，因此我们直接评估输出的匹配。我们将公开可用的实现用于所有方法，并使用 LC 的作者预先训练的模型。

Table 5 具有基于深度学习特征的 GMS 的结果（FM % Recall）粗体和下划线分别表示第一和第二性能。

Method	Dataset			
	TUM	KITTI	T&T	CPC
CODE	<u>67.50</u>	<u>91.90</u>	**92.70**	**61.80**
DoG-HardNet-RT-GMS	**68.60**	**92.10**	<u>92.20</u>	60.10
HesAff-HardNet-RT-GMS	66.40	91.80	90.90	<u>60.80</u>

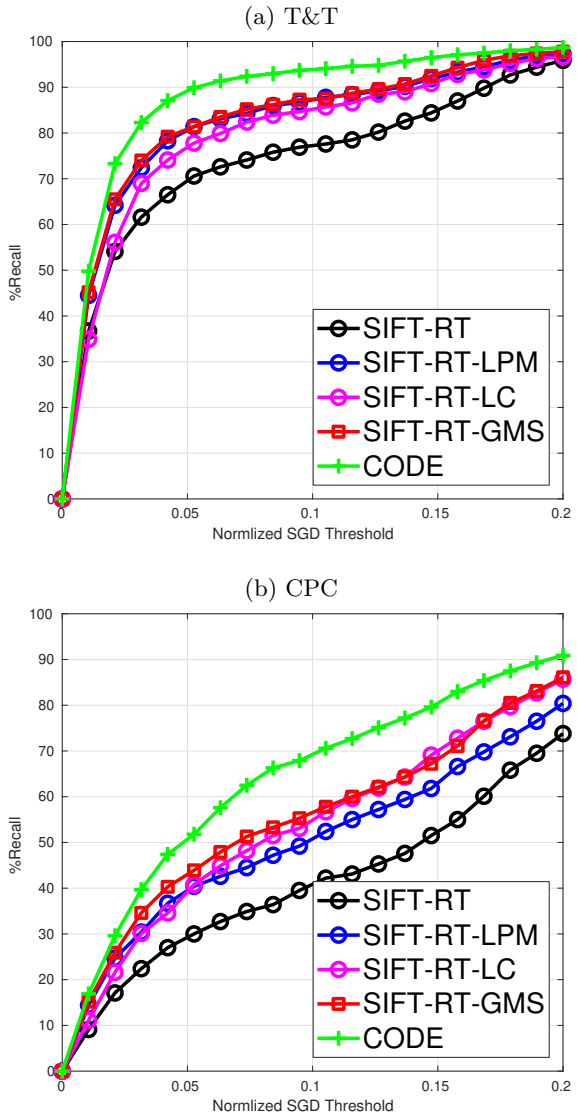


图 13 宽基线数据集的 FM 估计。使用相同特征的匹配作为输入，GMS 可以胜过最近的 LC 和 LPM 方法。

指标. 为了比较在不同匹配系统上的整体性能，我们将其匹配输入到基于 RANSAC 的 8 点估计器 (Fischler & Bolles 1981; Hartley 1997)中以恢复 FM，然后使用归一化对称几何距离（NSGD） (Bian et al. 2019)将估计的 FM 与 Ground Truth 进行比较有关详细信息，请参见附录。之后，我们报告了 FM 估计

的成功率（**%Recall**），其中 NSGD 的阈值为 0.05，并且我们还展示了不同误差阈值的结果。

此外，我们报告了基于 RANSAC 的离群值移除后的正确率（%lier），用于匹配质量比较。这里的正确率指与 Ground Truth 极线的距离小于 $\alpha * l$ 的匹配，其中 l 代表对角线的长度，$\alpha = 0.003$。更多细节可以在附录和(Bian et al. 2019)中找到。

实验结果. Tab. 2展示了 FM 估计的召回率。Fig. 13显示了在两个宽基线数据集上具有不同错误阈值的结果。

Tab. 3 报告了正确率. 上述所有结果都表明了, 在使用相同特征的匹配输入时,GMS 可以在 LC 和 LPM 中表现出更好的性能, 即 SIFT-RT (Lowe 2004).

然而, 我们的匹配系统并不如强大的 CODE (Lin et al. 2017)系统. 与 SIFT-RT 比较，我们的方法可以在 T& T 和 CPC 数据集上带来显着更好的结果，证明了 GMS 进行高精度匹配的功效。从运行时间角度考虑，CODE 需要几秒钟的时间进行对应选择，而 GMS 则要快 1000 倍以上。正如其作者提到的，LC 需要在 GPU 上运行 13 毫秒（或 CPU 上 25 毫秒）来从 2000 个推定的匹配中找到好的匹配，LPM 可以在几毫秒内从 1,000 个推定对应中识别出错误匹配。

它们都比我们提出的算法慢（Fig. 11）。

RT 和多尺度的影响. 为了了解我们的多尺度解决方案以及在 GMS 之前使用 RT 会如何改善整体性能，我们分别进行了消融研究，以将其分别移除。结果展示在 Tab. 4中。它表明在没有 RT 的情况下性能会显着下降。原因是包括了不正确的匹配，它们限制了几何估计的性能。此外，宽基线匹配数据集（T&T 和 CPC）的结果表明，使用提出的多尺度解决方案可以显着提高性能。

与深度特征配对. 正如 Eqn. 6中展示的，GMS 的性能与特征质量有关，我们尝试了最近提出的基于深度学习的特征，包括 HardNet (Mishchuk et al. 2017) 检测器和 HessAff (Mishkin et al. 2018) 检测器。特别的，我们分别使用了 DoG (Lowe 2004)和 HessAff，用于插入点检测。然后，我们使用 HardNet 来计算描述符，从而得出两个推定的匹配生成解决方案。选用强大的 CODE 作为强大的基准。

Table 6 KITTI 里程表数据集上的单目 SLAM 初始化结果.

Seq	Frag	Success Ratio		Orders		#3D Points	
		ORB	*GMS*	*ORB*	*GMS*	*ORB*	*GMS*
00	227	0.77	**0.95**	2.8	**1.18**	140.44	**929.59**
01	55	0.11	**0.80**	4.16	**3.88**	119.00	**519.77**
02	233	0.73	**0.98**	4.28	**1.14**	124.28	**858.01**
03	40	0.78	**0.98**	2.77	**1.28**	136.97	**881.74**
04	13	0.69	**0.92**	6.78	**1.08**	123.11	**875.00**
05	138	0.80	**0.95**	2.65	**1.38**	132.99	**848.60**
06	55	0.70	**0.96**	5.13	**1.38**	116.62	**704.24**
07	55	0.73	**0.87**	2.0	**1.31**	133.33	**882.92**
08	203	0.66	**0.96**	3.65	**1.23**	126.69	**786.53**
09	79	0.65	**0.98**	3.39	**1.24**	129.94	**790.01**
10	60	0.75	**0.98**	5.31	**1.32**	122.00	**843.56**

在这里，我们使用基于 LMedS (Rousseeuw & Leroy 1987)的 FM 估计器代替 RANSAC，因为它比 FM-Bench 中比 RANSAC 显示出更好的性能。

Tab. 5展示了结果，其中具有深层特征的 GMS 可以获得能与 CODE 竞争的性能。由于这些深层特征非常有效，而 CODE 的速度却慢了几个数量级，因此结果是惊人的，并对实时应用具有重要意义。

4.5 用于单目 SLAM 初始化的 GMS

单目 SLAM 方法 (Mur-Artal et al. 2015) 在追踪和定位之前必须要进行系统初始化，通过对匹配进行三角剖分以创建深度来创建初始 3D 地图。可靠的初始化对于单目 SLAM 系统具有重要意义，而高质量的匹配是此步骤的关键。由于 Visual SLAM 系统对整体实时性能的方法的运行时间有很高的要求，因此在这种情况下不能使用许多先进的匹配方法。幸运的是, GMS 在这个目的下足够快。在本节中，我们展示了可以在流行的 ORB-SLAM 系统 (Mur-Artal et al. 2015) 中使用 GMS 进行更好的初始化。

整合. 在 ORB-SLAM 的初始化程序中，我们将原始的基于词袋的匹配替换为蛮力最近邻匹配，并应用 GMS 选择良好的匹配。选定的匹配用于恢复几何形状并通过三角剖分创建 3D 地图。为了系统稳定，我们使用默认的特征检测参数，即检测 $4K$ 个分布良好的 ORB 特征以在类似 KITTI 的图像中进行初始化。

数据集和指标. 我们根据 KITTI 里程表数据集 (Geiger et al. 2012)的序列 00-10 评估方法进行评估。对于每个序列，我们将其裁剪为非重叠片段，每个片段包含 20 个连续的帧。报告所有片段的平均性能。在每一个片段中，我们评测（a）初始化是否成功；（b）系统初始化有多快；和（c）生成了多少 $3D$ 点。关于 (a)，我们将估计的相机姿态与地面真实性进行比较，如果姿态误差在旋转和平移中均小于 5 度，则这些被认为是成功的。关于（b），我们报告每个片段中第一个成功初始化的图像的顺序。

实验结果. Tab. 6展示了实验结果，其中我们对比了原始的初始化程序。它表明，提出的初始化程序可导致更高的成功率，更快的初始化速度和更密集的 3D 地图。这对单目 SLAM 系统具有巨大影响，尤其是当它们在具有挑战性的环境中工作时，以前的解决方案无法为初始化提供可靠的匹配。所提出的方法可以缓解此问题，并可以在更多实际场景中使用 SLAM 系统。

5 总结

本文提出了一种快速的匹配选择算法，我们称之为基于网格的运动统计（GMS）。通过利用运动平滑度约束，它可以快速有效地将真实的匹配与错误的分离开来。全面的实验结果证明了其在不同环境中的鲁棒性。我们还表明，它可以促进特征匹配和几何估计。此外，我们将 GMS 插入到 Monocular ORB-SLAM 系统中进行初始化，从而证明了其在实时应用中的巨大潜力。算法代码已经发布并且被集成到了 OpenCV 库中。

致谢

这项工作得到了澳大利亚机器人视觉卓越中心 CE140100016 的支持。这项工作得到了新一代人工智能重大项目（No. 2018AAA0100403），国家自然科学基金（61922046）和天津自然科学基金会（No. 18JCYBJC41300 和 No. 18ZXZNGX00110）的支持。这项研究得到了新加坡教育部（MOE）学术研究基金（AcRF）一级资助的支持。这项研究得到新加坡教育部学术研究基金 MOE2016-T2-2-154 的部分支持，以及 HKUST 的内部资助（R9429）。本文作者们感谢 Yashiyuki Matsushita 教授对先前版本的帮助。

参考文献

Bian, J., Lin, W.-Y., Matsushita, Y., Yeung, S.-K., Nguyen, T. D., & Cheng, M.-M. (2017). GMS: Grid-based motion statistics for fast, ultra-robust feature correspondence. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, (pp. 4181–4190).

Bian, J.-W., Wu, Y.-H., Zhao, J., Liu, Y., Zhang, L., Cheng, M.-M., et al. (2019). An evaluation of feature matchers for fundamental matrix estimation. In *British Machine Vision Conference (BMVC)*.

Bradski, G. (2000). The OpenCV Library. *Dr. Dobb's Journal of Software Tools*.

Causo, A., Chong, Z.-H., Luxman, R., Kok, Y. Y., Yi, Z., Pang, W.-C., et al. (2018). A robust robot design for item picking. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, (pp. 7421–7426).

Cheng, M.-M., Liu, Y., Hou, Q., Bian, J., Torr, P., Hu, S.-M., et al. (2016). Hfs: Hierarchical feature selection for efficient image segmentation. In *European Conference on Computer Vision (ECCV)*. Springer, (pp. 867–882).

Davison, A. J., Reid, I. D., Molton, N. D., & Stasse, O. (2007). Monoslam: Real-time single camera slam. *IEEE Transactions on Pattern Recognition and Machine Intelligence (PAMI)*, *29*(6), pp. 1052–1067.

Fischler, M. A., & Bolles, R. C. (1981). Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, *24*(6), pp. 381–395.

Geiger, A., Lenz, P., & Urtasun, R. (2012). Are we ready for autonomous driving? the kitti vision benchmark suite. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, (pp. 3354–3361).

Harris, C., & Stephens, M. (1988). A combined corner and edge detector. In *Alvey Vision Conference*. (pp. 10–5244).

Hartley, R. I. (1997). In defense of the eight-point algorithm. *IEEE Transactions on Pattern Recognition and Machine Intelligence (PAMI)*, *19*(6), pp. 580–593.

Heinly, J., Dunn, E., & Frahm, J.-M. (2012). Comparative Evaluation of Binary Features. In *European Conference on Computer Vision (ECCV)*. (pp. 759–773).

Knapitsch, A., Park, J., Zhou, Q.-Y., & Koltun, V. (2017). Tanks and Temples: Benchmarking large-scale scene reconstruction. *ACM Transactions on Graphics (TOG)*, *36*(4), p. 78.

Kushnir, M., & Shimshoni, I. (2014). Epipolar geometry estimation for urban scenes with repetitive structures. *IEEE Transactions on Pattern Recognition and Machine Intelligence (PAMI)*, *36*(12), pp. 2381–2395.

Lin, W.-Y., Lai, J.-H., Liu, S., & Matsushita, Y. (2018). Dimensionality' s blessing: Clustering images by underlying distribution. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. (pp. 5784–5793).

Lin, W.-Y., Wang, F., Cheng, M.-M., Yeung, S.-K., Torr, P. H., Do, M. N., et al. (2017). Code: Coherence based decision boundaries for feature correspondence. *IEEE Transactions on Pattern Recognition and Machine Intelligence (PAMI)*.

Liu, Y., Cheng, M.-M., Hu, X., Bian, J.-W., Zhang, L., Bai, X., et al. (2019). Richer convolutional features for edge detection. *IEEE Transactions on Pattern Recognition and Machine Intelligence (PAMI)*.

Liu, Y., Jiang, P.-T., Petrosyan, V., Li, S.-J., Bian, J., Zhang, L., et al. (2018). Del: Deep embedding learning for efficient image segmentation. In *International Joint Conference on Artificial Intelligence (IJCAI)*.

Liu, Z., & Marlet, R. (2012). Virtual line descriptor and semi-local matching method for reliable feature correspondence. In *British Machine Vision Conference (BMVC)*. (pp. 16–1).

Lowe, D. G. (2004). Distinctive image features from scale-invariant keypoints. *International Journal on Computer Vision (IJCV)*, *60*(2), pp. 91–110.

Ma, J., Zhao, J., Jiang, J., Zhou, H., & Guo, X. (2019). Locality preserving matching. *International Journal on Computer Vision (IJCV)*, *127*(5), pp. 512–531.

Ma, J., Zhao, J., Tian, J., Yuille, A. L., & Tu, Z. (2014). Robust point matching via vector field consensus. *IEEE Transactions on Image Processing (TIP)*, *23*(4), pp. 1706–1721.

Mikolajczyk, K., & Schmid, C. (2005). A performance evaluation of local descriptors. *IEEE Transactions on Pattern Recognition and Machine Intelligence (PAMI)*, *27*(10), pp. 1615–1630.

Mishchuk, A., Mishkin, D., Radenovic, F., & Matas, J. (2017). Working hard to know your neighbor's margins: Local descriptor learning loss. In *Neural Information Processing Systems (NIPS)*. (pp. 4826–4837).

Mishkin, D., Radenovic, F., & Matas, J. (2018). Repeatability is not enough: Learning affine regions via discriminability. In *European Conference on Computer Vision (ECCV)*. Springer, (pp. 284–300).

Morel, J.-M., & Yu, G. (2009). Asift: A new framework for fully affine invariant image comparison. *SIAM Journal on Imaging Sciences*, *2*(2), pp. 438–469.

Muja, M., & Lowe, D. G. (2009). Fast approximate nearest neighbors with automatic algorithm configuration. *VISAPP (1)*, *2*(331-340), p. 2.

Mur-Artal, R., Montiel, J. M. M., & Tardos, J. D. (2015). ORB-SLAM: a versatile and accurate monocular slam system. *IEEE Transactions on Robotics (TOR)*, *31*(5), pp. 1147–1163.

Philbin, J., Chum, O., Isard, M., Sivic, J., & Zisserman, A. (2007). Object retrieval with large vocabularies and fast spatial matching. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, (pp. 1–8).

Ranftl, R., & Koltun, V. (2018). Deep fundamental matrix estimation. In *European Conference on Computer Vision (ECCV)*. (pp. 284–299).

Rousseeuw, P. J., & Leroy, A. M. (1987). *Robust regression and outlier detection*, volume 589. John wiley & sons.

Rublee, E., Rabaud, V., Konolige, K., & Bradski, G. (2011). Orb: An efficient alternative to sift or surf. In *IEEE International Conference on Computer Vision (ICCV)*. IEEE, (pp. 2564–2571).

Schonberger, J. L., & Frahm, J.-M. (2016). Structure-from-motion revisited. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. (pp. 4104–4113).

Sturm, J., Engelhard, N., Endres, F., Burgard, W., & Cremers, D. (2012). A benchmark for the evaluation of rgb-d slam systems. In *IEEE International Conference on Intelligent Robots and Systems (IROS)*. IEEE, (pp. 573–580).

Vedaldi, A., & Fulkerson, B. (2010). VLFeat: An open and portable library of computer vision algorithms. In *ACM International Conference on Multimedia (ACM MM)*. ACM, (pp. 1469–1472).

Yi, K. M., Trulls, E., Ono, Y., Lepetit, V., Salzmann, M., & Fua, P. (2018). Learning to find good correspondences. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. (pp. 2666–2674).

Yoon, J. S., Li, Z., & Park, H. S. (2018). 3d semantic trajectory reconstruction from 3d pixel continuum. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. (pp. 5060–5069).

Zhang, H., Hasith, K., & Wang, H. (2019). Gmc: Grid based motion clustering in dynamic environment. In *Proceedings of SAI Intelligent Systems Conference*.

Zhang, Z. (1998). Determining the epipolar geometry and its uncertainty: A review. *International Journal on Computer Vision (IJCV)*, *27*(2), pp. 161–195.

Appendix

归一化 SGD. 我们使用 NSGD (Bian et al. 2019)测量两个基本矩阵 (FMs) 之间的几何距离，这是 SGD error (Zhang 1998)的扩展。该方法使用两个模型生成虚拟匹配，并交叉计算与另一个模型生成的对极线的对应距离。结果是对称的，即，从 F1 到 F2 的误差等于从 F2 到 F1 的误差。为了在不同分辨率的图像中进行泛化，我们通过划分图像对角线的长度将误差重新缩放到范围（0,1)。

FM Ground truth. 图像对之间的基本矩阵可以从相机的固有和非固有参数得出。TUM 和 KITTI 数据集提供了相机参数的 Ground Truth，而 T&T 和 CPC 数据集则没有这些数据。我们通过使用 COLMAP (Schonberger & Frahm 2016)库重建图像序列来为后者推导相机参数的 Ground Truth，就像 (Ranftl & Koltun 2018; Yi et al. 2018)中的一样。请注意，SfM 的 pipeline 会在全局考虑 3D 点和相机的一致性，得到比较准确的估算值，且平均重投影误差低于一个像素。