

指导教师：_____ 杨涛 _____

提交时间：_____ 2016/3/21 _____

CVPR2015 Paper

Translation

No: _____ 01 _____

姓名：_____ 林明哲 _____

学号：_____ 2013302552 _____

班号：_____ 10011304 _____



基于树搜索的高效全局最优一致性最大化

Tat-Jun Chin *, Pulak Purkait *, Anders Eriksson † and David Suter *

School of Computer Science, The University of Adelaide *

School of Electrical Engineering and Computer Science, Queensland University of Technology

引言

最大一致性是计算机视觉中最常用的稳定性估计的标准之一。尽管它被广泛使用，但优化准则还是习惯通过随机抽样和测试技术，这是不能保证结果最优性的。虽然有几个全局最优算法存在，但他们计算太过缓慢而无法挑战随机方法的主导地位。我们的目标是通过提出一种高效的全局最大一致性算法来改变这个现状。在框架下 LP 型方法，我们展示了将最大一致性运用到各种各样的视觉任务中可以构成一种树搜索问题。这一观点引出了一种基于 a^* 搜索的新算法。我们打算通过有效的启发式函数和支持组更新程序，使 A^* 搜索快速找到全局最优的结果。关于一般的估计问题，我们的算法比以前的精确算法快了几个数量级。我们的研究确定了全局最优一致性最大化是一个有前景的解决方案。

1. 简介

最大一致性是计算机视觉中最常用的稳定性估计的标准之一。给定一组测量值 $\mathcal{X} = \{\mathbf{x}_i\}_{i=1}^N$ ，该标准的目的

是找到估计 θ 尽可能多的数据达到到阈值

$$\begin{aligned} \max_{\theta, \mathcal{I} \subseteq \mathcal{X}} & |\mathcal{I}| \\ \text{subject to} & r_i(\theta) \leq \epsilon \quad \forall \mathbf{x}_i \in \mathcal{I}, \end{aligned} \quad (1)$$

$r_i(\theta)$ 是 \mathbf{x} 的残差。例如，在三角测量时我们希望在 N 个视图中评估 3 维的点 θ ，其中 \mathcal{X} 包含该点的 2 维观察要点。残差 $r_i(\theta)$ 是在第 i 个视图中二次投影错误。

RANSAC 算法一直是占主导地位的方法。RANSAC 基本思想描述如下：

- ① 考虑一个最小抽样集的势为 n 的模型 (n 为初始化模型参数所需的最小样本数) 和一个样本集 P ，集合 P 的样本数 $\#(P) > n$ ，从 P 中随机抽取包含 n 个样本的 P 的子集 S 初始化模型 M ；
- ② 余集 $S^c = P \setminus S$ 中与模型 M 的误差小于某一设定阈值 t 的样本集以及 S 构成 S^* 。 S^* 认为是内点集，它们构成 S 的一致集；
- ③ 若 $\#(S^*) \geq N$ ，认为得到正确的模型参数，并利用集 S^* (inliers: 正确的数据) 采用最小二乘等方法重新计算新的模型 M^* ；重新随机抽取新的 S ，重复以上过程。

④在完成一定的抽样次数后,若未找到一致集则算法失败,否则选取抽样后得到的最大一致集判断内外点,算法结束。

这些随机方法是包括 RANSAC 算法的主要缺点是无法获得绝对最优的确定性解决方案,或者无法确认它是否是一个令人满意的近似解。一个不太公认的事实是即使对所有的子集进行检查,我们可能无法找到全局最优解,因为 θ^* 一般不对应于 p-tuple(见下文)的估计。

为了精确地解决这些问题,一些作者提出了基于分支限界法的方法。然而分支限界法通常相当的慢,特别当 θ 是高维数据时。事实上,RANSAC 应该用以预处理数据(可能导致真正的正确数据被丢弃),再调用分支限界法完善结果。更重要的是,分支限界法特别适合处理线性残差。在许多计算机视觉问题中,需要线性化 $r_i(\theta)$ 并且采用非几何意义的代数残差。

由于较大的计算费用,目前的全局可行性方法无法和 RANSAC 及其它的变体竞争。在这篇论文中,我们在准确、高效地解决问题的过程中取得显著进展。用线性规划的框架[18, 14],我们展示了如何将最大化一致性问题转换为一个树搜索问题。然后,我们提出了一个基于 A* 算法搜索[8]来遍历树。类似于[15, 5]的想法,我们的目标是找到最优数据子集。而不是抽样或

列举子集,确定性算法定位最好的子集。A* 搜索确保只有一小部分可用的子集需要探索。尽管它的组合复杂,我们的算法在几个常见的估计问题中,比以前的精确方法快了了几个数量级。此外,我们的方法不需要线性化残差。

1.1 准备工作

最小化问题是建立在几何估计 [11, 10] 的背景下。任务是找到估计 θ 最小化最大残。

$$\min_{\theta} \max_i r_i(\theta). \quad (2)$$

虽然本质上是非鲁棒自它有效地配合 θ 应对异常值,它包含一个单一的(全局)最小值, $R_i(\theta)$ 是严格拟凸和数据是非退化的。基于二分法和其他方案的算法已经被提出来用以解决该问题。

由于最大的残差由离群贡献,我们可以构造一个异常值去除方案,其中为第一解决和对应于最大残差数据被除去[19, 16, 20]。在适应和移除步骤可以迭代进行直到低于阈值的最大误差? 没有保证,然而,该算法其余的数据是最大的一致集,因为真正的正确数据也能在可迭代过程中移除。

如果,如果我们最大最小化 j , 而不是最大的 j 剩余 ($j \leq N$), 可以忽略异常

值和一个健壮的估计。然而, 现在问题是因为存在多个局部最小值而更加困难。Ke 和 Kanade 扩展他们的算法大约减少 j 最大残留。Matou sek 提出一个方法来检查所有的极小值问题找到全局最小值。李率先使用 Matou sek 进行鲁棒三角测量的方法。它是可证明的, 那里的时间复杂度 Matou sek 的过程是一个阶多项式的离群值。然而, 传统规模的问题搜索慢得令人痛苦。

一般来说, 提前知道正确的 j 的价值可以是不可忽视的。另一方面是正确数据的阈值设置问题? 例如, 重投影误差集的三角剖分是通常在几个像素内。

2. 背景

2.1. 可解问题

我们首先定义的类型的问题, 可以解决的最大一致性算法。我们要求残差函数 $R I (\theta)$ 是拟凸函数。这包含了许多常见的算法。例子如下

1. 线性回归
2. 三角测量
3. 单应矩阵拟合

2.2. LP-type 问题

LP-type 问题可以看作是广义线性规划问题。极大极小问题, 如他们是紧密联系的。特别是, 它建立了极大极小的

一组伪凸函数是一个 LP-type 问题 [2, 6]。这意味着几个关键属性的满意度 [1]。

首先, 我们定义 $f(X)$ 的解决方案 (极大极小问题的最小客观价值) 数据 X , 和 $\theta(X)$ 相应的全局最优估计。

特性 1, 单调性

特性 2, 局部性

这个概念的基础是线性规划类型问题的积分。

定义 1 (基础) A 是根据 B 的一个子集每一个适当的子集都有一个很小的值相对于 B 本身

定义 2 (组合维度) 组合维度是一个基础的最大可能大小。为 θ 在于 p 维域伪凸问题, 组合维度是 $p + 1$

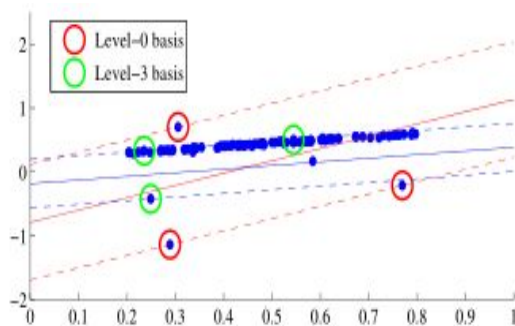
定义 3 (违反集, 覆盖范围和水平) 违反集 $V(B)$ 的基础上, $B \subseteq x$ 包含 X 的数据当对其进行评价时, 其残差大于 $\theta(B)$ 。

定义 4 (支持集) 0 级的数据集 X 基础被叫做 X 的支持集。

定义 5 (基础邻接)

特性 3 (基础路径的存在) 存在一个路径从 0 基每个 level- k 基 ($k > 0$) 相邻的基。

上述属性意味着该集合的基可以在一个树结构中，有 2 个碱基被连接在一起如果他们相邻的边缘。Matou sek 建议生成树到第 k 个水平（通过删除数据从基和将在未来获得相邻基水平），这样所有的 k 级基地可以检查；图 2 (a) 说明。此外，他还证明了 k 级基地数量的上界 $O((k+1)P)$ 。然而，在实践中，基的数量可以是非常高的，和方法只适用于少数的异常值（1 或 2）



3. 一致性最大化的树搜索

我们现在已经准备好描述我们的新算法如何解决一致性最大化问题

定义 6 (可行性)

我们可以重写如下相关问题

$$\min_B l(B), \quad \text{s.t. } f(B) \leq \epsilon. \quad (8)$$

换句话说，我们寻求的最低水平的基础，是可行的。给出了一个最优的 B^* ，因此，一致性最大化集 L 就是 $C(B^*)$ 。

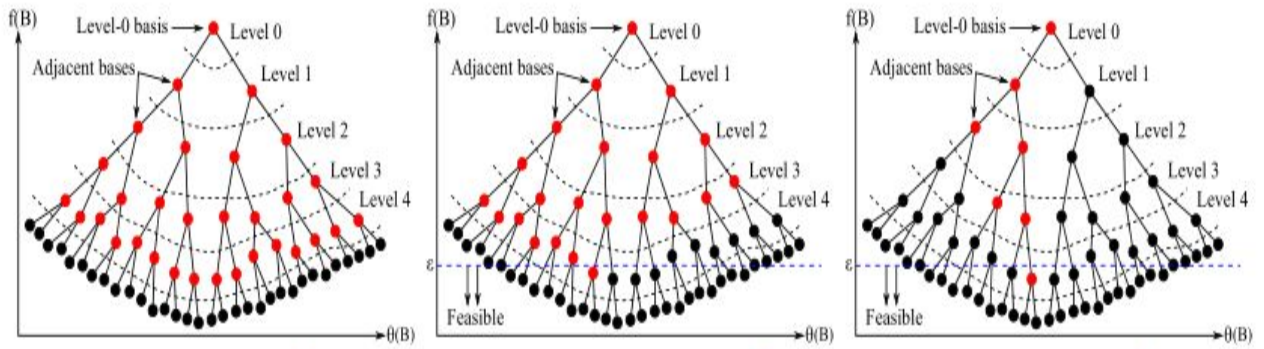
为了获得 B^* ，我们可以采用广度优先搜索 (BFS) 遍历 Matou sek 的树结构，即测试的可行性和在一个特定的水平扩展（生成相邻基础），在一个特定的水平，在进入下一阶段之前。这样的搜索机制保证第一个可行的基础上发现是最浅的(最低级别)。图 2(b) 描述了这个想法。

假设 B^* 发生在水平的 K 与同解 K ，BFS 算法并不试图生成所有 k 级基。但是，它仍然必须生成所有的基础在以前的水平，因此在实践中高炉效率不高。然而，相当大的改进基本的计划，可以作出，正如我们在下面的描述。为便于参考，图 1 总结了我们的方法。

3.1 避免重复的基集

从根节点开始(0 基础), 有多条路径到达 level-k 基础相同。对应于这一事实, 只要同一组数据删除, 删除的顺序并不重要。重要的冗余存在于树中。这一特点 Matou sek 的原始方法并不存在。请注意, 图 2 中的树, 冗余已经被移除。

在我们的算法中，一个基集 B 如果之前被发现的话，它就被忽略了



(a) Matoušek's method for solving (6) with $k = 4$. (b) Breadth-first search for solving (8). (c) A* search for solving (8).

Figure 2. Tree traversal for solving (6) (panel a) and (8) (panels b and c) on a problem with $p = 1$. The ordinate is $f(\mathcal{B})$ while the abscissa is $\theta(\mathcal{B})$. Nodes in red are bases expanded in the respective algorithms. In Matoušek's method, all level- k bases (here, $k = 4$) must be tested, since there are multiple local minima. In panels b and c, the two different globally optimal solutions found are equally good w.r.t. (8).

Algorithm 1 A* search for consensus maximisation

- 1: $\mathcal{B} \leftarrow$ Support set of \mathcal{X} .
- 2: Insert \mathcal{B} with priority $e(\mathcal{B})$ into queue q .
- 3: Initialise hash table T to null.
- 4: **while** q is not empty **do**
- 5: Retrieve from q the \mathcal{B} with the lowest $e(\mathcal{B})$.
- 6: **if** $f(\mathcal{B}) \leq \epsilon$ **then**
- 7: Exit with $\mathcal{C}(\mathcal{B})$ as the maximum consensus set.
- 8: **end if**
- 9: **for each** $x \in \mathcal{B}$ **do**
- 10: **if** indices of $\mathcal{V}(\mathcal{B}) \cup \{x\}$ do not exist in T **then**
- 11: Hash indices of $\mathcal{V}(\mathcal{B}) \cup \{x\}$ into T .
- 12: $\mathcal{B}' \leftarrow$ Support set of $[\mathcal{C}(\mathcal{B}) \setminus \{x\}]$.
- 13: Insert \mathcal{B}' with priority $e(\mathcal{B}')$ into q .
- 14: **end if**
- 15: **end for**
- 16: **end while**
- 17: Return error (no feasible solution).

不断重复检测。例如，设定的 $\mathcal{X} = \{\times 71, \times 10, \times 5, \times 4, \times 28\}$ 转换为字符串 0405102871 和哈希散列。这简单的想法带来了显着的计算消耗的节省。

3.2 带有 A*算法的启发式搜索

我们的方法是基于 A* 算法, 这是一个启发式搜索方法广泛用于路径查找 [8]。

当 BFS 队列中选择最优的解集进行测试和扩大时, A* 选择最低的基础评估值 e (图一步骤 5), 这被定义为

$$e(\mathcal{B}) = l(\mathcal{B}) + h(\mathcal{B}), \quad (9)$$

h 是一个启发式的地方。直观地说, $h(\mathcal{B})$ 估计的数据从 $\mathcal{C}(\mathcal{B})$ 使其可行。解集 e 是最优的, 接近可行性较低。据 e 从而加速扩大搜索; 见图 2(c)。请注意, 如果对所有 $\mathcal{B} h(\mathcal{B}) = 0$, BFS *减少。

由于有不同的覆盖两个平等的解集是有可能的, 两个基地 \mathcal{B}_1 和 \mathcal{B}_2 的声明相同只有 $\mathcal{C}(\mathcal{B}_1) = \mathcal{C}(\mathcal{B}_2)$, 或等价地, 如果 $\mathcal{V}(\mathcal{B}_1) = \mathcal{V}(\mathcal{B}_2)$ 。我们提出了哈希散列的违反集的支持集。让 $Q \subseteq X$ 的任意子集。各项指标在问题的数据被排序, 得到一个整数序列。我们将整数 ASCII 字符和连接他们变成一个字符串。字符串然后散列到一个表通过标准技术, 以方便

定义7(可容许性)启发式h是容许的如果

$$h(B) \geq 0 \text{ and } h(B) \leq h^*(B), \quad (10)$$

在 $h^*(B)$ 是必须的最小数量的数据从 $C(B)$ 使数据可行。暗示了上面的是 $h(B) = 0$ 如果 B 是可行的。注意, 计算 $h^*(B)$ 精确的解决最初的最大一致性 问题 $C(B)$ 。

定理一 在启发函数 h 条件下, A^* 是最 优解。

证明如下[8], 我们需要显示在队列 中存在一个次优的可行的基础上 B^\dagger 时, 它不会选择和测试之前, 一个最 佳的基础上 A^* 存在于 B 队列。设 B 是另 一个基础的队列在通往 B^* 路径。以下 可以成立

$$\begin{aligned} e(B^\dagger) &= l(B^\dagger) && \text{(since } B^\dagger \text{ is feasible)} \\ &> l(B^*) && \text{(since } B^\dagger \text{ is suboptimal)} \\ &= l(B) + h^*(B) && \text{(since } B \text{ leads to } B^*) \\ &\geq l(B) + h(B). && \text{(since } h \text{ is admissible)} \end{aligned} \quad (11)$$

上述结论表示, 基于 B^* 的 B 方法总是 优先于 B^\dagger 被测试。因此 A 总是能先 于 B^\dagger 找到 B^* 。

3.3 启发式的一致性最大化

以前我们的启发式是受异常值删除重 新插入技术[20]19日16日但重要的修

改。我们叫我们的启发式 h_{ins} 和证明 A^* 适用。计算 $h_{ins}(B)$, 让

$$O = \{B_1, B_2, \dots, B_M\}, \quad f(B_m) > \epsilon \quad \forall m \quad (12)$$

包含必须递归序列集的支持 $C(B)$, $C(B)$ 可行的

$$B_m = \text{Support set of } [C(B) \setminus \{B_1, \dots, B_{m-1}\}] \text{ for } m > 1$$

$$F = C(B) \setminus O, \quad \text{where } f(F) \leq \epsilon. \quad (13)$$

然后, 我们试图插入的个人数据从一 个由一个到 F 。如果插入使不可行的, 支持组扩大 F 被启发式 $h_{ins}(B)$ 是 一个递增的; 否则如果扩大 F 仍然是 可行的, 插入是永久的没有启发式变 化。算法 2 给出了一个正式的描述。

定理 2 h_{ins} 是可容许的

证明了 h_{ins} 是负的是显而易见的。 让 $C(B) = F^* \cup O$, F^* 是最大的可行的 $C(B)$ 的子集。 O^* 因此必须从最小的子 集 $C(B)$ 实现的可行性, 即 $h^*(B) = |O^*|$ 。我们称 F^* 为“真正的”正确值和 异常值。从 O 让 x 是一个基准, O B 组 ($F \cup O$ 的支持 $\{x\}$)。如果 $B \in O$ 是不可行, 然后通过属性 1(单调性)

$$f(F^* \cup B') \geq f(B') > \epsilon, \quad (14)$$

$B \in O$ 必须至少包含一个真实的异常值。 除去的 $B \in O \setminus \{x\}$ 从而消除至少一个真

正的异常 $F \cup \{x\}$ (注意剩余集 $F \leftarrow F \cup \{x\} \setminus B_0$ 总是可行的。因此, 在大多数 $|0^*|$ 可行支持设置清除。由于每一个删除增量 $(B) 1$, 其最终值不能大于 $H^*(B)$ 。注意, 在一般情况下 0^* 不是 0 的子集并且上面的证明也假设它不是。

3.3 提前终止

$g(B)$ 定义为一个上界的数量数据必须从 $C(B)$, $C(B)$ 的可行性。再加上启发式函数, 我们可以因此得到

$$h(B) \leq h^*(B) \leq g(B). \quad (15)$$

鉴于 $g(B)$, 有可能提前终止算法一。具体来说, 如果 $h(B) = g(B)$ 目前正在测试中 B 都是可行的。最大一致性集 L^* 在 $F(13)$ 中定义。

Algorithm 2 Calculation of $h_{ins}(B)$.

```

1: If  $f(B) \leq \epsilon$ , return 0.
2:  $\mathcal{O} \leftarrow \emptyset$ .
3: while  $f(B) > \epsilon$  do
4:    $\mathcal{O} \leftarrow \mathcal{O} \cup B$ .
5:    $B \leftarrow$  Support set of  $[C(B) \setminus B]$ .
6: end while
7:  $h_{ins} \leftarrow 0$ ,  $\mathcal{F} \leftarrow C(B)$ .
8: for each  $B \in \mathcal{O}$  do
9:   for each  $x \in B$  do
10:     $B' \leftarrow$  Support set of  $[\mathcal{F} \cup \{x\}]$ .
11:    if  $f(B') \leq \epsilon$  then
12:       $\mathcal{F} \leftarrow \mathcal{F} \cup \{x\}$ .
13:    else
14:       $h_{ins} \leftarrow h_{ins} + 1$ .
15:       $\mathcal{F} \leftarrow \mathcal{F} \cup \{x\} \setminus B'$ .
16:    end if
17:   end for
18: end for
19: Return  $h_{ins}$ .
```

定理三 A*算法是最优的提前终止

证明: 权重由下给出

$$e^*(B) = l(B) + h^*(B) \quad (16)$$

是必须除去的最小数目的数据为达到可行性, 鉴于我们已取消了 $V_B(2)$ 。再与 (9) 比较, 很明显, $E(B) \leq E^*(B)$ 。如果当前 B $h(B) = g(B)$, 然后从 (15), $e(B) = e^*(B)$ 。因为我们根据 A*算法, $e(B) \leq 0$ $e(B(0))$, B 是其他任何依据队列。因此 $e^*(B) \leq e(B(0)) \leq e^*(B(0))$, 即的路径, 最优的可行的解集必须通过 B 。

我们建议以下上界

$$g_{rem}(B) = |C(B) \setminus \mathcal{F}| = |\mathcal{O}|, \quad (17)$$

可以算是算法二的附加品

定理三 $g_{rem}(B) \geq h^*(B)$.

证明：由于 F 是可行的，然后 $|F| \leq |F^*|$ f^* 是最大的可行子集。这意味着 $|g_{rem}(B)| \geq |0^*| = H^*(B)$ ，其中 $0^* = C(B) \setminus F^*$

4. 有效支持集的更新

很明显，我们的算法的整体效率取决于有效的计算支持集。这相当于解决极大极小问题的相关数据子集。幸运的是，除了算法 1 的初始化步骤 1 中需要解决的问题在所有 X ，从头设置计算仅仅需要其他支持更新一个已知的支持后，少量的数据删除（在算法 1 步骤 12，在算法 2 步骤 5）或插入（在算法 2 步骤 10）。因此，我们的方法可以很快如果更新程序实现。

可以用来计算各种方案集的支持为伪凸残差。普遍 LP，二分或伪凸编程。事实上，非线性优化（迭代）可以用来直接寻求全局最优的解决方案。为此，重写作为约束非线性和解决问题。

$$\min_{\theta, \gamma} \gamma, \quad \text{s.t. } r_i(\theta) \leq \gamma, \quad (18)$$

删除最大正确数据的位置。请注意某些应用程序，附加的限制，必须强加于 θ ，例如，cheirality 三角（4）。

给出解决方 θ^* 和 γ^* ，支持设置可获得的数剩余的 θ^* 等于 γ^* 。在我们的工作中，我们使用 MATLSQP 算法 fmincon 求解（18）。

我们建立一个高效的支持集更新程序为基础的在 SQP 算法。设 P 和 Q 为两类 X ，在 $\theta(P)$ 知道我们想更新它获得 $\theta(Q)$ 。还有，从磷的去除或增加磷，得到的少量的数据；因此我们预期 $\theta(Q)$ 接

$\theta(P)$ 。我们的更新程序是简单的“热启动”与 $\theta(P)$ 在解决（18）对问题给出了解决方 $\theta(Q)$ ，集 Q 的支持可以“读出”从残差。在实践中，这种简单的更新方案是非常有效的更大的效率可以通过专门某些类型的残差的更新例程。线性回归（3），我们采用众所周知的顶点到顶点下降算法（见 [4, 2 章]）为切比雪夫回归（ L_∞ 最小化），这是非常快的。

5. 结论

我们进行了许多实验在几个常见的估计问题。我们关注比较算法对全局最优方法(*) 稳健估计。分位数回归量(7)，我们测试了 Matou sek 的方法 [14]。的最大一致性估计值(1)，我们测试了 BFS(秒。3)，BnBmaxf 配方 4[21]，BnB 与双线性公式(比邻)5[13]。RANSAC 也作为一个基准运行。

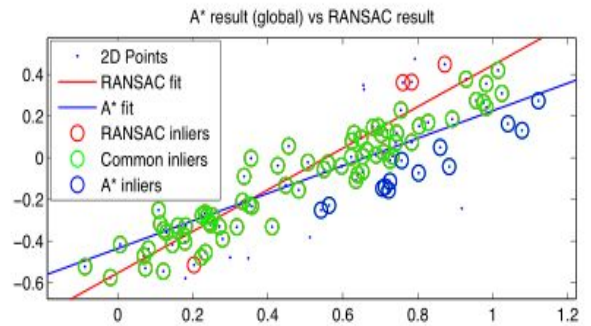
我们将我们的结果分成两类:线性回归(3)和伪凸残差的问题。请注意 maxf 和比邻只适用于使直线化残差。

一个完全的 Matlab 实现:支持集更新利用 fmincon 的拟凸的情况进行了,而顶点到顶点的算法,自己编程。这些例程也被用于 Matou sek 和 BFS。maxfs 需要求解混合整数线性规划

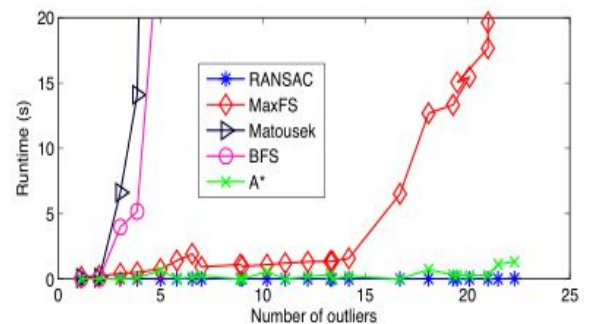
(MILP) 问题,为此我们使用国家的最先进的 gurobi 求解。为 LP 的子程序比邻,我们使用了高效 MOSEK 求解。

RANSAC, 标准的停止准则[7]是用信心 $\rho = 0.99$ 。然而,如果标准已经满足了 100 次迭代之前,我们不得 RANSAC 继续直到 100 次迭代。另一方面一方面,如果对问题的独特 p-subsets 数小于 100,列举并测试所有的 p-subsets。

所有的实验运行在一个 2.70GHz 主频处理器和 8 的内存的系统环境下。



(a) Typical data instance and lines fitted by RANSAC and A*. In this data instance, RANSAC found 76 inliers while A* found 87 inliers.



(b) Average runtime of different methods against number of outliers.

Figure 3. Line fitting results.

5.1 线性回归

直线拟合:我们生成 $N = 100$ 点在一 条直线在高斯噪声和 $\sigma = 0.1$ 。许多大的高斯点被破坏噪音 ($\sigma = 1$) 模拟离群值。正确数据阈值(1)被选为 $E = 0.3$, 和 $k(7)$ 设置为真正的离群值后(解决 (1) 准确)。

图 3(一个)显示了一个典型的数据实例和结果。内围层发现 RANSAC 用红

笔圈出来的, 而这些发现 A*用蓝色圈。点绿色的圆圈是内围层发现 RANSAC 和 A*。正如所料, RANSAC 发现一个次优的解决方案与 76 个正确解, 而发现全局最优解 87 个。

为了使直线化单应性我们单应性测试评估使用线性回归, 维度 θ 是 8 (见 [9, 第四章] 使直线化单应性约束)。在 5.2 秒。我们将结果单应性使用几何 (伪凸) 残差估计。注意, 每个关键点匹配产生两个剩余函数; 因此在实际的优化中数据 N 和离群值 k 的

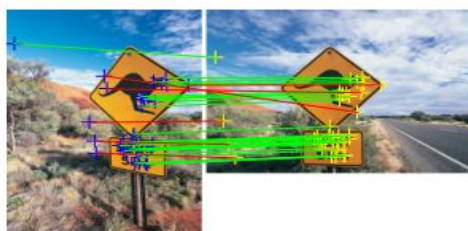
Methods	Linearised homography estimation						Linearised fundamental matrix estimation			
	Road Sign		Cambridge		Keble College		Valbonne Church		Merton College III	
	NSub	Runtime (s)	NSub	Runtime (s)	NSub	Runtime (s)	NSub	Runtime (s)	NSub	Runtime (s)
RANSAC	100	0.0903	100	0.1130	100	0.3689	100	0.0417	100	0.0448
MaxFS	67548	1.0433	65118	1.8661	1976336	> 3600	16267	9.2132	15515	5.7162
BILIN	2526	> 3600	1252	> 3600	8	> 3600	2220	> 3600	140	> 3600
Matoušek	220743	> 3600	189901	> 3600	179353	> 3600	238006	> 3600	237124	> 3600
BFS	166852	> 3600	143750	> 3600	137062	> 3600	71178	525.2548	194183	3234.6948
A*	911	1.2936	1473	2.2619	560	2.9121	1838	1.3922	589	0.6458
	$N = 36, p = 8, \epsilon = 0.1$ $ \mathcal{I}_R = 29.67, \mathcal{I}^* = 30$		$N = 55, p = 8, \epsilon = 0.1$ $ \mathcal{I}_R = 44.80, \mathcal{I}^* = 47$		$N = 299, p = 8, \epsilon = 0.1$ $ \mathcal{I}_R = 282.74, \mathcal{I}^* = 284$		$N = 58, p = 8, \epsilon = 0.1$ $ \mathcal{I}_R = 43.66, \mathcal{I}^* = 52$		$N = 187, p = 8, \epsilon = 0.1$ $ \mathcal{I}_R = 176.69, \mathcal{I}^* = 181$	

N: data size, p: number of parameters, ϵ : inlier threshold
 $|\mathcal{I}_R|$: average consensus size of RANSAC, $|\mathcal{I}^*|$: optimal consensus size, NSub: number of subproblems solved

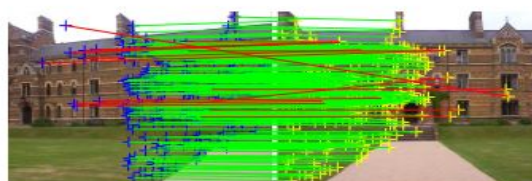
Table 1. Results for linearised homography estimation and linearised fundamental matrix estimation.

我们每次都有数量变化的点被损坏为异常值, 并调用所有的比较的方法。每个方法运行 100 次, 平均运行时间被记录下来。需要注意的是, 除了 RANSAC 方法确定, 给出相同的回归结果。图 3 (2) 显示了不同方法对异常值的平均消耗时间 (再次, 这个数字是已知的 (1) 是解决的)。据预测, Matoušek 和 BFS 规模极其严重的异常点的数量, 因为较大的树的深度需要探讨。maxfs 也迈不好, 异常值的个数。我们使用二维作图函数因为它需要超过 120 秒甚至 1 的异常性。A*比 RANSAC 只能缓慢但大大优于其他方法。一个 A*的缩放属性也更好。我们观察到, RANSAC 产生 97%的运行中的次优解。

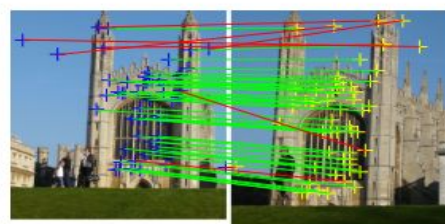
数量翻了一番 (有关详细信息, 请参阅 [9, 第四章])。



(a) Road Sign (image index 1 and 2)



(b) Keble College (image index 2 and 3)



(c) Cambridge

Methods	Homography estimation (pseudoconvex residual)						Triangulation (pseudoconvex residual)			
	Road Sign		Cambridge		Keble College		Cathedral		House	
	NSub	Runtime (s)	NSub	Runtime (s)	NSub	Runtime (s)	NSub	Runtime (s)	NSub	Runtime (s)
RANSAC	100	0.0815	100	0.0824	100	0.1371	35	2.5845	56	4.1530
Matoušek	54921	1909.1582	34187	> 3600	12912	> 3600	280	12.8718	427	18.0139
BFS	8530	747.3163	37368	> 3600	14104	> 3600	119	5.0947	151	5.9715
A*	688	120.9280	1599	262.2883	56	19.7666	136	5.5920	281	8.7528
$N = 36, p = 8, \epsilon = 0.1$ $N = 55, p = 8, \epsilon = 0.1$ $N = 299, p = 8, \epsilon = 0.1$ $N = 7, p = 3, \epsilon = 0.01$ $N = 8, p = 3, \epsilon = 0.01$ $ \mathcal{I}_R = 29.85, \mathcal{I}^* = 30$ $ \mathcal{I}_R = 44.70, \mathcal{I}^* = 47$ $ \mathcal{I}_R = 282.47, \mathcal{I}^* = 284$ $ \mathcal{I}_R = 1, \mathcal{I}^* = 2$ $ \mathcal{I}_R = 3, \mathcal{I}^* = 4$										
N: data size, p: number of parameters, ϵ : inlier threshold $ \mathcal{I}_R $: average consensus size of RANSAC, $ \mathcal{I}^* $: optimal consensus size, NSub: number of subproblems solved										

Table 2. Results for homography estimation and triangulation with pseudoconvex residuals.

我们测试了三对图像，第一对采用的仍是传统的鲁棒性几何估计：路标（图片索引 1 和 2）[16]，Keble 大学（图片索引 2 和 3），和剑桥（下载自 Flickr）；见图 4。SIFT 算法配合使用 VLFeat 工具箱进行了关键点匹配，劳氏第二近邻测试调用删除匹配。关键点在线性化约束之前进行正常化处理。正确数据阈值 E 选为 $E = 0.1$ ， k 是作为真正的离群值计数。表 1 展示了所有采用 RANSAC 方法的结果（平均运行超过 100 次）。除了报告实际运行用时，我们还展示了图 4。基于 A* 的单应矩阵估计结果在三对图像中展示出来。在绿色和红色的线分别表示检测到的正确数据和异常的 SIFT 特征点匹配。

根据子问题的数量级，可以总结为以下这些方法。

1. RANSAC
2. MaxFS
3. BILIN
4. Matoušek, BFS and A*

5.2 拟凸残差

单应矩阵估计：我们使用相同数据集

进行线性化的单应矩阵估计评估实验。正确数据阈值 E 设为 0.1 像素，但由于 (5) 使用伪凸剩余函数，最大一致性解集与线性化的结果并不相同。结果如表 2 所示。很明显，在线性化情况下即使支持集由于伪凸残差更新慢，BFS 算法和那 Matoušek 算法都变得更快。这背后的原因是，毫无疑问的，有效的数据大小和离群值数量的翻倍取决于单应性直线化约束 (NSub 值在表 1) 高得多。尽管在这个实验中 A* 速度是较慢的，但还是比 BFS 和 Matoušek 快得多。请注意，A* 在几十秒内成功返回全局最优估计。

6. 总结

我们提出了一种有效的全局最优的最大一致性算法，是基于 A* 搜索的算法。提出了一种启发式函数并且被证明是和 A* 配合使用的。实验表明，我们的新方法运行速度较于以前的精确算法快了几个数量级。我们的实验确定了一个有前景的方向来完全解决最大一致性问题。A* 的良好性能依赖于合适的启发式函数 h 及其下界 h^* 。我们的方法通过为目标应用程序进一步构建专门的启发式函数还可以挖掘出更多的在速度提高上的潜力。

引用:

[1] http://en.wikipedia.org/wiki/LP-type_problem. 2

[2] N. Amenta, M. Bern, and D. Eppstein. Optimal point placement for mesh smoothing. In SODA, 1997. 2

[3] B. Chazelle and J. Matoušek. On linear-time deterministic algorithms for optimization problems in fixed dimensions. In Symp. Discrete Algorithms, 1993. 6

[4] E. W. Cheney. Introduction to approximation theory. McGraw-Hill, 1966. 6

[5] O. Enqvist, E. Ask, F. Kahl, and K. Åström. Robust fitting for multiple view geometry. In ECCV, 2012. 1, 8

[6] D. Eppstein. Quasiconvex programming. Combinatorial and Computational Geometry, 25, 2005. 2, 3, 6

[7] M. A. Fischler and R. C. Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. Comm. of the ACM, 24(6):381–395, 1981. 1, 6

[8] P. E. Hart, N. J. Nilsson, and B. Raphael. A formal basis

for the heuristic determination of minimum cost paths. IEEE Trans. on Systems Science and Cybernetics, 4(2):100–107, 1968. 1, 4

[9] R. Hartley and A. Zisserman. Multiple view geometry in computer vision. Cambridge University Press, 2nd edition, 2004. 7, 8

[10] F. Kahl. Multiple view geometry and the l_∞ -norm. In ICCV, 2005. 2, 6

[11] Q. Ke and T. Kanade. Quasiconvex optimization for robust geometric reconstruction. In ICCV, 2005. 2, 6

[12] H. Li. A practical algorithm for l_∞ triangulation with outliers. In CVPR, 2007. 2

[13] H. Li. Consensus set maximization with guaranteed global optimality for robust geometry estimation. In ICCV, 2009. 1, 6

[14] J. Matoušek. On geometric optimization with few violated constraints. Discrete and computational geometry, 14(1):365–384, 1995. 1, 2, 3, 6

[15] C. Olsson, O. Enqvist, and F.

Kahl. A polynomial-time bound for matching and registration with outliers. In CVPR,

2008. 1, 8

[16] C. Olsson, A. Eriksson, and R. Hartley. Outlier removal using duality. In CVPR, 2010. 2, 5, 7, 8

[17] C. Olsson, A. Eriksson, and F. Kahl. Efficient optimization for l_1 -problems using pseudoconvexity. In ICCV, 2007. 2, 6

[18] M. Sharir and E. Welzl. A combinatorial bound for linear programming and related problems. In STACS, 1992. 1, 2

[19] K. Sim and R. Hartley. Removing outliers using the l_1 norm. In CVPR, 2006. 2, 5

[20] J. Yu, A. Eriksson, T.-J. Chin, and D. Suter. An adversarial optimization approach to efficient outlier removal. In ICCV, 2011. 2, 5

[21] Y. Zheng, S. Sugimoto, and M. Okutomi. Deterministically maximizing feasible subsystems for robust model fitting with unit norm constraints. In CVPR, 2011. 1, 6