**Deep Rectangling for Image Stitching: A Learning Baseline**

代码：<https://github.com/nie-lang/DeepRectangling>

## 摘要

拼接图像提供了宽视场 (FoV)，但存在令人不快的**不规则边界**。为了解决这个问题，现有的**图像矩形化**方法通过两个阶段来致力于搜索初始网格和优化目标网格来形成网格变形。然后可以通过warp拼接图像来生成矩形图像。然而，这些解决方案**仅适用于具有丰富线性结构的图像**，导致具有非线性对象的人物和风景明显失真。

在本文中，我们通过提出**第一个图像矩形的深度学习解决方案**来解决这些问题。具体来说，我们预先定义了一个刚性目标网格，并且只估计一个初始网格以形成网格变形，从而有助于一个紧凑的单阶段解决方案。使用具有**残差渐进回归策略**的全卷积网络预测初始网格。为了获得具有高内容保真度的结果，提出了一个综合目标函数，以同时鼓励边界矩形、网格形状保持和内容感知自然。此外，我们构建了第一个图像拼接矩形数据集，在不规则边界和场景中具有很大的多样性。实验证明了我们在数量和质量上都优于传统方法。

## 1. 介绍

图像拼接算法 [3, 13, 21, 30] 可以从正常 FoV 图像（图 1a）生成宽 FoV 图像（图 1b）。这些方法优化全局或局部扭曲以对齐不同图像的重叠区域。然而，非重叠区域总是受到不规则边界的影响[2]。使用图像拼接技术的人必须容忍不愉快的界限。

**为了处理不规则的边界，其中一种解决方案是用矩形裁剪拼接图像**。然而，裁剪不可避免地会降低拼接图像的 FoV，这与图像拼接的初衷相矛盾。图 1c 演示了一个示例，其中暗区域表示通过裁剪丢弃的区域。另一方面，图像补全可以将缺失区域合成为矩形图像。尽管如此，目前还没有工作为图像拼接中的不规则边界设计掩码，甚至 SOTA的图像补全工作 [26, 28] 在处理拼接图像时表现出不令人满意的性能（图 1d）。此外，补全方法可能会添加一些看似和谐但与现实不同的内容，使其在自动驾驶等高安全性应用中不可靠[12]。



Fig1 图像拼接中不同解决不规则边界的方案。(a) 一个经典的图像拼接数据集，不包含在建议的数据集中（APAP-conssite [30]）。 (b) 使用 UDIS [24] 拼接图像，具有不可避免的不规则边界。 (c) 裁剪边界以获得矩形图像。 (d) 使用 LaMa [26] 完成缺失区域。 (e) He 等人的矩形 [7] 有明显的扭曲。 (f) 我们的矩形具有高内容保真度。

为了获得具有高内容保真度的矩形图像，提出了图像矩形方法[6,7,14]，通过网格变形将拼接图像扭曲为矩形。然而，**这些解决方案只能保留具有直线/测地线的结构**，例如建筑物、盒子、柱子等。对于非线性结构，例如肖像 [27]，通常会产生失真。实际上，**保留线性结构的能力受到线检测的限制**，因此有时线性结构也会出现失真（图1e）。此外，这些传统方法是两阶段解决方案，依次搜索初始网格和优化目标网格，使得并行加速具有挑战性。

为了解决上述问题，我们提出了第一阶段学习基线，即我们预定义了一个刚性目标网格并仅预测一个初始网格。具体来说，我们设计了一个简单但有效的全卷积网络，以使用残差渐进回归策略从拼接图像中估计内容感知初始网格。此外，提出了一个**由边界项、网格项和内容项组成的综合目标函数，以同时鼓励边界矩形、网格形状保持和内容感知自然**。与现有方法相比，由于我们的内容约束中的有效语义感知，我们的内容保存能力更通用（不限于线性结构）和更健壮（图 1f）。

由于没有现成的合适数据集，我们构建了一个**深度图像矩形数据集（DIR-D）**来监督我们的训练。首先，我们将 He 等人的矩形 [7] 应用于真实拼接图像以生成合成矩形图像。然后我们**利用矩形变换的逆将真实矩形图像变形为合成拼接图像**。最后，我们从数以万计的合成图像中**严格地手动过滤出没有失真**的图像，经过几个时期，产生了一个**包含 6,358 个样本的数据集**。

实验结果表明，我们的方法可以有效地生成保留内容的矩形图像，在数量和质量上都优于现有解决方案。总而言之，我们将我们的贡献总结如下：

• 我们提出了第一个用于图像拼接的深度矩形解决方案，它可以有效地以残差渐进方式生成矩形图像。

• 现有方法是两阶段解决方案，而我们的解决方案是单阶段解决方案，可以使用预定义的刚性目标网格进行高效并行计算。此外，我们的可以保留线性和非线性结构。

• 由于没有合适的拼接图像和矩形图像对的数据集，我们构建了一个具有广泛不规则边界和场景的深度图像矩形数据集。

## 2. 相关工作

本文提供了图像拼接的一个基于矩形化的深度学习方法，因此，本节回顾了之前图像拼接和图像矩形化的相关工作。

### 2.1 图像拼接

对齐重叠区域[15]是图像拼接的核心目标。但是为了产生自然的拼接图像，还需要最小化非重叠区域的投影失真。在 [2, 19] 中，重叠区域的投影变换被平滑地外推到非重叠区域，并且由此产生的扭曲逐渐从图像中的投影变为相似。李等人[17]提出了一种准单应性扭曲，它依赖于全局单应性，同时挤压非重叠区域。Liao 和 Li [18] 提出了两个单透视扭曲，以保持透视一致性并减少投影失真。 最近，贾等人[10]考虑长线的场景并在拼接过程中保持全局共线线段的形状。

现有的图像拼接算法虽然可以减少投影畸变，保持自然外观，但无法解决拼接图像边界不规则的问题。

### 2.2 图像矩形

为了获得矩形拼接图像，He 等人[7] 提出优化保线网格变形。然而，所提出的能量函数只能保留线性结构。考虑到直线可能在全景图中弯曲（ERP 格式），Li 等人[14]将保留线性结构的能量项改进为测地线保持能量项。但是这种改进限制了它在全景图中的应用，并且不能直接从拼接图像中检测到测地线。后来，张等人[32]在统一优化中桥接图像矩形和图像拼接。尽管如此，为了减少最终矩形结果的失真，他们对矩形形状做出了妥协，改为采用分段矩形边界约束。

图像矩形很少被研究，因为不稳定的性能和大量的时间消耗使其在应用中不切实际。 在本文中，我们提出了一个简单但有效的学习基线来解决这些问题。

## 方法

首先在 3.1 节分析传统pipeline 和本文提出的 pipeline。接着分别在 3.2 和 3.3 节讨论本文方法的网络结构和目标函数。

### 3.1 Traditional Baseline vs. Learning Baseline

一个矩形方法应该可以解决初始化网格和目标网格从而形成网格变形的工作，然后可以通过 warp 得到矩形化结果。

3.1.1 Traditional Baseline

对于一个经典的传统方法[7,14]，一般是分为两阶段的：局部阶段和全局阶段（如图 2a 所示).

Stage 1:局部阶段。首先，在拼接图像中插入丰富的接缝，使用接缝雕刻算法 [1] 得到初步的矩形图像。然后，在初始矩形图像上放置一个规则网格并移除所有接缝以获得具有不规则边界的拼接图像的初始网格。

Stage 2：全局阶段。此阶段通过优化能量函数来解决最佳目标网格，以保留有限的感知属性，例如直线。

他们通过将拼接图像从初始网格扭曲到目标网格来生成矩形图像。

3.1.2 Learning Baseline

正如图 2b 所示，本文提出的学习基线是一种一阶段方法。

给定拼接图像，我们的解决方案只需要通过神经网络预测内容感知的初始网格。至于目标网格，我们将其预定义为刚性形状。此外，刚性网格形状可以很容易地使用矩阵计算来加速后向插值[23]。通过将拼接图像从预测的初始网格扭曲到预定义的目标网格，可以获得矩形图像。

与传统的基线相比，学习基线由于是一阶段的管道，效率更高。内容保留能力使我们的矩形结果在感知上更加自然（在第 3.3.1 节中解释）。

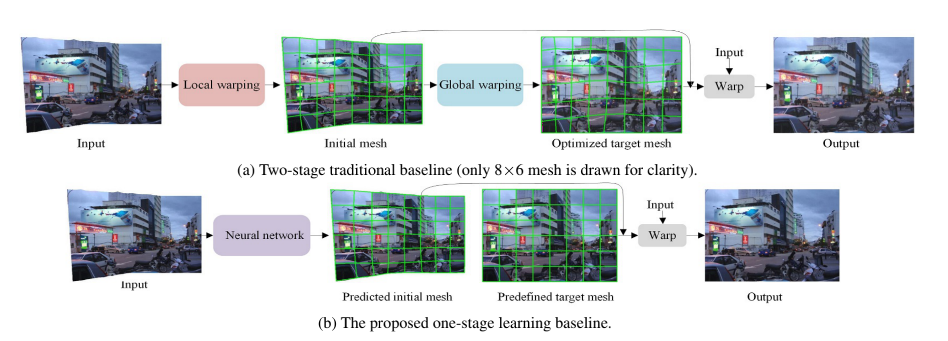


Fig2 传统基线与学习基线。传统的baseline通过搜索初始网格和连续优化目标网格分两个阶段解决矩形扭曲，而我们的解决方案是一个阶段解决它，因为我们只预测初始网格。

### 3.2 网络结构

与图像补全任务 [26, 28] 类似，缝合掩码也包含在建议网络的输入中。如图 3 所示，我们将通道维度上的拼接图像 I 和掩码 M 连接起来作为输入。输出是预测的网格运动。

**特征提取器**。我们堆叠简单的卷积池块以从输入中提取高级语义特征。 形式上，采用 8 个卷积层，其滤波器数量分别设置为 64、64、64、64、128、128、128 和 128。在第 2、第 4 和第 6 卷积层之后使用最大池化层。

**网格运动回归器**。 在特征提取之后，使用自适应池化层来固定特征图的分辨率。 随后，我们设计了一个全卷积结构作为网格运动回归器，以**基于常规网格预测每个顶点的水平和垂直运动**。假设网格分辨率为 U × V ，输出体积的大小为 (U + 1) × (V + 1) × 2。

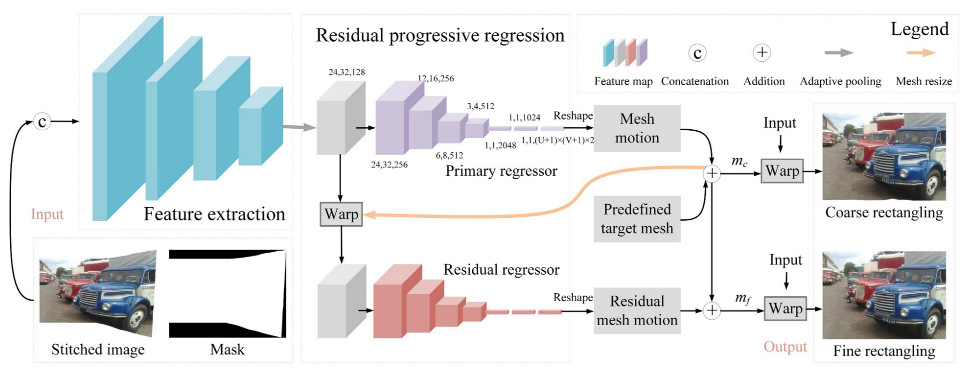


Fig3 我们网络的整体结构，以拼接图像和掩码为输入。它以渐进方式输出预测的网格运动。可以通过使用预测的扭曲来 warp拼接 的图像来产生矩形结果。

**残差渐进回归**。观察到扭曲的结果可以再次被视为网络输入，我们设计了一种残差渐进回归策略，通过渐进的方式估计准确的网格运动。首先，我们**不直接使用扭曲图像作为新网络的输入，因为这会使计算复杂度加倍**。相反，我们扭曲了中间特征图，在计算量略有增加的情况下提高了性能。然后，我们设计了两个具有相同结构的回归器来分别预测初级网格运动和残差网格运动。尽管它们共享相同的结构，但由于不同的输入特征，它们被指定用于不同的任务。

### 3.3 目标函数

我们采用包含三个损失函数的综合目标函数来优化网络参数。优化目标如下所示：

（1）

其中 , 分别表示边界项、网格项和内容项。

3.3.1 内容项

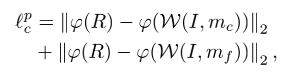
传统方法 [7, 14] 通过保留直线/测地线的角度来保留图像内容，无法处理其他非线性结构。 为了克服它，我们建议从两个不同的角度学习内容保存能力。

**外观损失**。 给定预测的主网格 和最终网格 ，我们强制矩形结果在外观上接近矩形标签 R，如下所示：

(2)

其中 W 表示 warp 操作。

**感知损失**。为了让结果感知上更加自然，我们在高级语义感知中最小化矩形结果和标签之间的 L2 距离，如等式3：

（3）

其中 φ(·) 表示从 VGG19 [25] 的“conv4 2”层提取特征的操作。以这种方式，可以感知各种感知特性（不限于线性结构）。

总之，内容损失是通过同时强调外观和语义感知的相似性形成的，如下所示：

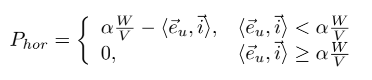
（4）

其中 和 分别表示两个损失的权重。

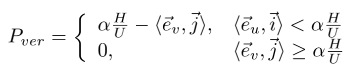
3.3.2 网格项

为了防止矩形图像中的内容失真，预测的网格不应过度变形。因此，我们设计了一个网格内约束和一个网格间约束来保持变形网格的形状。

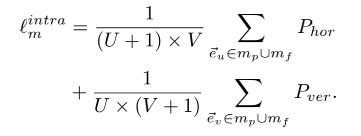
**网格内约束**。在网格中，**我们对网格边缘的大小和方向施加约束**。如图4a所示，我们鼓励每个水平边缘的水平投影方向向右，连同其范数大于阈值 α（假设拼接图像的分辨率为H×W）。我们使用惩罚 来描述这个约束如下：

（5）

其中 i 是指向右侧的水平单位向量。 对于每个网格中的垂直边 ，我们施加类似的惩罚 ，如下所示：

（6）

其中 j 是指向底部的垂直单位向量。网格内约束损失如下式所示，它可以有效防止网格内形状变形。

（7）

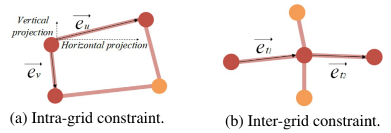
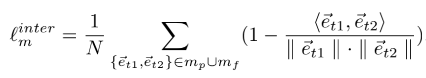


Fig4 网格形状保持限制

**网格间约束**。我们还采用网格间约束来鼓励相邻网格一致地变换。如图4b所示，两个连续的变形网格边缘{}被鼓励共线。

（8）

我们如上所述制定网格间网格损失，其中 N 是网格中两个连续边的元组数。

总之，总的网格项总结如下

（9）

3.3.3 边界项

至于边界项，**我们约束掩码而不是预测网格**。给定拼接图像的 0-1 蒙版（如图 3 所示），我们扭曲蒙版并将扭曲蒙版约束为接近全一矩阵 E，如下所示：

（10）

## 数据准备

为了训练深度图像矩形网络，我们构建了一个图像矩形数据集（DIR-D），其中每个样本都是由拼接图像（I）、掩码（M）和矩形标签（R）组成的三元组。 我们通过以下步骤准备此数据集：

第1 步：采用 ELA [16] 拼接来自 UDISD 数据集 [24] 的图像，以收集大量真实拼接图像。然后我们忽略那些外推面积小于整个图像 10% 的图像。

第 2 步：使用 He 等人 [7]的算法从这些真实拼接图像中生成丰富的不同网格变形 ()，如图 5（右）所示。

第 3 步：应用网格变形的逆向()将真实矩形图像（来自 MS-COCO [20] 和收集的视频帧）扭曲到合成拼接图像，如图 5（左）所示。可以通过扭曲全为一的矩阵来获得掩码。然后我们得到真实矩形图像 (R)、合成拼接图像 (I) 和扭曲矩阵 (M) 的三元组。

第 4 步：手动消除三元组中 I 有失真的数据。每次手动操作需要 5-20 秒。形式上，我们将这个过程重复三个时期，并从 60,000 多个样本中保留 5,705 个三元组。

第 5 步：**将真实拼接图像混合到我们的训练集**以增加泛化能力。具体来说，我们在步骤 2 中从 5000 多个样本中过滤掉了 653 个 R 没有失真的样本。

总之，我们准备了具有广泛不规则边界和场景的 DIR-D 数据集，其中包括 5,839 个用于训练的样本和 519 个用于测试的样本。数据集中的每张图像的分辨率为 512 × 384。

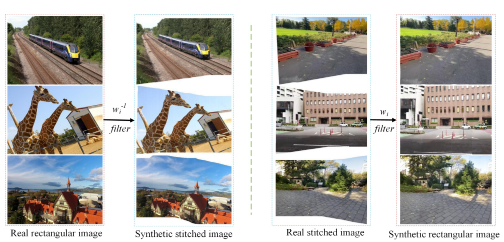


Fig5 数据集准备。左：从真实的矩形图像合成拼接图像。右图：从真实拼接图像合成矩形图像。 然后，合成的样本将经过严格的手动过滤，以形成可靠的数据集，而不会出现明显的失真。

## 实验

在 5.1 节先讨论实验的配置和速度。然后在 5.2 和 5.3 节分别描述比较结果和消融实验结果。

### 5.1 实验配置和速度

网络的优化方法采用 Adam 优化器，学习率初始值为 ，并采用指数衰减的学习率下降策略，总共迭代次数是 100k。Batch 大小为 4，激活函数采用 RELU，但回归器的最后一层不采用激活函数。 分别设置为 1，以及 0.125。UXV 设置为 8x6，并且采用 Tensorflow 实现。采用一个NVIDIA RTX 2080 Ti 的 GPU来完成训练和推理过程。

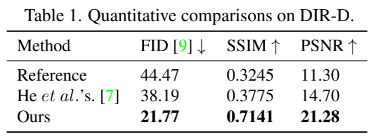
处理一张 10 兆像素的图像只需不到 0.4 秒。与 [7] 的实验配置类似，输入图像将首先被下采样到 1 兆像素，并以下采样分辨率解决网格变形。 然后网格变形将被上采样，并且可以通过使用该上采样变形扭曲全分辨率输入图像来获得矩形结果。**运行时间主要是扭曲（插值）全分辨率图像**。

### 5.2 比较结果

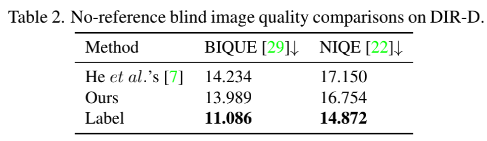
为了全面展示我们的优势，我们在定量比较、定性比较、用户研究和跨数据集评估方面进行了比较实验。

5.2.1 定量比较

我们将我们的解决方案与 He 等人 [7] 在 DIR-D 上的方法进行了比较，其中每种方法测试了 519 个样本。我们用标签计算平均 FID [9]、SSIM 和 PSNR 来评估这些解决方案。定量结果如表 1 所示，其中‘Reference’以拼接图像作为矩形结果供参考。



从这个表中可以看出，在 DIR-D 的每个指标中，所提出的学习解决方案都明显优于传统解决方案。**这种显着的改进归功于我们可以保留线性和非线性结构的内容保留属性**。此外，当矩形结果中物体的位置稍有变化时，它看起来也很自然，但度量可能会有所不同，这使得定量实验不能完全令人信服。因此，我们进一步进行了盲图像质量评估的比较。如表 2 所示，我们采用 BIQUE [29] 和 NIQE [22] 作为“无参考”评估指标，我们的解决方案可以产生更高质量的结果。**这些评估方法是无意识的方法，试图在不需要任何训练数据的情况下量化失真**。我们添加了“标签”的评估以供参考，这表明了性能的上限。



5.2.2 定性比较

为了全面比较定性结果，我们将测试集分为两部分。 第一部分包含丰富的线性结构，适合传统的基线，而第二部分包含大量的非线性结构，例如肖像。

从图 7 的结果中，我们可以观察到我们的方法在两个场景中明显优于传统解决方案。 我们的优势在于**能够保持网格形状保持和内容感知自然的内容保留能力**。由于线检测能力有限，传统解决方案在具有线性结构的场景中表现不佳。失败发生在具有非线性对象的肖像中，因为非线性属性不包括在其优化的能量中。

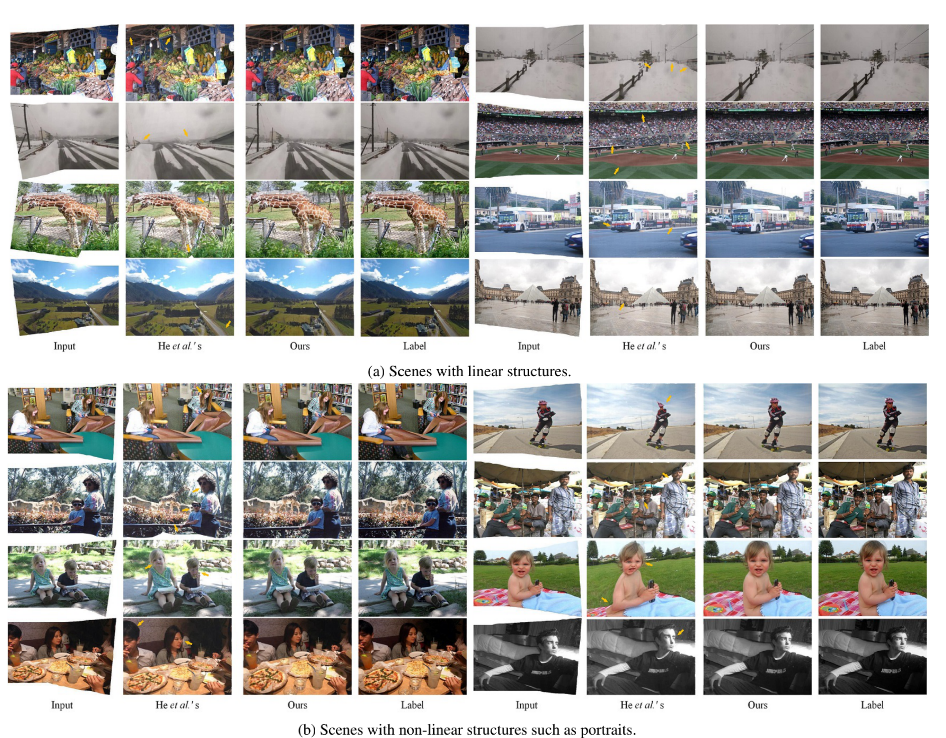


Fig7 在 DIR-D 数据集上的定性比较

5.3.3 User Study

图像矩形的动机是用户对拼接图像中不规则的边界不满意。因此，我们的目标是制作让大多数用户满意的矩形图像。

我们对视觉偏好进行用户研究。形式上，我们将 He 等人的算法与我们的初级矩形和最终矩形（如图 3 所示）一一进行比较。每次，三个图像都显示在一个屏幕上：输入、He 等人的矩形和我们的（主要或最终）。我们每次都打乱不同方法的顺序。用户可以放大图像并且需要选择哪个结果是优选的。在这项研究中，我们邀请了 10 名参与者，其中包括 5 名具有计算机视觉背景的研究人员/学生和 5 名社区以外的志愿者。结果如图6所示，我们的解决方案受到更多用户的青睐。

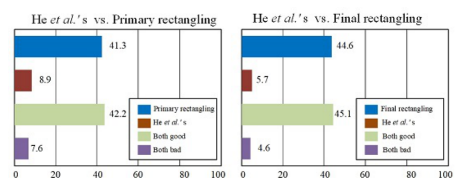


Fig6 在 DIR-D（519 个测试样本）上的用户研究。图中所展示数字表示比例并且是 10 名参与者的平均值。

5.2.4 跨数据集评估

在跨数据集评估中，我们采用 DIR-D 数据集训练模型，然后用其他数据集进行测试。

形式上，我们采用不同的图像拼接方法（SPW [18]、LCP [10] 和 UDIS [24]）来拼接**经典图像拼接数据集** [4,5,10,30]。然后，使用不同的算法将拼接的图像用于矩形。结果如图 8a 所示，我们的解决方案在矩形结果中产生的失真更少。

为了展示我们在更一般的场景中的有效性，其中没有消除伪影和投影失真，我们展示了图像拼接失败案例的矩形结果。如图 8b 所示，我们的方法仍然有效。

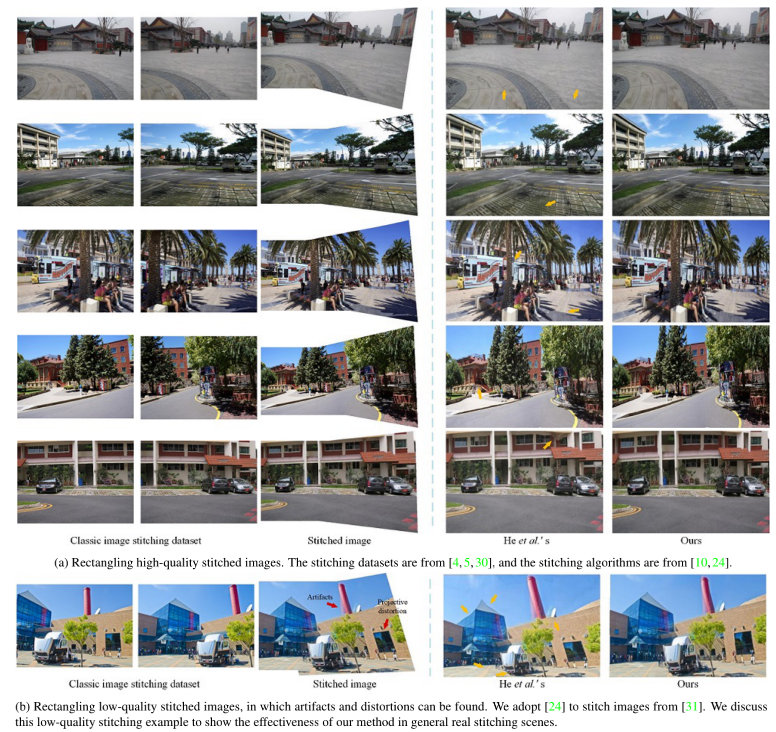


Fig 8 跨数据集评估。箭头表示失真区域

### 5.3 消融实验

本文提出的方法简单但有效。我们在 DIR-D 上验证每个模块的有效性。

**损失函数**。我们将残差回归器作为基线结构，并评估目标函数中不同约束项的有效性。如表 3 的实验 1-3 所示，**内容项和网格项都可以显着提高我们的性能**。

**网格分辨率**。我们测试了 4×3、8×6 和 16×12 的不同网格分辨率。如表 3 的实验 3-5 所示，4×3 网格降低了矩形性能，而 8×6 网格和 16×12 网格给出了相似的结果。然而，16 × 12 网格带来了更多的计算成本，因此我们在实现中采用了 8 × 6 网格。

**残差渐进回归**。我们在表 3 的实验 6-7 中验证了我们的残差渐进回归策略的有效性，其中残差回归器继续基于主回归器细化矩形结果。**尽管对 DIR-D 数据集的改进很小，但这种策略增强了我们的泛化能力，以避免其他数据集的边界不均匀**，如图 9 所示。

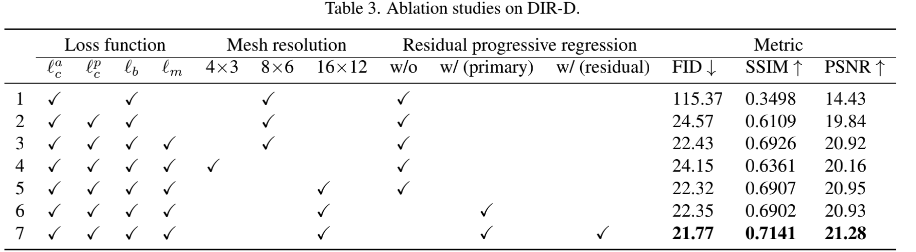




Fig9 其他数据集（“potberry”[10]）上残差渐进回归策略的消融研究。 圆圈突出显示边界不均匀的区域。

## 结论

在本文中，我们提出了第一个用于图像拼接的深度图像矩形解决方案和数据集。 与传统的两阶段方法相比，所提出的解决方案是单阶段方法，可以使用预定义的刚性目标网格进行高效的并行计算。此外，我们的解决方案可以同时保留线性和非线性结构，在数量和质量上都证明了优于现有方法的优势。