基于轮廓估计曲面细分的表面平整化方法

Yuhan Xu* and Renqing Luo†

Department of Applied Mathematics, University of Washington, Seattle, Washington 98195

Department of Physics, University of Washington, Seattle, Washington 98195

(Dated: 2023 年 3 月 9 日)

摘要：三维物体的表面投影到二维上的过程中，由于透视的影响，在物体表面的图像会根据表面曲率大小产生不同程度的畸变。本文将介绍一种将规则弯曲物体表面的该类型畸变平整化的非精确方法。该方法的主要思路是通过物体二维图像的轮廓曲线，大致的估计出可用于描述三维物体表面的网格化曲面细分，接下来将每个不同大小和形状的网格区块透视逆变换到形状大小完全相等的矩形，最后将每个相同的矩形依次拼接重组就可以得到一个粗略平整化的矩形。本文将介绍和展示运用该方法解决弯曲书页展平的具体过程和结果，以此论证该方法的可行性和局限性。

关键词：扭曲图像校正，文档图像处理，文档图像平整化，轮廓估计曲面细分

I. 介绍：动机与问题陈述

本文实现的方法最初是为了解决相机拍照扫描弯曲书页带来的图像扭曲问题。网课期间，我们需要将笔记本上的作业拍照上传，但是由于书页受应力产生了弯曲，页面上的内容也会因此在相片中呈现畸变。拍摄照片中书页越是靠近书脊处的部分，文字的形变越是剧烈。如图 1 所示，图像中的文字在书页横轴的形变会在凸起部分呈现拉伸，在凹进书脊的部分为收缩，文字在纵轴也会根据表面的凹凸向下或向上产生扭曲。文字的这样的形变是连续且规则的一种由于透视效应所产生的畸变 [1]。尽管，这样的形变并不会改变文字的结构，但它还是会极大地降低电子档案扫描文件的质量，进而给我们带来不愉快的阅读体验。因此，我们开始试着探索和解决在合理范围内任意位置和角度下拍摄弯曲书页的展平方法。在找到解决书页展平的特定方法后，我们进一步地认为该方法可以简单地推广到任意单调变化的三维物体表面平整化问题。

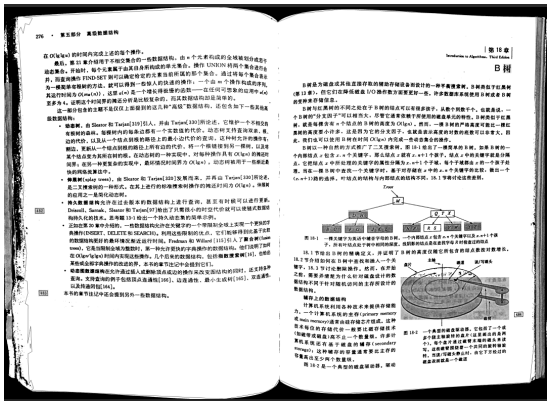

图 1: 经过预处理的书页弯曲实例图

A. 问题分析与解决思路

在解决问题之前，我们更加深入地考察了透视效应导致扭曲出现的原因。透视效应简单的来说就是同样的物体距离镜头越远成像的大小越小，反之亦然。书本页面在相机中成像的扭曲，也可以近似的被透视效应所描述，书本的透视畸变可以分为全局和局部。全局的透视畸变指的是书本轮廓整体的形变。这是由于镜头没有垂直于书本对角线交会的中心点，也就是说如果镜头在书本中心的正上方，整个打开书本的轮廓四个边角点应当连接成矩形。但是，如果镜头在其他角度和位置，书本的轮廓则会大致上是不太规则的四边形。局部的透视畸变指的是书页弯曲的高度差带来的文本局部内容的扭曲。由于书页本身是一张平整的矩形纸张，因此书页上的每一条纵向直线（长边）应当是等长的。不过由于书页横轴的弯曲导致了每条长边到镜头的距离发生了变化，所以在靠近书脊处的长边会被收缩，靠近凸起处的长边会被拉伸。如图 2 所示，其中灰色的圆柱体类似于书页的长边，这些圆柱体本身是完全一致的，不过它们到镜头的距离不同导致了其成像大小的不同，而每一个圆柱体的形变就是局部透视畸变。全局的透视畸变只需要将整个书本的

* yuhanx5@uw.edu

† renqing@uw.edu

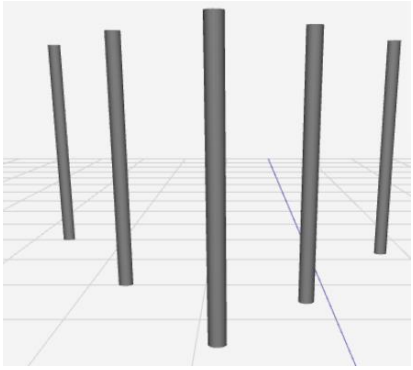

图 2: 轮廓由透视畸变产生弯曲的类比

四边形近似轮廓，通过透视变换映射到一个指定大小的矩形即可解决。不过，局部的透视畸变就较为麻烦，这也是本文将要解决的中心问题。书本长边由于透视投影导致的长度变化，直接地反映在了轮廓的弯曲上，因此我们可以将轮廓的形状作为切入点，找到解决弯曲文本表面平整化问题的解决方法。将页面细分成一条条长边组成的条带，然后将每一个条带线性地拉伸或收缩至指定长度是不可行的。因为尽管在同一条带上的弯曲高度是相同的，但是上方到下方镜头视角的变化，导致了局部页面实际到镜头的距离是不同的。

本文给出解决问题的主要思想是：确定透视原理作为基本原则，能够解释文本图像的畸变现象。然后，将复杂的页面弯曲高度差带来的扭曲问题细分为可以被基本原则解决的子问题，最后将所有基本子问题拼接回去，复杂的问题就能够被近似的解决。我们已知最为基本的四边形透视畸变的解决方法，那么我们可以将曲面细分成为一个个分立的四边形区块，然后对于每一个区块透视逆变换至相等的小矩形，最后将所有小矩形依次拼接会一个整体的大区块。这是非常简单且被人熟知的古典思想，不过本文解决问题的方法还有一些需要技巧的技术性问题。比如，如何通过轮廓的形状具体的确定曲面细分的规则，使得细分的曲面能够尽可能的还原真实的三维曲面。这也是本文的方法及其一些相关工作的重点。

B. 相关工作

平整化三维物体表面在二维的投影图像具有多样的解决方法，不同的解决方法主要是由具体的适用场景所决定的。本文方法的适用场景有两个主要条件：1. 任意角度和距离下单镜头单次成像图片。2. 平整化处理的算法满足简易实现，时效性，和轻量级的需求。一种通过页面上文本内容形成的纹理流场，平行性，和等行距来恢复表面形状的方法 [2]，满足我们需要的第一个条件，但是它的算法实现会比较的复杂。另一种通过透视几何结合书页弯曲的圆柱形建模能够较为精确的矫正扭曲的文档 [3]，不过这种方法需要提供摄像头的距离和角度用于透视几何的计算，因此它适用于固定相机位置的高拍仪，而不能是手机在任意位置下的拍摄。还有方法通过拍摄连续视频提取多张关键帧的再拼接处理，以达成纠正文档扭曲的效果 [4]。有开源项目用极小优化算法纠正扭曲文章 [5]，但是该方法需要耗费大量运算资源和时间。还有一些借助额外设备辅助的方法，比如激光网格三维重建 [6] 或者立体相机 [7] 等。本文提供的解决方法或许效果比不上上述的方法，不过该方法的使用非常简便，只需要一部能拍照的手机，并且算法的实现也非常的简易。

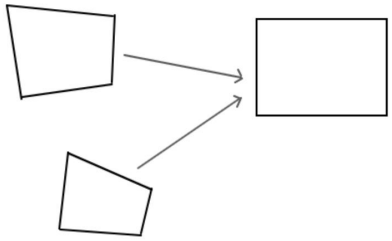

图 3: 四边形透视逆变换到矩形

II. 解决问题的方法

本文在之前已经介绍了问题解决的基本思想，这里我们将详细的提供方法的步骤和具体的技术细节问题的解决。该方法的步骤可分为四个部分：1. 轮廓预处理，2. 轮廓函数拟合，3. 网格化曲面，4. 透视变换后重组。在轮廓预处理中，我们首先将提取二维的曲面图像的边界轮廓，然后将图像按照轮廓切分为若干个光滑曲面，最后将每个光滑曲面划分为上下左右四条边界线。对于轮廓函数拟合，首先我们对于每条边界轮廓线按指定精度采样坐标点，然后采取合适的函数并在边界范围内进行拟合，最后从上/左边界线的拟合函数连续地演化到下/右边界线的拟合函数。对于

图 4: 异常轮廓线示意图，黑色线条为轮廓线。

网格化曲面，我们根据边界线的曲率大小及其正负决定函数族采样间距，然后由上下左右函数族交点可得到细分小曲面四边形网格点。最后的透视变换后重组，我们首先对于每个网格进行透视变换到统一大小和形状的小矩形，然后将每个经过透视变换的小矩形依次拼接重组会完整的大矩形，这样我们就得到了一张扭曲内容经过了展平的图像。

A. 轮廓预处理

既然我们需要通过页面的轮廓得到表面弯曲程度，那么我们首先需要提取照片中书页的轮廓。本文采用 OpenCV[8] 中实现的轮廓提取算法 [9], 该算法需要我们拍照时书页和背景需要清晰并且有较为明显的黑白对比度。因为背景需要与书页有较为鲜明的黑白对比度，所以在实际中我们还需要先对图 1 进行二值化处理。对于书本文档，在实际处理中我们首先提取了整个书本的四边形近似轮廓，然后对该四边形进行透视变换到一个矩形，以此完成对全局透视的矫正。

在完成了对轮廓的提取后，接下来我们可以看到在书本的轮廓中间，也就是书脊处的上下边界线有一个连续但不可求导的点。这样的轮廓线会极大的干扰我们后续的轮廓线拟合，并且我们是无法在一个有不可求导点的函数上计算曲率的。因此，接下来我们需要将该轮廓切分为左右两个光滑的曲面，也就是左右两张书页。在计算机中提取的轮廓本来不是光滑的，而是离散点对弯曲轮廓的近似。因此，我们可以很轻易的获得近似曲线上每两个点连成的线段的斜率，然后依次求当前线段到下一线段斜率的改变，并且设置一个阈值，如果超过该阈值就可以得到两个线段的交汇点为不可求导点。在实际的处理中，我们需要尽可能的缩小寻找不可求导点的范围，因为如图 4 所示，轮廓的提取取决于图片的质量会出现不合理的多个不可导点。对于书页来说，四个边角都是不可求导点，但是我们不用根据上述方法获得，而是之前我们近似提取的书本四边形轮廓本身就是由这四个边角点所构成的。最后，根据这四个点加上中间两个不可求导点，我们可以将书页划分为两个光滑曲面，左右两个光滑曲面都可以对应地划分为上下左右四条光滑的边界线。接下来，我们就可以对这些边界线求拟合函数。

B. 轮廓函数拟合

轮廓函数的拟合是为了让边界曲线从上到下和从左到右的演化和计算轮廓各个位置的曲率。曲线演化是为了解决局部文字在纵轴的上下弯曲，曲率计算是为了解决的是横轴文字左右的收缩或拉伸。之前我们得到的边界线，实际上是离散的点构成的，点的数量取决于轮廓提取的采样精度。因此，我们需要用这些离散点去拟合一个光滑函数。这里我们选择采用 n 阶多项式函数进行边界轮廓线的拟合，

$$P(x) = c_n x^n + c_{n-1} x^{n-1} + \cdots + c_1 x + c_0 \quad (1)$$

这是因为多项式函数能够很快速的被拟合，在有限小范围能够拟合任意的光滑曲线，并且曲率的计算也极为的简单。得到边界线的多项式的表达式后，我们就可以计算演化多项式函数族，以此模拟透视效应下局部扭曲在离镜头越远处越明显，这样的演化的确并非线性。比如，从一束激光在大角度仅仅转动一点点，扫过的长度和结束点到激光器的距离都会极大的变动。也就是说，正确描述镜头转动 θ 度，镜头划过距离与结束点到镜头的距离之比应当是 $\sin\theta$，其中 θ 为两条线的夹角。不过，在书页这个例子中我们可以采用小角度近似，使得 $\sin\theta \approx \theta$。因为，精确的计算需要知道镜头到书页的垂线距离，所以我们选择了近似地均匀的演化边界线拟合函数。

C. 网格化曲面

网格化曲面是用于描述三维物体表面投影到二维上的方法，其中的每一个细分区块都可以被看作原本平整表面，均匀分割过后的一个矩形在其位置上经过透视变换的结果。本文将通过书页的边缘轮廓，大致

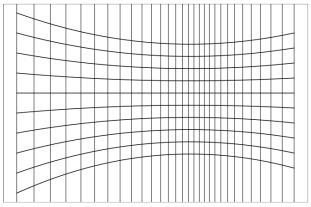

图 5: 向内凹陷

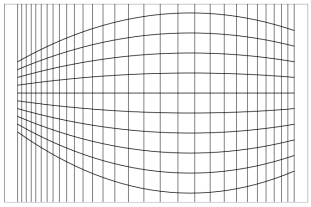

图 6: 向外凸起

的估计细分曲面的规则。其中根据边缘轮廓的拟合函数族的演化和曲率的计算是整个方法中最为关键的步骤，因为该步骤决定的曲面细分的规则，这直接决定了最后展平的效果和质量。网格化曲面的构成是建立在横轴和纵轴的一系列经过演化的多项式函数相互交叉所切分出来的，其中交叉切分的演化函数的采样间隔是有边界线函数曲率及其凹凸性所决定的。

1. 多项式函数的演化

之前已经讨论过为什么可以采用均匀的多项式函数演化，这里我们将详细的介绍我们该如何演化多项式函数。一个多项式函数能够通过调整其系数的方式，均匀地演化到另一个多项式函数上，只要保证这两个多项式函数的最高阶数相等。这样我们就可以让多项式的每一项系数，按照指定的步长均匀地变化到另一个多项式的对应系数上。准确的说，多项式函数 $P_0(x)$ 均匀地演化到 $P_n(x)$ 的步长为，

$$d = \frac{P_n(x) - P_0(x)}{n+1} \tag{2}$$

其中 $n+1$ 为多项式演化的个数（步数）。根据该步长，多项式函数均匀演化步进规则为，

$$P_{i+1}(x) = P_i(x) + d \tag{3}$$

其中 i 为取值从 0 到 n 的正整数索引。尽管多项式的演化是均匀的，但是由于我们需要考虑曲率导致的非均匀采样间隔，因此可以设置一个较多的演化步数，使得步长较小，从而让接下来的间隔采样的精度提高。

2. 凹凸性与曲率的计算

曲率的计算是为了解决书页在横向的收缩或拉伸，并且它需要对函数求一阶导数和二阶导数，因此我们

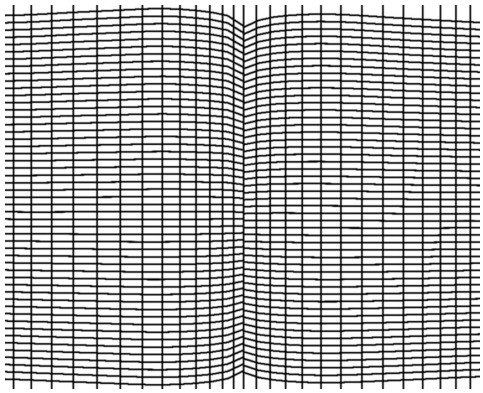

图 7: 书页的近似曲面细分

需要拟合的多项式函数至少是二阶多项式。不过在实际中我们应当采取较为高阶的多项式，以便应对更近复杂的弯曲形状。这里我们不关心中间的函数演化族，曲率的计算仅仅只作用于边界轮廓的拟合函数。一般来说，书页的边界曲线只有一个极值点，因此我们可以通过二阶导数判断函数的凹凸性。不过，如果书页反复弯曲，有多个极值点，那么我们就需要根据拐点进一步的将曲线划分为多个具有凹凸性的函数。拐点的求取不能通过离散的采样点判断，而是需要对拟合多项式函数解析式的二阶导数求解，也就是对方程 $P''(x) = 0$ 求解 x 的取值。接下来，对于每一个根据拐点划分的边界多项式曲线 $P(x)$，我们判断其二阶导数在划分范围内的正负性，以决定函数的凹凸性。横轴的边界线函数的曲率 κ 在 x 点为，

$$\kappa(x) = \frac{P''(x)}{(1 + P'(x))^{3/2}} \tag{4}$$

曲率是描述曲线弯曲的程度，书页凸起越高处，文字被拉伸的程度越严重，轮廓呈现的弯曲程度也越大，此处曲率的绝对值也越大。因此，演化函数在曲率绝对值大的地方，采样间隔应该被拉伸，这样透视逆变换时就能够将被拉伸的文字尽可能地收缩至原本尺寸。

由于函数无论是凹或凸，其弯曲的程度都是一样的，比如 $f(x) = x^2$ 和 $f(x) = -x^2$ 在各点的曲率绝对值都是一样的，但是边界函数两侧凹凸性的不同应当影响曲率的取值。如图 5 所示，尽管上下边界函数的曲率绝对值是一样的，但是在曲率较大的地方函数演化采样的间隔反而是缩小了，而不是和图 6一样增大采样的间隔。因此，从图 5 不难看出从凸函数演化到凹函数时，采样间隔反而会在曲率较大的地方收缩，

从图 6 可以看出从凹函数演化到凸函数时，采样间隔会随着曲率的增加而扩大。此外，如果边界函数从上到下的改变越小，那么可以想象采样的间隔就会越来越接近均匀。也就是说，实际上在具体的计算实现中，我们不能仅仅只考虑一侧边界函数的曲率，而是需要计算两侧曲率之差的平均，

$$k = \frac{\kappa_f - \kappa_i}{2} \tag{5}$$

最后决定采样间隔的缩放系数为，

$$\gamma = |k|^{-\frac{k}{|k|}} \tag{6}$$

其中 $\frac{k}{|k|}$ 是用于判断 k 符号的正负，图 5 中上边界函数的曲率 κ_i 和下边界曲率 κ_f 有 $\kappa_i = -\kappa_f$，可得平均曲率之差为 $k = \kappa_f$，由下边界函数曲率为正可计算出缩放系数为 $\gamma = \kappa_f^{-1}$。因此，就有了图 5 在曲率绝对值较大的地方呈现更为密集的网格分布。类似的，图 6 的缩放系数计算可得为 $\gamma = \kappa_f$，使得在曲率绝对值较大的地方呈现更为稀疏的网格分布。

最后，在有了拟合函数的演化和曲率的计算后，我们可以处理具体的网格划分。设 M 为横轴的演化函数族的采样精度（个数），N 为纵轴的采样精度。在横轴的边界曲线上计算曲率时，需要根据纵轴的采样精度将其分为 N 个均匀的采样点 x_i，然后对于每一个采样点我们求取在该点上的缩放系数 γ_i，也就有横轴上的所有缩放系数组成的数列为 $\{\gamma_i\}_{i=0}^{N-1}$，纵轴上的缩放系数数列为 $\{\gamma_j\}_{j=0}^{M-1}$。接下来，我们需要将这些缩放系数对应到整个横轴的尺度上，因此需要对缩放系数数列进行归一化处理，

$$\{\Gamma_i\} = \frac{\{\gamma_i\}_{i=1}^{N-1}}{\sum_{i=1}^{N-1}\gamma_i} \tag{7}$$

同样的对纵轴归一化处理后可得 $\{\Gamma_j\}$。注意到在归一化处理时，我们将 $i = 0$ 排除在外，这是因为在接下来，计算横轴采样间隔点 X 的时有，

$$X_{i+1} = X_i + \Gamma_i(x_{N-1} - x_0),\ X_0 = x_0 \tag{8}$$

这里的 $i = 0$ 为起始点不参与缩放系数的计算。最后，我们就可以根据横轴上的间隔点数列 $\{X_i\}_{i=0}^{N-1}$ 计算其对应点的每一个纵轴演化函数，同样的也可以计算纵轴上关于数列 $\{Y_j\}_{j=0}^{M-1}$ 的所有对应横轴演化函数。这样纵横交错的演化函数族就完成了对细分曲面的划分，

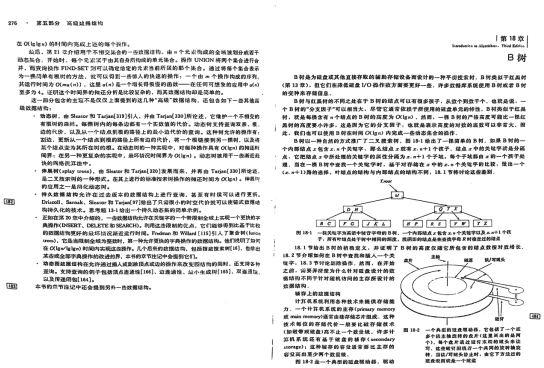

图 8: 书页经过平整化处理后的结果

图 7 为图 1 的曲面细分估计。该划分曲面尽管是根据轮廓及其曲率的非精确估计，但是在完成接下来的最后一个步骤后还是可以得到不错的结果。

D. 透视逆变换后重组

重组的思路和原理是整个方法中最为简单和直接的，也是具有正确且单一实现的一个步骤。这里我们全部所需要做的仅仅只是将之前曲面细分结果中的每一个四边形区块，经过透视逆变换后，依次的拼接回一张完整的图像。但是，这个部分也是整个方法在程序实现中最为繁琐，最多由索引引发报错，代码量最大，运行耗时最长，和最需要并行优化的地方。首先，之前我们对网格的划分仅仅只是给出了纵横函数族的采样和间隔，但是计算机并不直接从中知道具体每一个四边形区块，也就无法对其进行处理。因此，我们首先需要从左到右从上到下地依次计算所有纵横函数族的交点，并将每个点 $p_k = (x_k, y_k)$ 存储在一个列表中 $\{p_k\}_{k=0}^{MN-1}$。然后，根据交点的索引以及采样个数决定四边形区块的划分。我们按照四边形的左下角点为基准，以向左，向上，和向左上一起四个点构成一个四边形区块，并且由于在最右侧边界的左下角点没有向右的下一个点，以及在最上方边界的点没有下一个向上的点，因此我们对于上边界和右边界点是不提取四边形区块。准确来说，点 p_k 具有所对应的四边形区块为

$$B_k = \{p_k, p_{k+1}, p_{k+M}, p_{k+M+1}\}, \tag{9}$$

$$\bmod(k) \neq M-1 \ and \ \ k < M(N-1). \tag{10}$$

接下来，在我们得到了所有的四边形区块点阵后，我们就可以通过透视逆变换依次将四边形映射到全等

矩形上。本文透视逆变换的处理是由 OpenCV[8] 中的内置函数完成的，其原理是求解下述等式中的 3×3 透视变换矩阵 M，其中 $t_i = x_i h_7 + y_i h_8 + h_9$ 为比例系数，(x_i, y_i)，$i = 0, 1, 2, 3$ 为输入四边形的四个边角点，(x_i', y_i') 为输出四边形的边角点。

$$\begin{bmatrix} t_i x_i' \\ t_i y_i' \\ t_i \end{bmatrix} = \underbrace{\begin{pmatrix} h_1 & h_2 & h_3 \\ h_4 & h_5 & h_6 \\ h_7 & h_8 & h_9 \end{pmatrix}}_{M} \cdot \begin{bmatrix} x_i \\ y_i \\ 1 \end{bmatrix} \tag{11}$$

接下来根据透视变换矩阵 M，我们可以将对应四边形区块上的图像内容（所有像素点）透视变换到目标全等矩形上 [10]。最后再依次将每一个经过透视逆变换得到的矩形按照顺序拼接重组到一张大的矩形平面上，到此整个平整化就完成了。

这个步骤中几乎所有的操作都是重复且独立的，在提取演化函数交点点阵时，每条函数上的计算都是一样且重复的，此外还有用点阵获取每个区块，对于每一个区块计算透视逆变换。这且重复且独立的计算都可以并行地同步处理，从而极大地提升计算机处理效率和减少算法的时间开销。

III. 结果与分析

如图 8 所示，这是我们经过上述方法将图 1 中弯曲的书页经过平整化处理后得到的结果。图 9 为书脊处的局部对比图，这里我们可以看到经过对每一个网格进行透视逆变换后，我们成功将原本扭曲的句子展平到了平直且相互平行句子，使得文字在纵向的弯曲和横向的缩放都得到了较为明显的改善。不过由于本文解决纵轴的收缩问题的方法是采取了非精确的估计，因此我们可以看到文字在不同地方还是有些许的尺寸不一致。此外，我们最初版本的代码实现假设了书页纵轴为垂线，但是实际上在图 1的左侧边界线存在轻微的倾斜，因此这导致了左页下方靠近书脊处能够看到一些轻微的扭曲。

此外，经过平整化的文档在传统的光学字符识别（OCR）具有较为良好的表现，图 9的文档在扭曲部分的字符识别率（CRR）为 81.73%，经过平整化的 CRR 为 96.28%。不过，对于基于深度学习文本检测的 OCR [11]，该方法并不会带来显著的提升，因为扭曲文档经过文本检测后也同样具有非常高的字符识别率。虽然该方法对现代 OCR 的提升并不明显，不过它对弯曲页面上的手写内容，公式，和图片等非文字内容的平整化对阅读体验有着明显的改善。

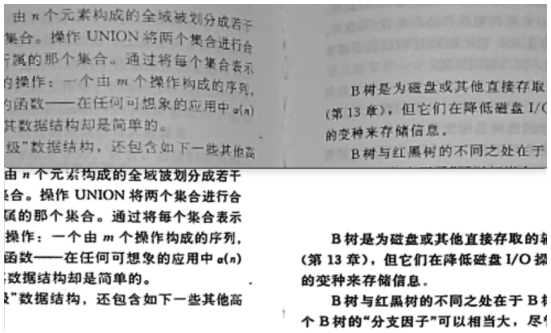

图 9: 局部书页弯曲平整化处理前后对比图

IV. 总结

最近几十年来，随着数字化和信息技术的发展，文档数字化和文档图像处理有着越发重要的实用和研究价值。其中扭曲文档的矫正方法也是日新月异，多样化的解决方法应对于解决不同的场景下的问题。本文从书页的轮廓形状出发，解决了弯曲书页在单镜头单次成像的平整化问题。本文提出的扭曲矫正方法，不仅能够适用于文字文档扭曲的矫正，还能够处理任意单调形变曲面上扭曲图像的矫正。同时，该方法对硬件设备的需求极低，只需要单镜头拍摄一张照片，并且该方法的算法复杂度也比较低，算法的实现也是十分的简单。不过，该方法由于采用的是轮廓的曲率去估计有限网格的切分，因此该方法是一种非精确的矫正。尽管，在实验测试中该方法的展平效果有着不错的表现，但是这种估计方法并不会有严格的理论支持。该方法的主要局限性是需要文档的轮廓十分清晰，并且无法处理褶皱和不规则扭曲的文档的展平，其主要优点是能够通过一个内存占用较低和算力需求极低的软件解决日常大多文本拍照扫描的扭曲问题。

[1] R. MOHR, Projective geometry and computer vision, in *Handbook of Pattern Recognition and Computer Vision*, pp. 369–393.

[2] J. Liang, D. DeMenthon, and D. Doermann, Flattening curved documents in images, in *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, Vol. 2 (2005) pp. 338–345 vol. 2.

[3] H. Cao, X. Ding, and C. Liu, Rectifying the bound document image captured by the camera: a model based approach, in *Seventh International Conference on Document Analysis and Recognition, 2003. Proceedings.* (2003) pp. 71–75 vol.1.

[4] B. Jiang, S. Liu, S. Xia, X. Yu, M. Ding, X. Hou, and Y. Gao, Video-based document image scanning using a mobile device, in *2015 International Conference on Image and Vision Computing New Zealand (IVCNZ)* (2015) pp. 1–6.

[5] M. Zucker, Page dewarping: https://mzucker.github.io.

[6] G. Meng, Y. Wang, S. Qu, S. Xiang, and C. Pan, Active flattening of curved document images via two structured beams, in *2014 IEEE Conference on Computer Vision and Pattern Recognition* (2014) pp. 3890–3897.

[7] A. Yamashita, A. Kawarago, T. Kaneko, and K. Miura, Shape reconstruction and image restoration for non-flat surfaces of documents with a stereo vision system, in *Proceedings of the 17th International Conference on Pattern Recognition, 2004. ICPR 2004.*, Vol. 1 (2004) pp. 482–485 Vol.1.

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

[9] S. Suzuki and K. be, Topological structural analysis of digitized binary images by border following, Computer Vision, Graphics, and Image Processing **30**, 32 (1985).

[10] A. Distante and C. Distante, *Handbook of Image Processing and Computer Vision*, Vol. 2 (Springer, 2021) pp. 174–176.

[11] S.-X. Zhang, X. Zhu, J.-B. Hou, C. Liu, C. Yang, H. Wang, and X.-C. Yin, Deep relational reasoning graph network for arbitrary shape text detection, in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (2020).