

第四章 基于球面中心投影模型的畸变矫正方法

4.1 引言

全景图像通常由多通道图像拼接而成，这个过程涉及到许多的几何变换和投影，这就使得图像矫正的过程显得相当复杂。

本章首先对全景图像拼接的过程进行整体认知，之后引入球面中心投影模型，根据全景图像的拼接过程来进行分解投影，通过严谨的数学推导，最终得到投影前后对应像素点的映射关系。

在此基础上，本章进一步阐述如何通过引入倾斜角因子，提高球面中心投影模型的矫正能力，使其能够处理摄像头高度不同，场边距不同，俯仰角不同的全景图像。

4.2 球面中心投影

全景图像往往是由若干张图像经拼接而成。

图像拼接示意图如图 4-1 所示，其基本流程如下：

- 1) 从同一位置拍摄一系列图像，即围绕光心旋转相机；
- 2) 计算第二个图像与第一个图像之间的变换；
- 3) 计算第二个图像以与第一个图像部分重合；
- 4) 将两者融合在一起创建拼图；
- 5) 如果有更多图像，重复以上步骤。

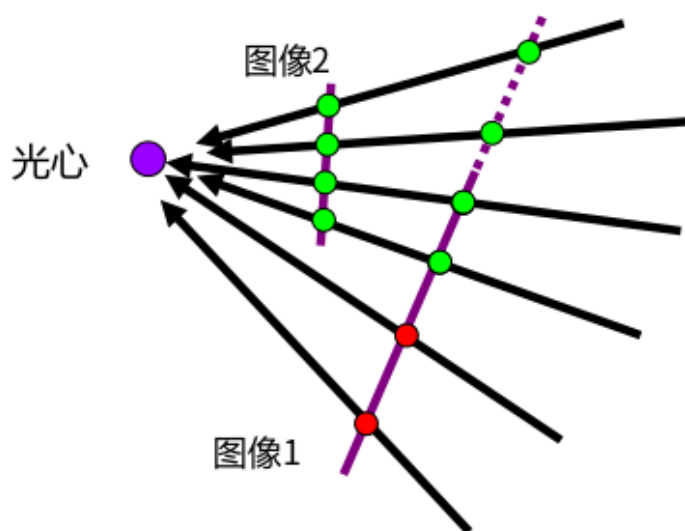


图 4-1 图像拼接示意图

假设拼接的图像位于球面上，如图 4-2 所示。

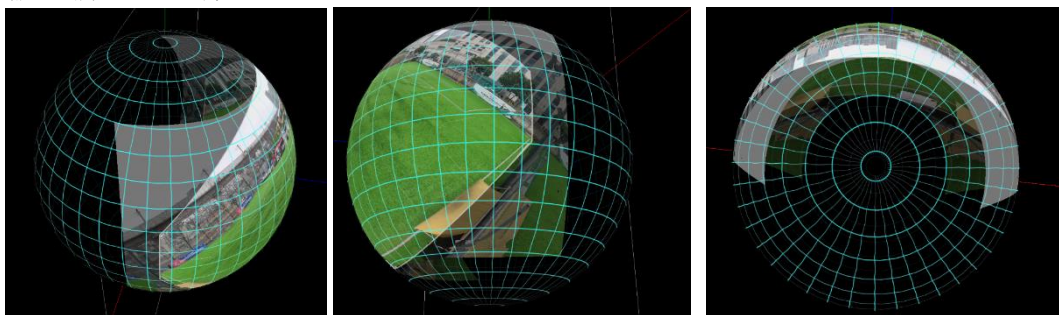


图 4-2 位于球面上的全景拼接图像

那么能否“反其道而行之”，过球心做中心投影，通过在内球面滑动一个内接于球面的平面，将原图逐次投影，分解得到“拼接前”的未畸变图像。这就是基于球面中心投影模型的矫正方法。

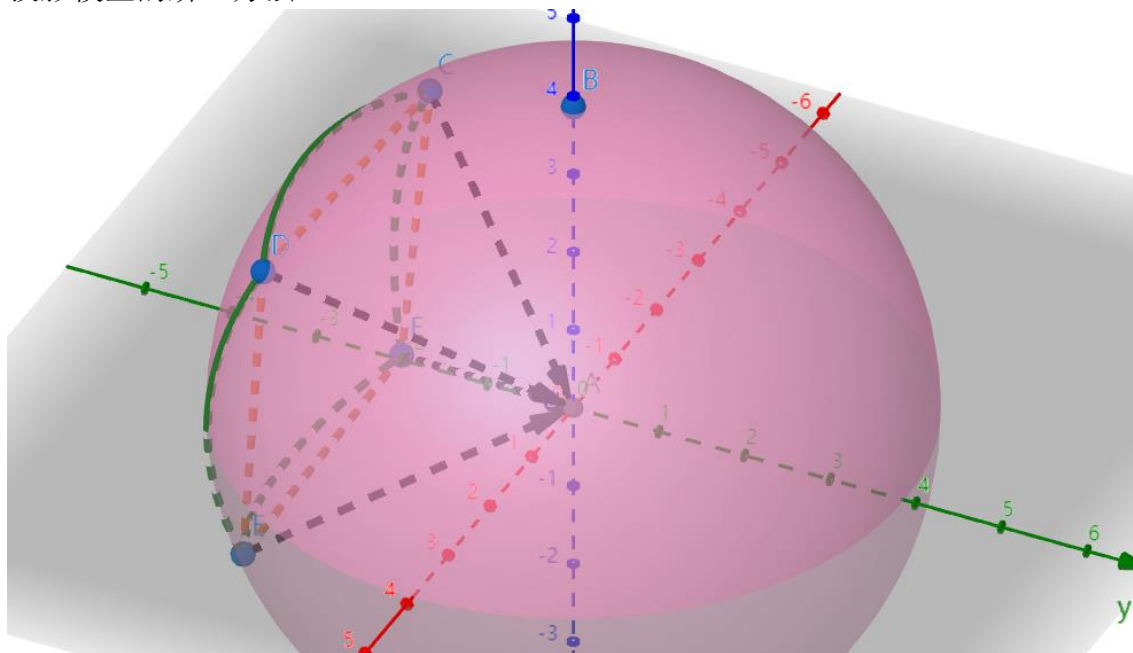


图 4-3 球面中心投影模型

假设原图像关于过球心的水平大圆对称，如图 4-3 所示。

记原图像大小为 (w, h) ，其沿 w 方向的张角为 $\text{angle}(0-180)$ ，内接平面大小为 $(x\text{Dim}, y\text{Dim})$ ，其相对于原图像的偏移为 $\text{initCol}(0-180)$ 。平面与球心的关系如下图所示。

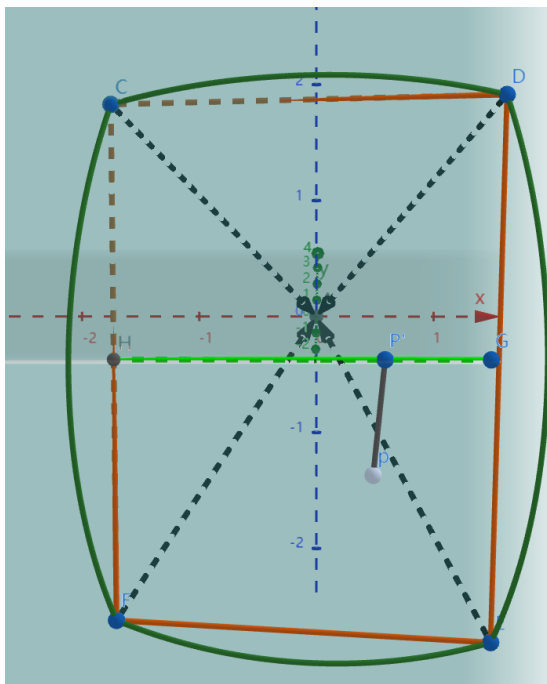


图 4-4 平面球心示意图

首先根据原图像宽 w 对应的角度 $angle$ 计算球面半径

$$r = w / (angle / 180 * \pi) \quad (4-1)$$

显然球心与平面构成了一个等腰四棱锥，由几何学知识很容易得到球心到平面的距离为

$$z = \sqrt{r^2 - (xDim/2)^2 - (yDim/2)^2} \quad (4-2)$$

记平面上一点为 $P(x, y)$ ，要求出中心投影前点 P 在原图的对应点 $Q(xd, yd)$ 。记平面上一点为 $P'(x, yDim/2)$ ，其到球心的距离为

$$\|op'\| = \sqrt{(x - xDim/2)^2 + z^2} \quad (4-3)$$

根据立体几何的知识，不难证明 PP' 与水平线 1 垂直（注意原图像关于过球心的水平大圆对称，则水平线 1 为水平大圆与平面的交线），则三角形 OPP' 是直角三角形，据此可得直线 OP 与水平面的夹角 θ 的正切值为

$$\tan \theta = \|pp'\| / \|op'\| = (y - yDim/2) / \sqrt{(x - xDim/2)^2 + z^2} \quad (4-4)$$

则中心投影前点 P 在原图的对应点 Q 的纵坐标为

$$\begin{aligned} y_d &= r \times \arctan(\theta) + \frac{h}{2} \\ &= r \times \arctan [(y - yDim/2) / \sqrt{(x - xDim/2)^2 + z^2}] + h/2 \end{aligned} \quad (4-5)$$

x 和 xd 的变换关系同理，不同的是，在计算得到直线 OP 与过 z 的垂直面的夹角 φ 后，需要将该值加上偏移量 $initCol$ 的一半，才是其对应的真正弧度值，将该值乘以半径 r 得到 xd 。将 x 、 y 互易得到中心投影前点 P 在原图的对应点

Q 的横坐标为（注意考虑偏移量）

$$x_d = r \times \arctan [(x - xDim/2)/\sqrt{(y - yDim/2)^2 + z^2}] + initCol * \pi/360 \quad (4-6)$$

然而，观察到原图（参考图 4-5）只在沿 h 方向发生严重畸变，而在沿 w 方向未发生形变，故不对该方向做上述球面映射，而是用一个近似的线性映射来代替（考虑原图像在球面上沿 h 方向的张角非常小，这个时候 x 和 x_d 的变换关系可以看成一段圆弧与圆弧两端连线的变换关系）。

上述映射即假定 $y - yDim/2$ 远小于 z ，这当然是不合理的，实际上也并非如此，但是从最终效果来看，这是一个好的假设。因此，不妨认为实际的模型并不是上述所说的球面中心投影模型，而是一个近似的更为奇异的模型，尽管不能很好地用图像来描述该模型，但当知晓该模型的投影变换关系后，这一点就不再重要了，即只需关心模型背后所暗含的投影关系。

根据上文所描述的近似模型，可以将式 (4-6) 简化为

$$x_d = r * (\arctan [(x - xDim/2)/z] + initCol * \pi/360) \quad (4-7)$$

最后根据投影前的坐标做双线性插值，即可得到投影后的图像。

基于上述球面中心投影模型，对如下图所示的全景图像 4-5 进行矫正（将海康威视 800 万像素全景拼接网络摄像机悬挂于球场边缘，场边距约 2m，高度约 6.4m，拍摄得到的全景图像）。



图 4-5 全景图像

对于图 4-5，设定 $angle = 135^\circ$ ， $xDim = 0.65w$ ， $yDim = 0.65h$ ， $init_col = 100^\circ$ ，得到的投影前坐标 x_d 每次增大一个固定值 $shift_col = 256$ ，得到的四幅“拼接前”图像如图 4-6 所示。

可以看到，尽管近场线不能完全拉直，但是远场线经过矫正后变得笔直，且

球场区域变得较为平整。

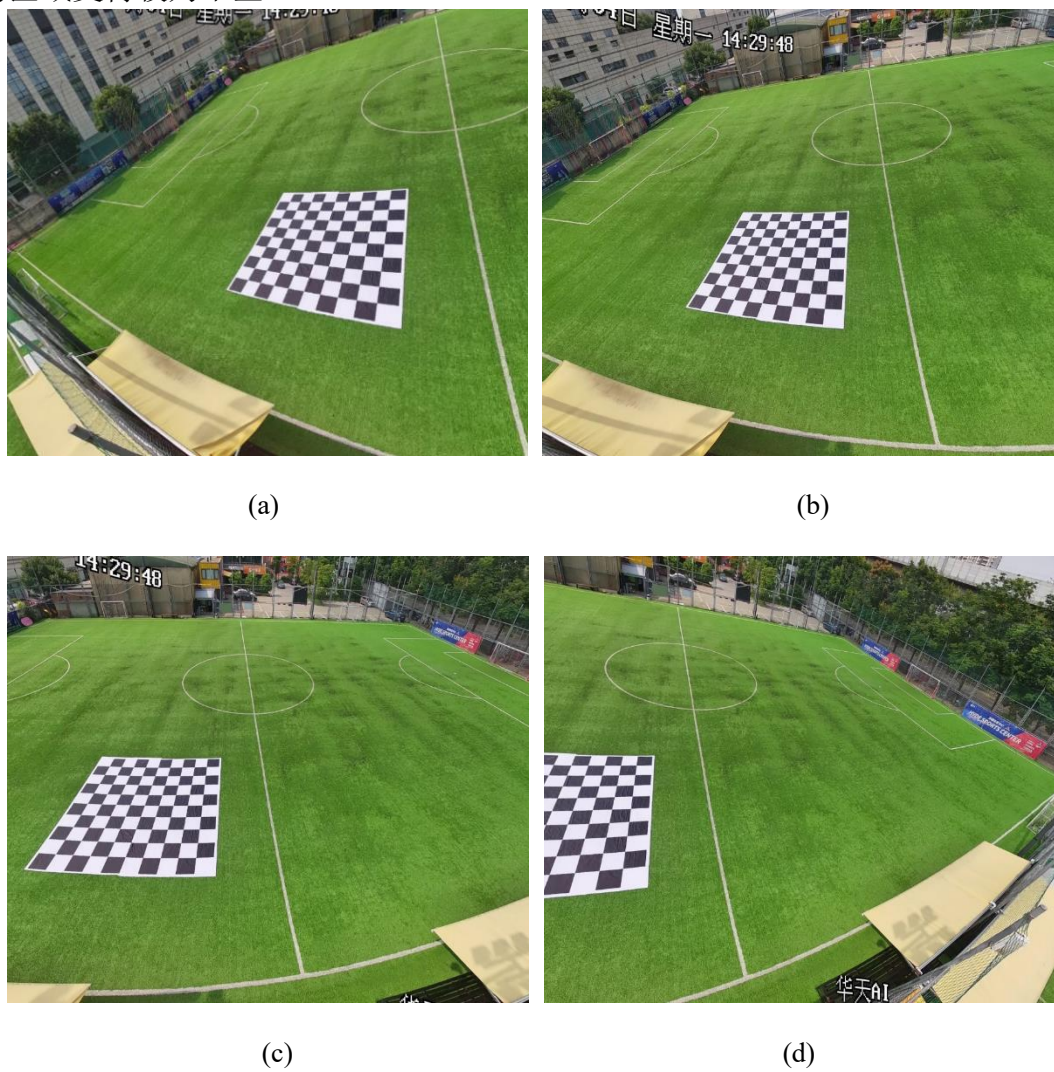


图 4-6 球面中心投影效果图

4.3 带倾斜角的球面中心投影

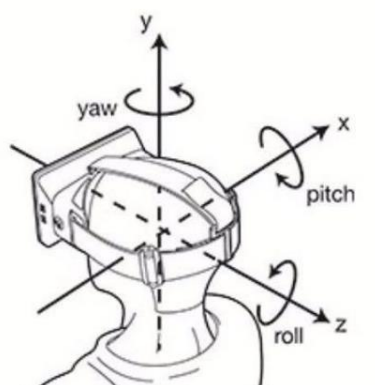


图 4-7 姿态角调整示意图

摄像头拍摄时可以调整姿态角，即偏航角（yaw）、翻滚角（roll）和俯仰角（pitch），如图 4-7 所示。

为了进一步增强球面中心投影模型的矫正能力，解除原图像关于过球心的水平大圆对称的假设，增加一个倾斜角因子 α ，表示原图 y 坐标为 $\alpha \cdot h$ 的像素在水平面处。显然，当 $\alpha=0.5$ 时，原图像关于过球心的水平大圆对称；当 $\alpha=0$ 时，原图像上边沿恰与水平面相交；当 $\alpha=1$ 时，原图像下边沿恰与水平面相交。

从姿态角来看，可以将 α 解释为俯仰角，initCol 解释为偏航角，而翻滚角不在本模型的考虑范围内。

增加倾斜角因子后的球面中心投影模型中平面与原图的对应关系（不显示球面）如图 4-8 所示，其中绿色弧线围成的区域代表原图像，用灰色平面来表示水平面，黑色线段所标明的向量即球心与原图像以及平面的连线，可以将橙色实线围成的区域看作原本未倾斜的状态下平面的位置，增加倾斜角因子后平面的位置如紫色实线围成的区域所示。

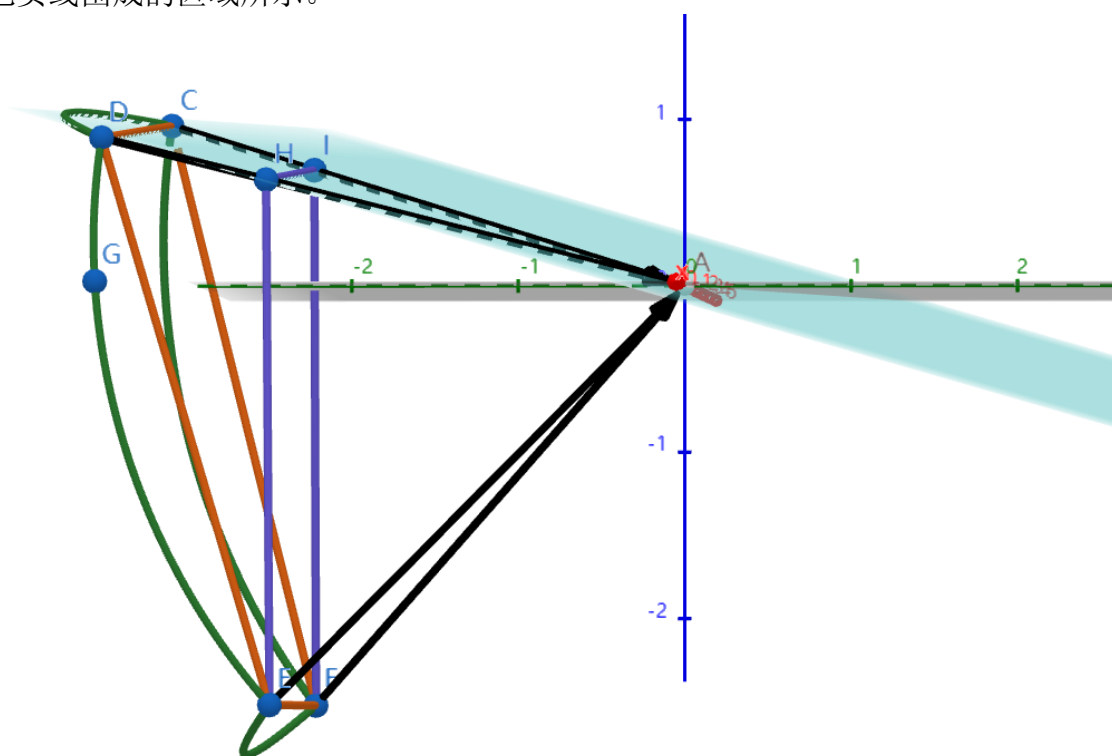


图 4-8 倾斜的模型中平面与原图的对应关系

同样地，记原图像大小为 (w, h) ，其沿 w 方向的张角为 angle ，内接平面大小为 $(x\text{Dim}, y\text{Dim})$ ，其相对于原图像的偏移为 initCol 。倾斜角因子为 α 。

根据原图像宽 w 对应的角度 angle 计算球面半径 r ，表达式同式 (4-1)。

根据图 4-8，很容易看出，球心到平面的距离 z 的表达式中与 $y\text{Dim}$ 有关的

项应当为 $yDim$ 被水平面分割后较大的部分, 记 $p = \max(\alpha, 1-\alpha)$, 则球心到平面的距离为

$$z = \sqrt{r^2 - (xDim/2)^2 - (yDim * p)^2} \quad (4-8)$$

记平面上一点为 $P(x, y)$, 要求出中心投影前点 P 在原图的对应点 $Q(xd, yd)$ 。记平面上一点为 $P'(x, yDim * \alpha)$, 其到球心的距离同式 (4-3)。

根据立体几何的知识, 不难证明 PP' 与水平面垂直, 则三角形 OPP' 是直角三角形, 据此可得直线 OP 与水平面的夹角 θ 的正切值 (注意点 P' 坐标与前文不同) 为

$$\tan \theta = ||pp'|| / ||op'|| = (y - yDim * \alpha) / \sqrt{(x - xDim/2)^2 + z^2} \quad (4-9)$$

则中心投影前点 P 在原图的对应点 Q 的纵坐标为

$$\begin{aligned} y_d &= r \times \arctan(\theta) + h * \alpha \\ &= r \times \arctan[(y - yDim * \alpha) / \sqrt{(x - xDim/2)^2 + z^2}] + h * \alpha \end{aligned} \quad (4-10)$$

x 和 xd 的变换关系同式 (4-7)。

显然当 $\alpha=0.5$ 时, 式 (4-8) 即式 (4-2), 式 (4-9) 即式 (4-4), 式 (4-10) 即式 (4-5), 这也进一步说明球面中心投影是带倾斜角因子的球面中心投影模型的一种特殊情况。

根据式 (4-7), 很容易知道平面的中央位置 ($x=xDim/2$) 对应于原图的横坐标为 $xd = r * \text{initCol} * \pi / 360$, 将式 (4-1) 代入得到 $xd = r * \text{initCol} * \pi / 360 = w/2 * \text{initCol} / \text{angle}$ 。

通过上述理论分析, 很容易发现当 $\text{initCol}=\text{angle}$ 时, 平面的中央位置对应于原图的中央位置, 本节中展示矫正效果将默认使用 $\text{initCol}=\text{angle}$ 这一条件, 并根据不同的拍摄图像适当调整 $xDim$ 和 $yDim$ 。

设定 $\text{angle}=155$, $\alpha=0.35$, 对全景图像 4-5 进行矫正, 效果如图 4-9 所示。

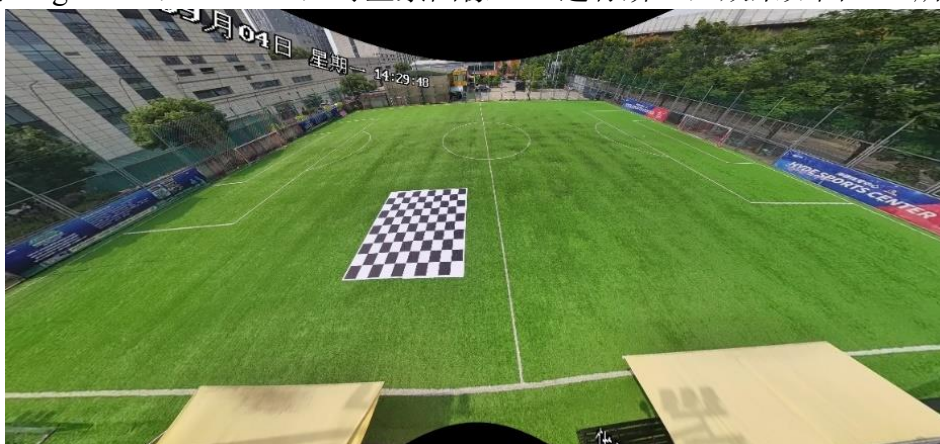


图 4-9 全景图像 4-5 矫正图

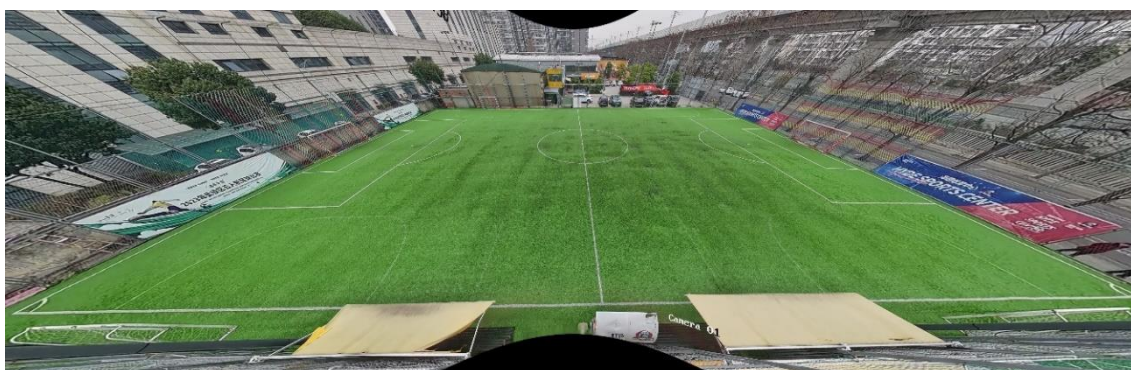
通过调节角度 angle 和倾斜角因子 α ，可以将处于不同高度及倾角的摄像头采集的全景图像进行鲁棒的矫正，全景图像（将海康威视 1600 万像素全景拼接网络摄像机悬挂于球场边缘，场边距约 2m，高度约 6.4m，拍摄得到的全景图像，两图像拍摄时俯仰角不同）及其矫正效果图如图 4-10 和图 4-11 所示。



(a)

(b)

图 4-10 原图



(a) $\text{angle}=165, \alpha=0.6$



(b) $\text{angle}=145, \alpha=0.5$

图 4-11 矫正图

4.4 本章小结

本章从全景图像的拼接方式入手，设计了一种基于像素重映射的新颖的图像矫正方法——球面中心投影，基于拼接的图像位于球面上的假设，过球心做中心投影，通过在内球面滑动一个内接于球面的平面，将原图逐次投影，分解得到“拼接前”的未畸变图像。

观察到摄像头还具有俯仰角，为了进一步增强球面中心投影模型的矫正能力，4.3 节提出了倾斜角因子的概念，通过应用倾斜角，增强的球面中心投影模型能够很好地对摄像头高度不同，场边距不同，俯仰角不同的全景图像进行矫正。

第五章 自标定的参数搜索

5.1 引言

上一章设计的球面中心模型通过调节角度 angle 和倾斜角因子 α ，能对不同的全景图像取得极佳的视觉效果。

然而，其需要的参数较多，即使令 initCol 与 angle 取值相同（参见第四章 4.3 节的分析），并在图像效果对 xDim 和 yDim 并不特别敏感的前提下（矫正效果主要取决于调节角度 angle 和倾斜角因子 α ， xDim 和 yDim 主要调整画面的尺寸，在其他参数确定的情况下，调整这两个参数且保证球心到平面的距离 z 大于 0 即可），依旧要手动探索 $\text{angle}(0\sim 180)$ 和 $\alpha(0\sim 1)$ 两者构成的参数空间，而且它们并不是离散的有限集合。

手动探索参数无疑费时费力，而且很难达到最佳的效果。有没有一个好的方法可以对每一幅全景图像的畸变特征进行检测，自动生成需要的矫正参数？

如果把问题看成一个回归问题，可以训练一个神经网络，其输入是一幅全景图像，通过各种网络结构对畸变特征进行提取，最后输出矫正参数。然而，这个简单的想法存在一些困难：

- 1) 需要拍摄大量的全景图像。足够数量的训练集才能够保证神经网络可以很好地学习到全景畸变的特征，并且防止过拟合。
- 2) 如何设计网络以提取关键的畸变特征
- 3) 如果采用监督学习，需要为每幅全景图像标注对应的矫正参数。

可以看到，上述方法纵使收集到了足够的数据，并且做了较为精确的标注，也无法保证有效性。

本章设计一种自标定的参数搜索方法，它把全景图像畸变矫正的问题转化为一个优化问题，对于任意一幅全景图像，提取其中的畸变特征来构建目标函数，通过最小化目标函数来确定模型的参数。

5.2 单目标的参数搜索

首先把畸变矫正的目标简化为，通过设定矫正参数，将图像中所有的畸变特征进行矫正（畸变特征暗含于本应为直线的曲线中，最直观地，矫正就是将歪曲的线条映射为直线）。如果能把这个优化目标表示成一个目标函数，那么全景图像畸变矫正的问题就自然而然的转化为了一个优化问题。

从最简单的一条曲线入手，对已经经过一定预处理的图像进行灰度化、二值化，根据阈值分离出目标的曲线，全景图像 4-5 的近场线提取如图 5-1 所示，其中白色亮点表示提取到的曲线点。



(a)



(b)

图 5-1 (a) 经过预处理用于提取的图像; (b) 提取到的曲线

在矫正参数已知的情况下，根据提取到的曲线点的坐标，可以计算出其矫正后的坐标。

这里与第四章中的投影过程有所不同，在球面中心投影模型中，遍历矫正后的坐标 (x, y) 并计算其对应的投影前的坐标 (x_d, y_d) ，然后进行双线性插值。而在参数搜索中，提取到的曲线点（畸变特征），其坐标是矫正前的坐标 (x_d, y_d) ，需

要重新推导以求得矫正后的坐标 (x, y) 。

为了参数搜索的便捷性和高效性，不妨将所有的系数归一化，即 $\text{angle}(0\sim 1)$, $\text{initCol}(0\sim 1)$, $\alpha(0\sim 1)$, 以及 $[xDim, yDim] = Dim \odot [w, h]$, 其中 Dim 是一个二维向量，元素的大小均在 $0\sim 1$ 之间。

同样地，记 $p = \max(\alpha, 1-\alpha)$, 引用第四章式 (4-1) 的结论，可以在参数归一化的情况下，计算得到球面半径

$$r = w / (\text{angle} * \pi) \quad (5-1)$$

球心到平面的距离参见式 (4-8)。

在参数归一化下，式 (4-7) 和式 (4-10) 表示为

$$xd = r * (\arctan [(x - xDim/2)/z] + \text{initCol} * \pi/2)$$

$$y_d = r \times \arctan [(y - yDim * \alpha) / \sqrt{(x - xDim/2)^2 + z^2}] + h * \alpha$$

则矫正的后的横坐标可以表示为

$$x = xDim/2 + z \tan (x_d/r - \text{initCol} * \pi/2) \quad (5-2)$$

矫正的后的纵坐标可以表示为

$$y = yDim * \alpha + \sqrt{(x - xDim/2)^2 + z^2} \tan ((y_d - h * \alpha)/r) \quad (5-3)$$

注意，式 (5-3) 中的 x 即式 (5-2) 的计算结果，联立式 (5-2) 和式 (5-3) 可以得到所有曲线点矫正后的坐标（尽管它们往往不是整数，甚至不一定是正数）。

固定参数 Dim （如 $Dim=(0.6, 0.6)$ ），将矫正参数 initCol 、 angle 和 α 设定为可学习参数，矫正后曲线的所有点应尽可能位于一条直线上。由于 initCol 可训练，不妨期望矫正后的所有点位于一条平行于 w 方向的直线上。

基于上述讨论，不假思索地以投影后 y 坐标的方差作为目标函数。记提取到的曲线像素点数量为 m ，则目标函数可以写为

$$\min_{\substack{\alpha \\ \text{angle} \\ \text{initCol}}} \left\{ \frac{1}{2m} \sum_{i=1}^m (y_i - \text{mean}(y))^2 \right\}$$

上述优化问题可以通过随机梯度下降来迭代求解。这里引入深度学习的经典框架 `Pytorch`，通过继承 `nn.module` 的自定义层来实现式 (5-2) 和式 (5-3) 所表示的前向传播。自定义层的参数即矫正参数，其中 initCol 、 angle 和 α 为可学习参数，其输入为 $m \times 2$ 矩阵（其中 m 为曲线的像素数量，第一列为 xd ，第二列为 yd ），输出为 $m \times 1$ 的向量 y 。

添加一个系数 $1/\sqrt{h*w}$ ，将归一化目标函数作为损失函数，如下式所示。

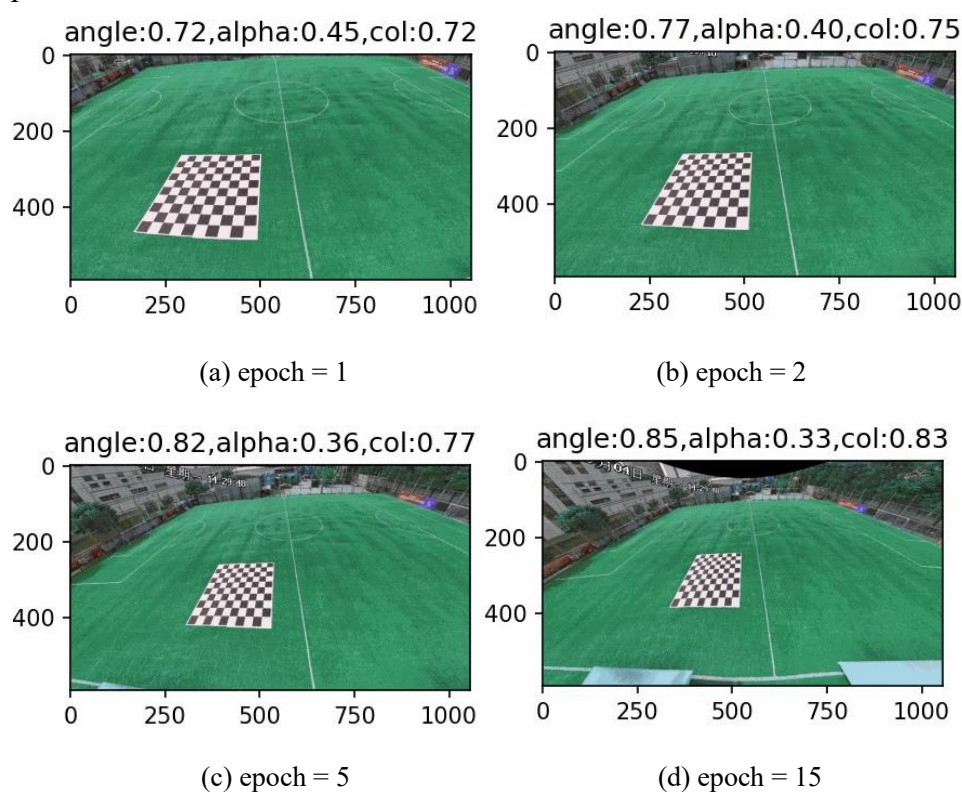
$$L = \frac{1}{2m\sqrt{h*w}} \sum_{i=1}^m (y_i - \text{mean}(y))^2 \quad (5-4)$$

优化算法为 `Adam`，全称为自适应时刻估计法 (Adaptive Moment Estimation)。

这种方法的独特之处在于它为每个参数的优化存储了先前平方梯度的指数衰减平均值（称为第二矩估计）和先前梯度的指数衰减均值（称为第一矩估计）。通过调整每个参数的学习率，Adam 算法可以对不同的参数应用不同的学习步长，具有自适应性。与其他自适应学习率算法相比（比如 AdaGrad 和 RMSProp），Adam 算法的收敛速度更快，这方面的性能特性使其在训练深度神经网络时效果更佳。同时，这个优化方法还可以有效地纠正一些常见的优化问题，比如梯度消失、慢收敛以及参数更新的高方差导致的损失函数波动大等问题。因此，在众多优化算法中，Adam 被广泛使用并且成效显著。

Adam 参数除学习率外采用固定值，无权重衰减。之所以不采用 SGD，原因在于 SGD 初始需要设置较小的学习率，否则容易发生梯度爆炸，使得参数变成 nan 无法被训练，而学习率太小又会无法收敛。

利用式 (5-4) 作为 loss 函数，对全景图像 4-5 进行矫正参数的搜索，在不同的 epoch 下的矫正参数以及相应的矫正结果如图 5-2 所示。



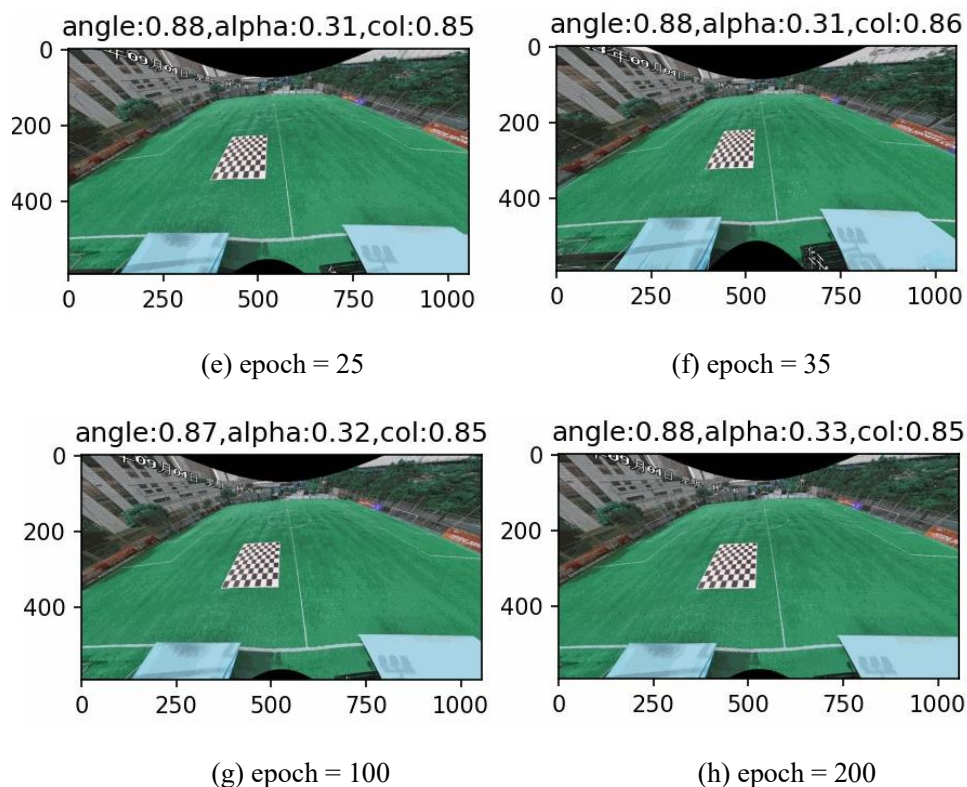


图 5-2 各 epoch 下优化得到的矫正参数以及相应的矫正图像

上图中设定 $\text{Dim}=[0.55, 0.55]$ 时，损失（即式 (5-4) 的计算结果）逐渐收敛到 0。如下图 5-3 (a) 所示。

然而，当设定 $\text{Dim}=[0.65, 0.65]$ 时，随着训练的进行，梯度和参数突变为 nan，发生了梯度爆炸。Loss 的变化如图 5-3 (b) 所示。

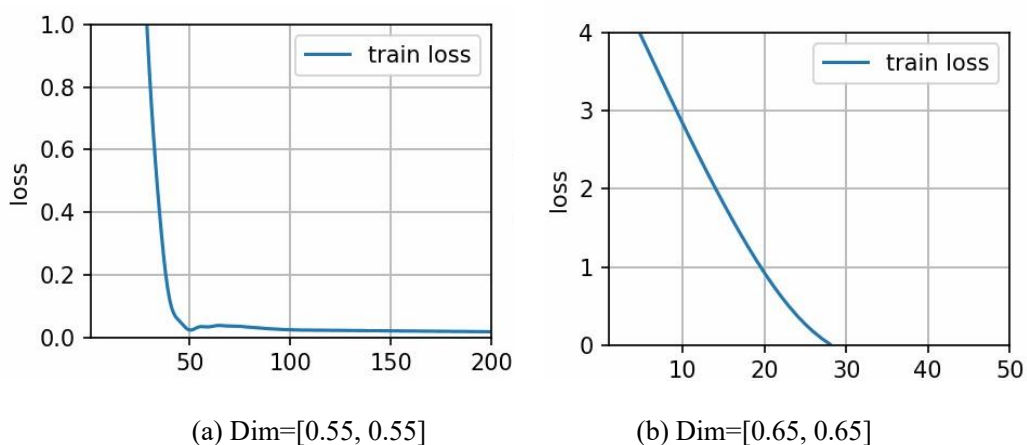


图 5-3 不同 Dim 下的损失曲线

通过多次调试，最终发现网络趋向于减小 z 来最小化目标函数（参考式 (5-3)，

z 越小, y 的方差就越小), 当 $z^2 < 0$ 时, 就会出现上述问题, 无法继续进行优化。

通过减小 z 来最小化目标函数, 允许这一优化方式的同时, 又必须避免过度依赖它而掉入了不期望的局部极小值中。一个简单有效的方法是增加惩罚项 $-\log(z/s)$, 其中 $s = \sqrt{h*w}$ 。当 z 与 s 比值过于小甚至逼近于 0 时, 会导致损失急剧上升, 通过反向传播的梯度对矫正参数进行迭代, 会使得 z 增大, 从而避免上述问题。

进一步地, 对倾斜角因子加以限制, 增加惩罚项 $-\log(1-\alpha) - \log(\alpha)$ 。类似于上面的理由, 倾斜角因子过大或过小都是所不期望的局部极小值, 过大或过小都将不得不缩减最终得到的平面矫正图像的大小 (即参数 Dim), 否则会导致 $z^2 < 0$ 。

综合以上论述, 将本节所讨论的参数搜索方法称为单目标搜索, 结合式 (5-4), 得到单目标搜索的损失函数如下

$$L = \left\{ \frac{1}{2m\sqrt{h*w}} \sum_{i=1}^m (y_i - \text{mean}(y))^2 \right\} - a \ln(z/\sqrt{h*w}) - b(\ln(\alpha) + \ln(1-\alpha)) \quad (5-5)$$

其中 a 和 b 都是超参数, 在后面的实验中设定 $a=b=1$ 。

设定 $Dim=[0.65, 0.65]$, 利用式 (5-5) 作为 loss 函数, 对全景图像 4-5 进行矫正参数的搜索, 最终的矫正参数以及相应的矫正结果如图 5-4 所示。

可以看到, 倾斜角因子 $\alpha = 0.39$, 相比于图 5-2(h) 的结果更加接近于 0.5, 整体上的矫正效果也更加好。图 5-4 的矫正结果与图 4-8 手动搜索参数矫正全景图像 4-5 的结果进行对比, 本节设计的单目标搜索耗时更短 (Intel(R) Core(TM) i7-10510U CPU @ 1.80GHz 训练单曲线共计 6760 像素点 200 epoch 耗时不到 1 分钟), 搜索得到的参数更精确 (矫正参数的效果更好)。

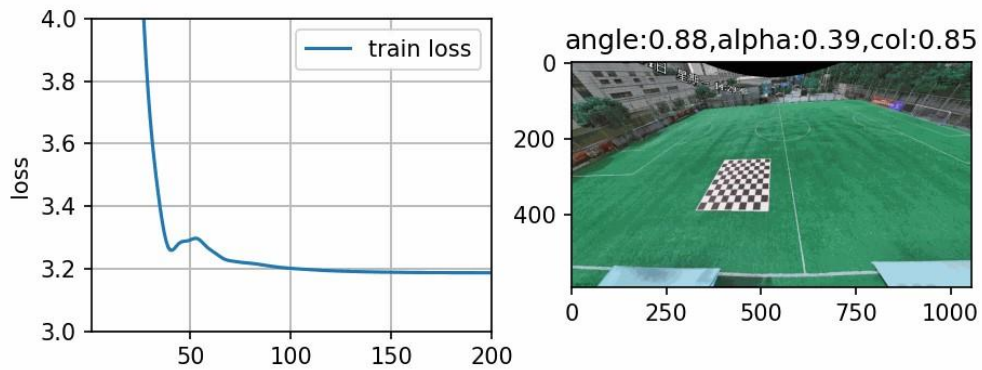


图 5-4 全景图像 4-5 单目标搜索结果

5.3 多目标搜索

尽管单目标搜索已经能够对参数进行有效的搜索，但不得不承认的是，它存在不能克服的致命缺陷——过分关注于采样曲线代表的局部畸变特征，而忽视了全景图像整体的畸变特征。

观察另外一个例子。使用单目标搜索对全景图像 4-9(a) 进行矫正，采样的曲线为近场线，在不同的 epoch 下的矫正参数以及相应的矫正结果如图 5-5 所示。

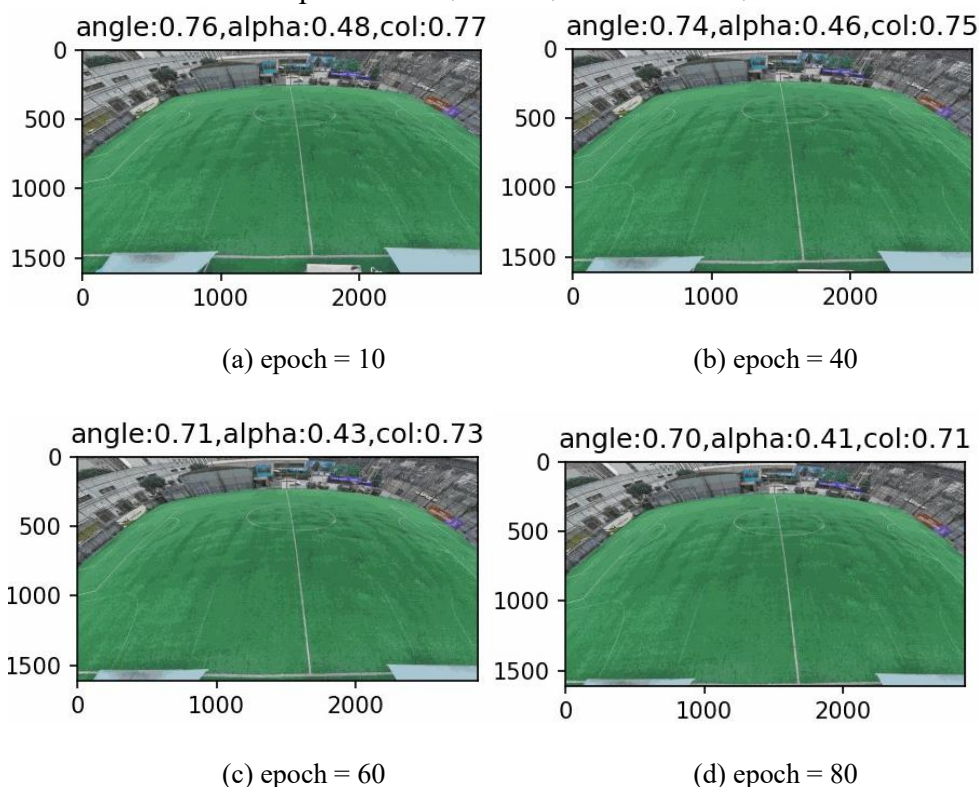


图 5-5 全景图像 4-9(a) 单目标搜索各 epoch 下的矫正结果

对应的损失曲线如图 5-6 所示。可以看到，整个训练过程中，近场线都被矫正成几乎笔直的曲线，这对于只关注单曲线的单目标搜索来说是理所应当的，然而这并非所希望看到的。

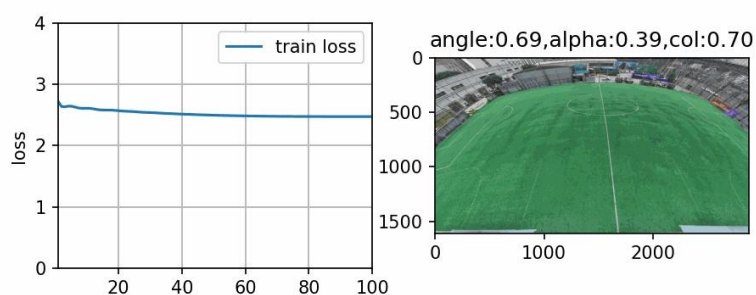


图 5-6 全景图像 4-9(a) 单目标搜索结果

解决这一问题的关键在于，如何在搜索时更多地关注整体的畸变特征，而不只是经过采样的局部特征。一个简单可行的方法是采样若干条曲线，只要采样的曲线在全景图像的分布足够均匀，数量足够多，那么通过采样获取到的畸变特征就能足够近似全景图像整体的畸变特征。

值得注意的是，不同的曲线，其像素点数量不尽相同，经过矫正后的中心位置不同，因此不能简单地将所有提取的像素点打乱后进行前向传播，并计算方差。借助 `torch.utils.data.Dataset`，可以设计一个数据集，其中一条曲线就是一个训练样本。不妨将曲线的像素点看成特征，尽管它们的特征数不尽相同。设定 `batch_size=1`，这样能保证每次都只对同属于一条曲线的特征进行前向传播和计算方差。

将每条曲线矫正后得到的方差的均值进行累加，加上上一节中讨论的惩罚项后再反向传播。

综合以上论述，将本节所讨论的参数搜索方法称为多目标搜索，结合式 (5-5)，得到多目标搜索的损失函数如下

$$L = \frac{1}{P} \sum_{p=1}^P \left\{ \frac{1}{2m_p \sqrt{h^*w}} \sum_{i=1}^{m_p} (y_{pi} - \text{mean}(y_p))^2 \right\} - a \ln(z/\sqrt{h^*w}) - b(\ln(\alpha) + \ln(1 - \alpha)) \quad (5-6)$$

其中 P 表示采样曲线的数量， m_p 表示第 p 条曲线的像素点数量。

将多目标搜索算法称为球面中心投影参数搜索，其算法伪代码如下。矫正参数初始化可自行设定，建议 `angle > 90`，`initCol = angle`，`alpha = 0.5`。

算法：球面中心投影参数搜索算法

算法：球面中心投影参数搜索

输入：全景图像（以及若干用于采样曲线的掩膜）

输出：球面中心投影模型矫正参数

1. 对掩膜执行灰度化、二值化，根据阈值分离出目标曲线的像素点，每条曲线的所有像素点构成 `data_iter` 的一个样本，设定 `data_iter` 的 `batch_size` 为 1；
2. 设定平面图大小系数 `Dim`，初始化矫正参数 `angle`, `alpha`, `initCol`。
3. **for** `epoch` in `[1, 2, ..., epochs]` **do**
4. $L \leftarrow 0$ ，清除矫正参数的梯度
5. **for** `i` in `[1, 2, ..., P]` **do**
6. $L += \text{loss}(\text{net}(\text{data_iter}[\text{i}]))$
7. **end for**
8. 将损失 L 对 P 条曲线取平均值后加上惩罚项，再进行反向传播

9. 调用优化器更新矫正参数

10. end for

11. Return (angle, alpha, initCol)

对于全景图像 4-9(a), 采样两条曲线——近场线和远场线, 采样结果如图 5-7 所示, 其中亮点表示采样到的像素点。显然, 远场线的像素点数量远少于近场线。

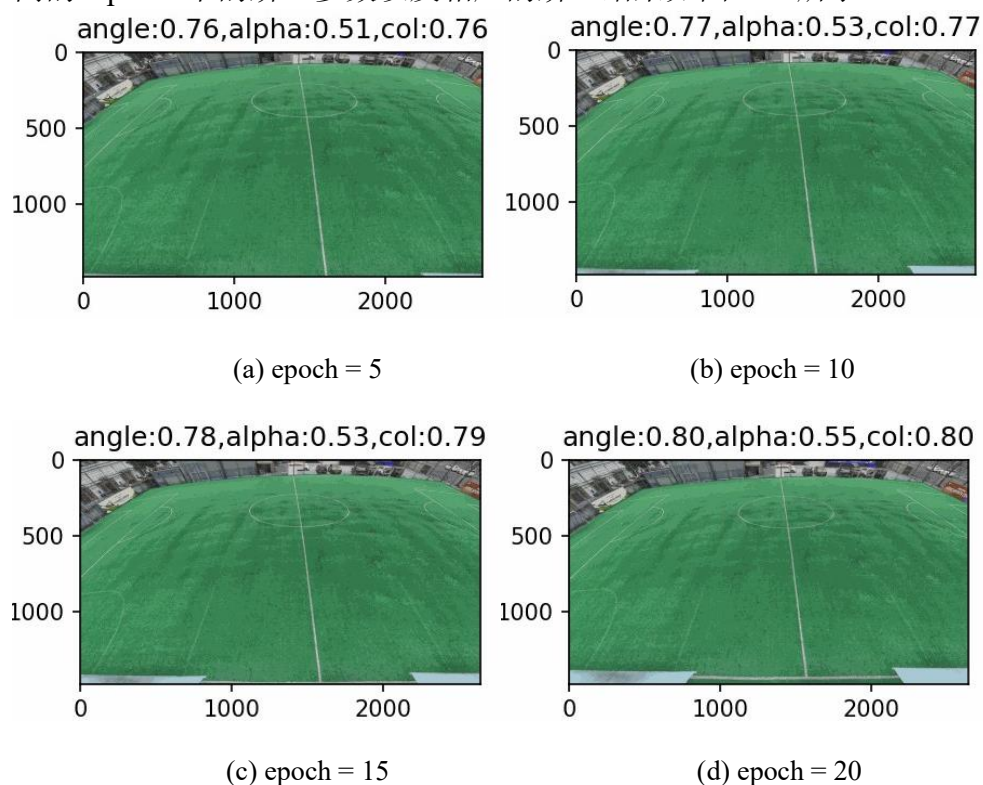


(a)

(b)

图 5-7 全景图像 4-9(a) 采样结果 (a) 近场线; (b) 远场线

设定 $a=b=1$, $\text{Dim}=[0.65, 0.65]$, 使用多目标搜索对全景图像 4-9(a) 进行矫正, 在不同的 epoch 下的矫正参数以及相应的矫正结果如图 5-8 所示。



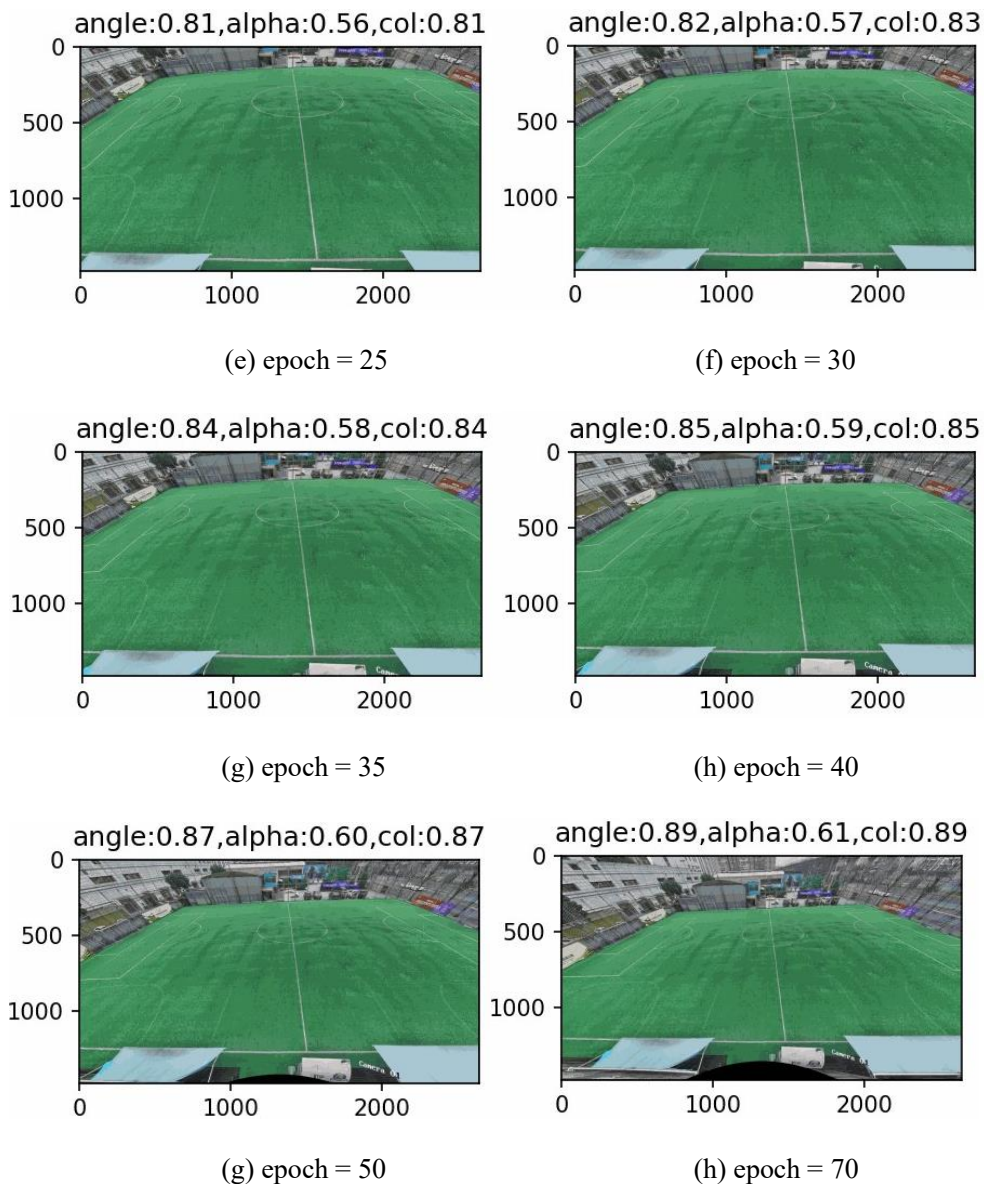


图 5-8 全景图像 4-9(a) 多目标搜索各 epoch 下的矫正结果

能看到在搜索矫正参数时会同时关注两条曲线。

利用式 (5-6) 作为 loss 函数，对全景图像 4-9(a) 进行矫正参数的搜索，最终的矫正参数以及相应的矫正结果如图 5-9 所示。

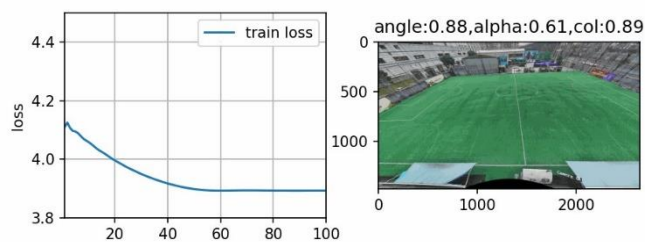


图 5-9 全景图像 4-9(a) 多目标搜索结果

5.4 本章小结

本章设计了一种新颖的参数搜索方法，不需要大量的数据及相应的标签，仅提取曲线的像素点作为畸变特征。通过重新推导上一章节设计的球面中心投影模型的映射关系，实现网络的前向传播，借助深度学习框架 Pytorch 的自动微分实现网络的反向传播，并使用 Adam 作为优化算法来更新矫正参数。

5.3 节进一步讨论了单曲线的局限性，设计通过多次采样全景图像各局部的畸变特征来近似整体的畸变特征，取得了极好的效果。

将这种基于球面中心投影，只利用自身特征对矫正参数进行标定的算法称为球面中心投影参数搜索算法。相比于手动搜索，其最大的优势就是收敛速度足够快，在经过一定 epoch 的训练后，损失逐渐趋于稳定，矫正参数收敛。此外，对于非离散参数，搜索得到的参数精度足够高，效果足够好。