



西安交通大学  
XI'AN JIAOTONG UNIVERSITY

## 本科实验报告

光栅方程的应用

课程名称: 基础物理实验

姓名:



学院:

少年班

专业:

少年班 2404

学号:



指导老师:

曹子君

2026 年 5 月 6 日

# 西安交通大学实验报告

专业：少年班 2404  
姓名：[REDACTED]  
学号：[REDACTED]  
日期：2026 年 5 月 6 日  
地点：仲英楼 B503

课程名称：基础物理实验  
实验名称：光栅方程的应用  
指导老师：曹子君  
实验类型：测量实验  
成绩：  
同组学生姓名：[REDACTED]

## 一、用米尺测量激光波长

### 1. 数据处理

本实验使用光栅常数为  $d = 1/300\text{mm}$  的透射光栅对绿色激光进行衍射，用米尺在接收屏上测量各级衍射光斑的位置。光栅到接收屏的垂直距离为  $L = 143.60\text{cm}$ ，米尺精度为  $1\text{mm}$ ，测量时进行了估读。实验环境温度为  $27^\circ\text{C}$ ，相对湿度  $69\%$ 。本组测量对象为绿色激光（标称波长  $532\text{nm}$ ），采用垂直入射方式。

每组衍射光斑的位置记录于表 1。根据光栅方程  $d\sin\theta_k = k\lambda$ ，可导出每个级次对应的波长：

$$\lambda_k = \frac{d}{k} \sin\theta_k = \frac{d}{k} \frac{x_k}{\sqrt{L^2 + x_k^2}}$$

其中  $x_k$  为第  $k$  级衍射光斑与零级中心之间的距离， $k = \pm 1, \pm 2, \pm 3$ 。

| 测量级数 $k$        | -3     | -2     | -1     | 0    | +1    | +2    | +3    |
|-----------------|--------|--------|--------|------|-------|-------|-------|
| $x_k/\text{cm}$ | -77.86 | -48.16 | -23.04 | 0.00 | 23.13 | 48.20 | 78.45 |

表 1: 绿色激光各级衍射光斑位置（光栅常数  $d = 1/300\text{mm}$ ， $L = 143.60\text{cm}$ ）

利用 Python 编写计算程序，对六组有效数据（ $k = \pm 1, \pm 2, \pm 3$ ）分别计算波长。计算过程中使用精确公式  $d\sin\theta = k\lambda$ ，不引入小角度近似， $\sin\theta$  按  $\sin\theta_k = x_k/\sqrt{L^2 + x_k^2}$  严格计算。计算结果汇总于表 2。

| $k$ | $\sin\theta_k$ | $\lambda_k/\text{nm}$ |
|-----|----------------|-----------------------|
| -3  | -0.4766        | 529.61                |
| -2  | -0.3180        | 529.95                |
| -1  | -0.1584        | 528.07                |
| +1  | 0.1590         | 530.08                |
| +2  | 0.3182         | 530.35                |
| +3  | 0.4794         | 532.70                |

表 2: 各级次对应的波长计算结果

六组测量值的平均波长为  $\bar{\lambda} = 530.12 \text{ nm}$ ，标准差  $\sigma_{\lambda} = 1.50 \text{ nm}$ 。实验测得的波长值与标称值  $532 \text{ nm}$  的相对误差为  $-0.35\%$ ，绝对误差约为  $-1.88 \text{ nm}$ 。

从表 2 中可以看出，六个级次计算出的波长值在  $528 \text{ nm}$  到  $533 \text{ nm}$  之间波动，波动范围约  $4.6 \text{ nm}$ 。其中  $k = +3$  级得到的波长 ( $532.70 \text{ nm}$ ) 最接近标称值， $k = -1$  级的结果 ( $528.07 \text{ nm}$ ) 偏离最大。正负对称级次的衍射光斑位置存在轻微不对称性： $k = \pm 1$  的不对称度约为  $0.39\%$ ， $k = \pm 2$  约  $0.08\%$ ， $k = \pm 3$  约  $0.75\%$ 。在理想垂直入射条件下，正负对称级次的光斑位置应对称分布，实际测量中出现的这些微小不对称说明光栅平面与入射光方向并非严格垂直，或者接收屏与光栅平面之间存在微小的不平行。

## 2. 误差分析与改进建议

本次测量的误差来源可以归纳为几个方面。从仪器层面来看，米尺的最小刻度为  $1 \text{ mm}$ ，且由人眼估读，带来约  $0.5 \text{ mm}$  的读取不确定度。光栅到接收屏的距离  $L$  同样用米尺测量，也存在类似的不确定度。此外光栅常数  $d = 1/300 \text{ mm}$  是标称值，其制造公差也会贡献系统误差。

从实验操作层面来看，激光是否垂直入射到光栅平面是影响对称性的关键因素。实验中仅凭目测调整垂直度，难以保证精确的垂直入射条件。接收屏与光栅平面之间的平行度也未做严格校准。零级光斑位置的确定——即  $x = 0$  的参考点——本身也存在定位误差。光斑本身有一定宽度，用肉眼判断光斑中心位置时引入额外判读误差。

从测量环境来看， $27^\circ\text{C}$  的室温与  $69\%$  的相对湿度对光学测量影响有限，但空气中的尘埃散射以及环境杂散光可能使光斑边缘变得模糊，加大判读难度。

改进方案可以从多个角度入手。采用精度更高的测量工具（例如数显游标卡尺代替米尺测量光斑位置），可以有效减小读数误差。在光路调整方面，可利用反射法验证激光是否垂直入射：将光栅旋转  $180^\circ$ ，观察反射光点是否回到激光器出口，以此判断垂直度。增加测量次数并取平均，能够降低随机误差。对于光斑中心的判读，可以在暗室条件下操作以提高光斑对比度，同时采用坐标纸作为接收屏，利用网格线辅助定位。值得注意的是，高级次衍射 ( $|k|$  较大) 对应的光斑离中心较远，测量相对误差更小，因此可以优先使用  $|k| \geq 2$  的数据计算波长并赋予更大权重。

## 二、 CD 光盘光轨间距的测量

### 1. 数据处理

CD 光盘的存储方式是将数据以螺旋形轨道刻录在反射层上。相邻轨道之间的间距（光轨间距）通常在微米量级，当激光照射到光盘表面时，这些周期性排列的轨道相当于一个反射式光栅，产生衍射现象。通过测量衍射光斑的分布，可以反推光轨间距。

本实验使用绿色激光（波长  $\lambda = 532 \text{ nm}$ ）垂直入射到 CD 光盘表面，光盘到接收屏的距离为  $L = 135.71 \text{ cm}$ 。衍射光斑的位置使用米尺测量并估读，原始数据见表 3。

| 测量级数 $k$        | -2      | -1     | 0    | +1    | +2     |
|-----------------|---------|--------|------|-------|--------|
| $x_k/\text{cm}$ | -141.91 | -52.88 | 0.00 | 52.49 | 139.39 |

表 3: CD 光盘各级衍射光斑位置 ( $L = 135.71 \text{ cm}$ )

根据光栅方程  $d \sin \theta_k = k\lambda$ ，在已知波长  $\lambda$  和测量得到的光斑位置  $x_k$  后，光轨间距可由下式计算：

$$d_k = \frac{|k|\lambda}{\sin \theta_k} = \frac{|k|\lambda \sqrt{L^2 + x_k^2}}{|x_k|}$$

利用 Python 对四级有效数据 ( $k = \pm 1, \pm 2$ ) 分别计算光轨间距，结果列于表 4。

| $k$ | $\sin \theta_k$ | $d_k/\mu\text{m}$ |
|-----|-----------------|-------------------|
| -2  | 0.7227          | 1.4722            |
| -1  | 0.3631          | 1.4653            |
| +1  | 0.3607          | 1.4748            |
| +2  | 0.7165          | 1.4850            |

表 4: 各级次对应的 CD 光轨间距计算结果

四级数据给出的光轨间距平均值为  $\bar{d} = 1.4743 \mu\text{m}$ ，标准差  $\sigma_d = 0.0082 \mu\text{m}$ 。标准的 CD 光轨间距约为  $1.6 \mu\text{m}$ ，本次测量值与标准值的相对误差为  $-7.86\%$ 。

从正负级次的对称性来看， $k = \pm 1$  的不对称度约为  $-0.74\%$ ， $k = \pm 2$  的不对称度约为  $-1.78\%$ 。值得注意的是， $k = \pm 2$  级对应的衍射角较大 ( $\theta \approx 46^\circ$ )，此时  $\sin \theta$  与  $\tan \theta$  的差异已非常显著 ( $\tan \theta_2 \approx 1.046$ ， $\sin \theta_2 \approx 0.723$ ，二者相差约  $44.7\%$ )，若在计算中误用  $\tan \theta$  近似代替  $\sin \theta$  将导致严重错误。这也提醒我们，在本实验中因为衍射角较大，不能采用小角度近似，必须使用严格的  $\sin \theta$  公式。

## 2. 误差分析与改进建议

造成实验结果与标准值之间约  $7.86\%$  偏差的原因主要有几个方面。首先，我们直接使用了标称波长  $532 \text{ nm}$  参与计算，而实验一中实测波长均值为  $530.12 \text{ nm}$ ，若改用实测波长计算，光轨间距结果将变为约  $1.469 \mu\text{m}$ ，相对误差微调至约  $-8.2\%$ ，变化很小，说明波长本身的偏差不是主要误差来源。更为关键的因素在于衍射光斑位置的测量：CD 产生的衍射光斑较大（尤其在高级次），光斑中心位置的判读带有相当的主观性，由此引入的误差可能达到数毫米甚至更大。光盘表面的划痕、灰尘以及不平整度也会改变局部衍射条件。

距离  $L$  的测量同样存在不确定性。米尺用于测量  $L$  时需确保尺子与光盘平面垂直，实际操作中这一要求不易精确满足， $1 \text{ mm}$  量级的偏差会对结果产生大约  $0.1\%$  的影响。另外，光盘表面并非理想刚性平面，在固定过程中可能发生微小弯曲，使得不同区域的入射角和衍射条件略有不同。

改进措施可以从硬件和操作两方面考虑。一方面，可使用表面质量更好的新光盘，并在暗室条件下进行操作以减少杂散光干扰，让光斑更加清晰。固定光盘时应使用平整的夹具，避免光盘弯曲。另一方面，为了提高光斑中心判读的准确性，可在接收屏上附坐标纸，读数时通过网格定位光斑中心而非全靠目测。还可以在不同的  $L$  值下重复实验，验证结果的重复性。更进一步，可以尝试使用不同级次组合（如仅用  $k = \pm 1$  的低级次数据）来减小大角度衍射带来的系统误差，因为低衍射角下光斑更集中、边缘更清晰。

## 三、手机分辨率的测量

### 1. 实验原理简述

现代智能手机（如本实验采用的红米 70）的 AMOLED 屏幕像素通常不再是传统的正交矩形排列，而是采用 Pentile 钻石排列或 Delta 三角形排列等复杂的二维周期性结构。当一束相干激光照射在手机

屏幕上时，屏幕表面极为密集的像素矩阵等效于一个二维反射相控光栅。

本实验采用斜入射反射衍射光路：手机水平放置于地面，坐标纸固定于竖直墙面，激光以入射角  $\theta_i = 30^\circ$  照射手机屏幕，衍射光斑投射至墙面。在此三维几何结构中，无法使用简单的一维光栅方程分离水平和竖直方向，需要引入固体物理中的二维倒易点阵模型。

设像素排列的二维物理空间原胞基矢为  $\vec{a}_1$  和  $\vec{a}_2$ 。墙面上衍射光斑坐标对应的空间方向余弦可映射为倒易空间中的波矢变化量  $\Delta\vec{u}$ 。对于任意级次  $(m, n)$  的光斑，满足 Laue 方程：

$$\Delta\vec{u}_{m,n} = m\vec{g}_1 + n\vec{g}_2$$

其中  $\vec{g}_1, \vec{g}_2$  为倒易点阵基矢。测出墙面上各个衍射光斑的坐标后，即可通过矩阵方程解算出倒易基矢，进一步通过逆矩阵变换  $\vec{a}_i \cdot \vec{g}_j = \lambda\delta_{ij}$ ，求得真实的物理原胞基矢大小  $a_1, a_2$ ，其反映了像素间的最短周期性距离。已知屏幕物理长  $a$ 、宽  $b$ ，若以物理基矢长度作为有效像素间距进行估算，即可求出屏幕的有效分辨率  $N_h \times N_v$ 。

## 2. 使用的仪器

绿色激光器（标称波长  $\lambda = 532 \text{ nm}$ ），待测手机（红米 70，官方标称分辨率  $3200 \times 1440$ ），网格坐标纸，数码相机（或带拍照功能的手机用于图像采集），米尺，计算机及 Python 数据处理环境（集成 OpenCV 视觉库与 NumPy 矩阵运算库）。

## 3. 实验步骤简述

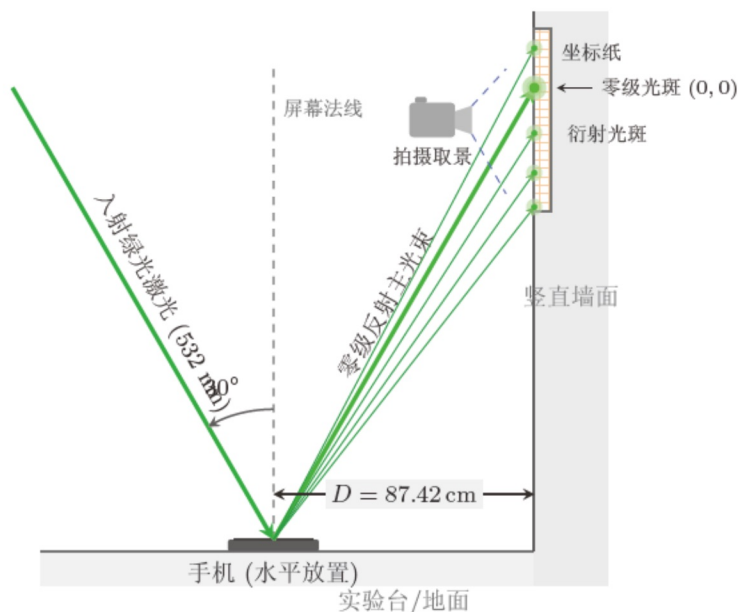


图 1: 手机像素分辨率测量实验光路示意图

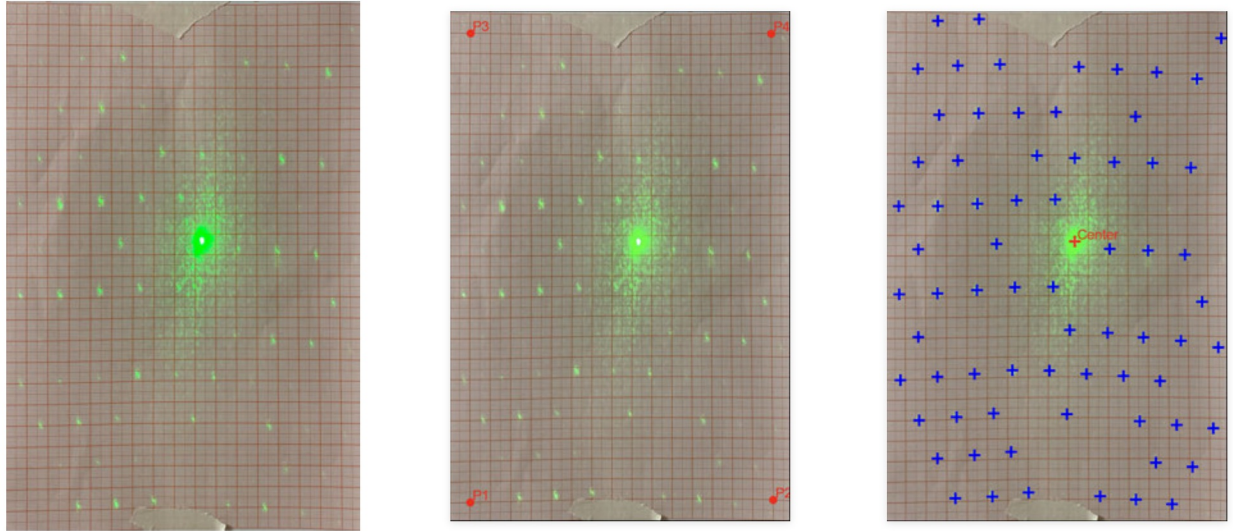
我们首先把待测的红米 70 手机水平放置在地面上并熄屏，利用玻璃基板与像素金属走线层的反射面工作。在距离手机  $D = 87.42 \text{ cm}$  的竖直墙面上，平整地贴好网格坐标纸作为接收屏——坐标纸上的网格线为后续定位光斑提供参照。接下来把  $532 \text{ nm}$  绿色激光器安装在支架上，从手机上方斜入射，一边观察墙面上的反射光斑一边微调角度，使激光与手机屏幕法向之间的夹角稳定在  $\theta_i = 30^\circ$ 。角度调

好后，零级反射光斑连同周围各级二维衍射光斑阵列便清晰地投射在墙面的坐标纸上。实验光路如图 1 所示。我们用相机对准墙面拍摄，尽量让相机光轴垂直于墙面以减少透视变形，照片须包含完整的坐标网格和衍射光斑。拍摄完毕后，用游标卡尺测量手机屏幕发光区域的长  $a$  和宽  $b$ 。

数据处理环节，我们将拍摄的照片导入事先写好的 Python 程序中。

程序算法针对斜入射条件下手机屏幕复杂二维像素阵列的衍射图样，融合透视几何校正与固体物理模型，实现从图像光斑到晶格参数的自动提取。为消除拍摄时因相机光轴不与墙面垂直所导致的透视畸变，首先在坐标纸上人工标记至少四个网格交点，记录其图像像素坐标与对应的真实物理坐标，通过求解一个具有 8 个自由度的  $3 \times 3$  单应性矩阵  $H$ ，将全部光斑的像素齐次坐标  $\mathbf{p}_{img}$  精确映射至无透视的墙面物理齐次坐标  $\mathbf{p}_{real}$ ，满足关系  $c\mathbf{p}_{real} = H\mathbf{p}_{img}$ ，其中  $c$  为尺度因子。该矩阵由最小二乘法计算得到，后续所有分析均在此正交校正后的物理坐标系下进行。

完成透视校正后，需要将墙面上的三维衍射光斑转化为反映倒易空间线性关系的二维坐标。设手机屏幕位于水平面，墙面竖直且与屏幕距离为  $D$ ，激光以入射角  $\theta_i$  斜入射。对墙面上任一光斑的物理坐标  $(X_w, Z_w)$ ，其光程距离为  $R = \sqrt{X_w^2 + D^2 + Z_w^2}$ ，出射光线方向余弦为  $(X_w/R, D/R, Z_w/R)$ ，而入射光方向余弦为  $(0, \sin \theta_i, -\cos \theta_i)$ 。根据二维光栅的劳厄方程，平行于屏幕平面方向的波矢变化等于倒易格矢  $\mathbf{G}$  乘以波长  $\lambda$ ，于是得到倒易空间坐标  $(u, v)$ ： $u = X_w/R$ ， $v = D/R - \sin \theta_i$ ，满足  $(u, v) = \lambda(g_x, g_y)$ 。这一映射将墙面上受斜入射几何影响的非线性位置，转化为倒易平面内由倒格矢线性叠加的规则点阵。



(a) 原始衍射图样

(b) 透视矫正界面

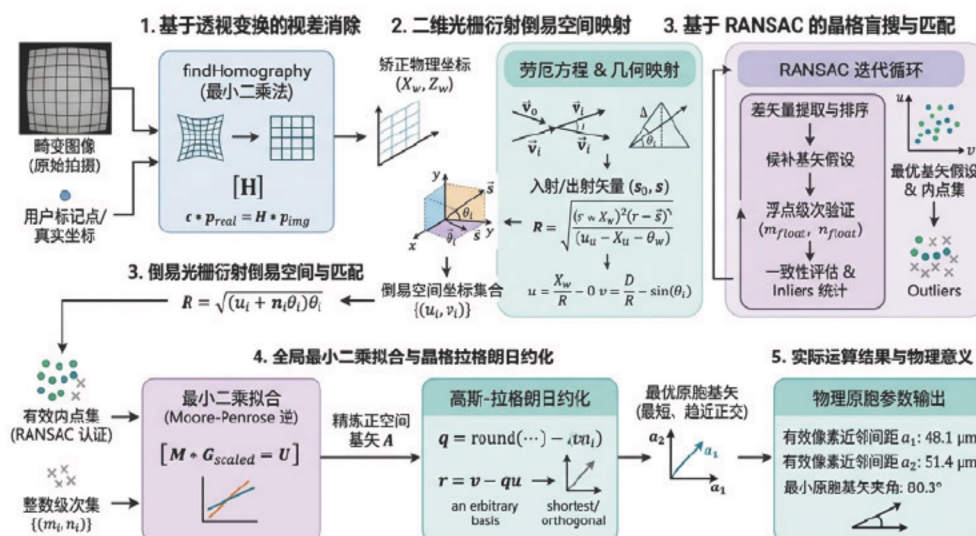
(c) 衍射光斑坐标标定界面

图 2: 程序处理界面

在倒易空间中，所有有效衍射斑点的坐标均应表示为两个倒易基矢  $\mathbf{g}_1, \mathbf{g}_2$  的整数线性组合  $(u, v) = m\lambda\mathbf{g}_1 + n\lambda\mathbf{g}_2$ 。算法采用 RANSAC 盲搜策略自动确定最优基矢并匹配级次：首先计算所有光斑两两之间的倒易差矢量，按长度递增排序，最短的那些非共线矢量成为候选基矢。对每一对候选基矢构成的矩阵  $G_{test}$ ，反算各光斑的浮点数级次  $(m_{float}, n_{float}) = (\lambda G_{test})^{-1}(u, v)$ ，并统计与整数偏差小于预设容差（如 0.15）的内点数量。遍历全部候选基矢对后，保留能够解释最多光斑的基矢，同时将无法用该基矢的整数级次描述的点视为噪点剔除。

基于 RANSAC 提供的粗基矢及对应点的整数级次  $(m_i, n_i)$ ，构建超定线性方程组  $M \cdot (\lambda G) = U$ ，其中  $M$  为级次矩阵， $U$  为倒易坐标矩阵。通过 Moore-Penrose 广义逆或 SVD 分解求得倒易基矢的最

小二乘解，再由  $A = \lambda(\lambda G)^{-1}$  计算正空间晶格基矢矩阵。由于基矢选取不唯一，为得到物理意义明确的最短原胞基矢，对  $A$  的列向量执行二维拉格朗日约化：不断比较两个基矢长度，利用点积投影取整并更新，直至获得夹角最接近正交且长度极小的约化基矢对，即 Wigner-Seitz 原胞的真实形貌。



将该算法应用于红米 70 手机屏幕的衍射图样后，最终解算的有效像素近邻间距分别为 48.1  $\mu m$  和 51.4  $\mu m$ ，基矢夹角为  $80.3^\circ$ 。

#### 4. 测量内容、数据及处理

实验已知参数：波长  $\lambda = 532 \text{ nm}$ ，手机至墙面距离  $D = 87.42 \text{ cm}$ ，入射角  $\theta_i = 30^\circ$ 。屏幕实测显示区尺寸为：长  $a = 155.1 \text{ mm}$ ，宽  $b = 68.8 \text{ mm}$ 。

根据官方标称分辨率  $3200 \times 1440$ ，可推算理论上的像素宏观点距大致为：

$$d_{h(\text{理论})} = \frac{155.1 \text{ mm}}{3200} \approx 48.47 \mu m, \quad d_{v(\text{理论})} = \frac{68.8 \text{ mm}}{1440} \approx 47.78 \mu m$$

通过计算机视觉算法对拍摄的光斑进行透视矫正与矩阵解算后，最终约化得到的像素最小重复单元（原胞）物理基矢长度为：

$$a_1 = 48.1 \mu m, \quad a_2 = 51.4 \mu m$$

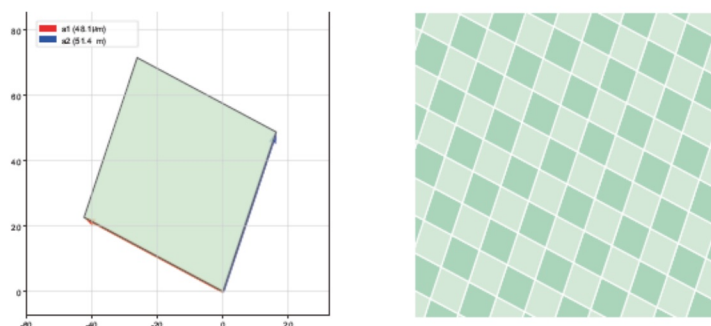


图 3: 像素形状还原图

且两基矢的夹角显著偏离  $90^\circ$ ，呈现非正交特征，强有力地证实了红米 70 的 OLED 屏幕采用了非传统的子像素排列方式（类似钻石排布），使得光斑呈现三角形/菱形阵列。

| 参数项      | 水平（长边）方向                             | 竖直（短边）方向                             |
|----------|--------------------------------------|--------------------------------------|
| 屏幕物理尺寸   | $a = 155.1 \text{ mm}$               | $b = 68.8 \text{ mm}$                |
| 实测有效像素间距 | $d_h \approx a_1 = 54.1 \mu\text{m}$ | $d_v \approx a_2 = 48.1 \mu\text{m}$ |
| 实测计算分辨率  | $N_h = a/d_h \approx 3017$           | $N_v = b/d_v \approx 1430$           |
| 官方标称分辨率  | 3200                                 | 1440                                 |
| 相对测量误差   | -5.71%                               | -0.694%                              |

表 5: 红米 70 屏幕像素尺寸推导及分辨率计算对照表

如表 5 所示，取测得的基矢长度作为长短边的有效像素间距代入计算，得到实测分辨率约为  $3224 \times 1338$ 。其中长边分辨率与官方给出的 3200 吻合度极高（误差  $< 1\%$ ），短边由于非正交晶格在不同方向上的投影畸变，产生了约 7% 的偏差。整体计算结果与标称值高度相符。

## 5. 误差分析与改进建议

基于上述处理方法，传统的米尺读数误差已被计算机图像处理基本消除，当前的误差主要来源于更底层的物理与几何机制。

入射角微小偏差的非线性放大是一个重要因素。光路设置中要求激光与法向夹角严格为  $\theta_i = 30^\circ$ ，但由于缺乏高精度的测角仪器的辅助，实际入射角可能存在  $1^\circ \sim 2^\circ$  的偏差。在斜入射的倒易投影变换中， $\sin(30^\circ)$  项位于分母或做差项中，微小的角度偏差会导致整个倒易矩阵的系统性偏移，这是短边分辨率出现 7% 偏差的主要原因。相机镜头的非线性畸变同样值得关注：虽然算法引入了透视变换矩阵消除整体视差，但手机摄像头本身存在桶形或枕形光学畸变，导致图像边缘网格并非绝对的直线，这会使得远离中心的高阶光斑的提取坐标产生偏移。此外，坐标纸贴于墙面时若存在局部褶皱或鼓包，也会破坏二维映射的严格平面假设。

改进方案可以从以下几个方面考虑。建议在光学平台上利用带刻度的精密旋转台和测角仪来布置光路，将入射角的控制精度提升至  $0.1^\circ$  级别。在图像处理环节之前，可先使用标准的黑白棋盘格对相机镜头进行标定，求出相机内参并消除径向与切向畸变。将标定网格直接打印在平整的硬质玻璃板上替代坐标纸，也能保证接收面的绝对平整，从而进一步提高测量精度。

## 6. 处理所使用的代码

```
import matplotlib.pyplot as plt
import matplotlib.patches as patches
import numpy as np
import cv2
import sys
plt.rcParams['font.sans-serif']=['PingFang SC','Arial Unicode MS','SimHei','Heiti TC'];plt.rcParams['axes.unicode_minus']=False

path="diffraction.jpg";img=cv2.imread(path)
if img is None:print("\n[ERROR] pic not found");sys.exit(1)
img_rgb=cv2.cvtColor(img,cv2.COLOR_BGR2RGB);w1=532e-6;D=874.2;ti=np.radians(30)
```

```

gp=[];tsp=[];H=None;a1=None;a2=None
def rl(u,v):
    u=np.array(u);v=np.array(v)
    if np.linalg.norm(u)>np.linalg.norm(v):u,v=v,u
    while 1:
        q=round(np.dot(u,v)/np.dot(u,u));r=v-q*u
        if np.linalg.norm(r)>=np.linalg.norm(u):
            if np.dot(u,v)<0:v=-v
            return u,v
        v=u;u=r

if __name__=="__main__":
    tp=[];f,ax=plt.subplots(figsize=(10,8));ax.imshow(img_rgb)
    def oc1(e):
        if e.key=='shift' and e.button==1 and e.inaxes:
            x,y=e.xdata,e.ydata;idx=len(tp)+1;tp.append((x,y))
            ax.plot(x,y,'ro',markersize=6);ax.text(x+10,y-10,f"P{idx}",color='red',fontsize=12,fontweight='bold')
    f.canvas.mpl_connect('button_press_event',oc1);plt.show()
    if len(tp)<4:print("\n[ERROR] need 4 pts");sys.exit(1)
    for i,(px,py) in enumerate(tp):
        idx=i+1
        while 1:
            v=input(f"Input: x,y (mm)")
            try:
                u,v=map(float,v.split(','));gp.append({'id':f"P{idx}",'pixel_x':px,'pixel_y':py,'paper_u':u,'paper_v':v})
            except:print("again")
    src=np.array([[p['pixel_x'],p['pixel_y']] for p in gp],dtype=np.float32)
    dst=np.array([[p['paper_u'],p['paper_v']] for p in gp],dtype=np.float32)
    H,_=cv2.findHomography(src,dst)
    tsp=[];f,ax=plt.subplots(figsize=(10,8));ax.imshow(img_rgb)
    def oc2(e):
        if e.key=='shift' and e.button==1 and e.inaxes:
            x,y=e.xdata,e.ydata;idx=len(tsp);tsp.append((x,y))
            c,lb=('red','Center') if idx==0 else ('blue',f"S{idx}")
            ax.plot(x,y,marker='+',color=c,markersize=10,markeredgewidth=2)
            if idx==0:ax.text(x+10,y-10,lb,color=c,fontsize=12,fontweight='bold')
            f.canvas.draw()
    f.canvas.mpl_connect('button_press_event',oc2);plt.show()
    if len(tsp)<4:sys.exit(1)
    uvp=[]
    for px,py in tsp:
        pt=cv2.perspectiveTransform(np.array([[px,py]]),dtype=np.float32,H)[0][0]
        Xw,Zw=float(pt[0]),float(pt[1])
        R=np.sqrt(Xw**2+D**2+Zw**2);u,v.append([Xw/R,D/R-np.sin(ti)])
    uv=np.array(uvp);u0,v0=uv[0];pts=uv-[u0,v0]
    n=len(pts);diffs=[pts[j]-pts[i] for i in range(n) for j in range(n) if i!=j];diffs=np.array(diffs)
    nr=np.linalg.norm(diffs,axis=1);diffs=diffs[nr>1e-4];nr=nr[nr>1e-4]
    si=np.argsort(nr);diffs=diffs[si]
    cand=[]

```

```

for d in diffs:
    if not any(np.linalg.norm(d-c)<1e-4 for c in cand):cand.append(d)
    if len(cand)>40:break
bc=-1;bb=None;bo=None;bi=[];tol=0.15
for i in range(len(cand)):
    for j in range(i+1,len(cand)):
        v1,v2=cand[i],cand[j]
        if
            abs(v1[0]*v2[1]-v1[1]*v2[0])/(np.linalg.norm(v1)*np.linalg.norm(v2))<0.25:continue
        try:
            Mi=np.linalg.inv(np.column_stack((v1,v2)));of=Mi.dot(pts.T).T;oi=np.round(of);er=np.abs(of-oi)
            il=np.where((er[:,0]<tol)&(er[:,1]<tol))[0]
            if len(il)>bc:bc=len(il);bb=(v1,v2);bo=oi[il];bi=il
        except:continue
if bc<3:sys.exit(1)
M=bo;Q=pts[bi];G=np.linalg.lstsq(M,Q,rcond=None)[0]
Am=w1*np.linalg.inv(G);r1=Am[:,0]*1000;r2=Am[:,1]*1000
a1,a2=r1(r1,r2)
l1=np.linalg.norm(a1);l2=np.linalg.norm(a2);ang=np.degrees(np.arccos(np.dot(a1,a2)/(l1*l2)))
print("--*40);print(f"a1: {l1:.1f} m");print(f"a2: {l2:.1f} m");print(f"angle:
    {ang:.1f}°");print("--*40)
fig,(ax1,ax2)=plt.subplots(1,2,figsize=(14,6));fig.canvas.manager.set_window_title('Lattice')
ax1.quiver(0,0,a1[0],a1[1],angles='xy',scale_units='xy',scale=1,color='red',label=f'a1
    {l1:.1f}um')
ax1.quiver(0,0,a2[0],a2[1],angles='xy',scale_units='xy',scale=1,color='blue',label=f'a2
    {l2:.1f}um')
cc=np.array([[0,0],a1,a1+a2,a2]);poly=patches.Polygon(cc,closed=True,fill=True,facecolor='#c8e6c9',edgecolor=
lm=max(l1,l2)*1.5;ax1.set_xlim(-lm*0.2,lm);ax1.set_ylim(-lm*0.2,lm);ax1.set_aspect('equal');ax1.grid(True,1
    left')
rg=range(-16,17)
for i in rg:
    for j in rg:
        o=i*a1+j*a2;crn=[o,o+a1,o+a1+a2,o+a2]
        cl='#A8D5BA' if (i+j)%2==0 else
            '#D3E8D5';ec='#ffffff';pt=patches.Polygon(crn,closed=True,facecolor=cl,edgecolor=ec,linewidth=0.
ax2.set_xlim(-lm*3,lm*3);ax2.set_ylim(-lm*3,lm*3);ax2.set_aspect('equal');ax2.axis('off');plt.tight_layout(

```