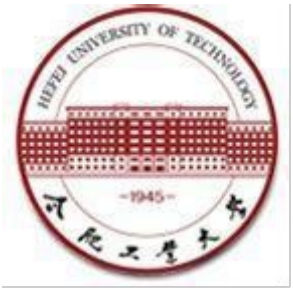
学　　号：2014213282　　密　级：

合肥工業大學

Hefei University of Technology

本科毕业设计（论文）

UNDERGRADUATE THESIS



类　　型：论文

题　　目：数字 Moiré 3D 测量及精度分析

专业名称：应用物理学

入校年份：2014 级

学生姓名：张凡

指导教师：袁自钧　副教授

学院名称：电子科学与应用物理学院

完成时间：2019 年 05 月

合　肥　工　业　大　学

本科毕业设计（论文）

数字莫尔三维测量及其精度分析

学生姓名：张凡

学生学号：201421382

指导教师：袁自钧　副教授

专业名称：应用物理学

学院名称：电子科学与应用物理学院

2019 年　5 月

A Dissertation Submitted for the Degree of Bachelor

Digital Moiré Profilometry and Its Accuracy Analysis

By

Zhang Fan

Hefei University of Technology

Hefei, Anhui, P.R.China

May, 2019

（论文）独创性声明

本人郑重声明：所呈交的毕业设计（论文）是本人在指导教师指导下进行独立研究工作所取得的成果。据我所知，除了文中特别加以标注和致谢的内容外，设计（论文）中不包含其他人已经发表或撰写过的研究成果，也不包含为获得 合肥工业大学 或其他教育机构的学位或证书而使用过的材料。对本论文成果做出贡献的个人和集体，本人已在设计（论文）中作了明确的说明，并表示谢意。

毕业设计（论文）中表达的观点纯属作者本人观点，与合肥工业大学无关。

毕业设计（论文）作者签名：　　签名日期：　　年　　月　　日

毕业设计（论文）版权使用授权书

本学位论文作者完全了解 合肥工业大学 有关保留、使用毕业设计（论文）i 的规定，即：除保密期内的涉密设计（论文）外，学校有权保存并向国家有关部门或机构送交设计（论文）的复印件和电子光盘，允许设计（论文）被查阅或借阅。本人授权 合肥工业大学 可以将本毕业设计（论文）的全部或部分内容编入有关数据库，允许采用影印、缩印或扫描等复制手段保存、汇编毕业设计（论文）。

（保密的毕业设计（论文）在解密后适用本授权书）

学位论文作者签名：　　指导教师签名：

签名日期：　　年　　月　　日　　签名日期：　　年　　月　　日

摘　要

数字莫尔三维测量，是在传统莫尔三维测量的理论基础上，利用计算机生成和处理莫尔条纹，并建立被测物体的三维模型。该测量过程最少仅需捕捉一张图像，能满足多样测量任务。数字莫尔测量方法的这一优点，使得该方法各项处理步骤成为学者们不断优化和改进的对象。本论文探究了数字莫尔的测量原理，推导了莫尔条纹相位分布和待测物体高度分布的映射。利用该理论基础，本论文使用 3ds Max 三维建模仿真软件，和 MATLAB 科学计算包，探索并验证了数字莫尔测量过程中关键步骤和对应的处理算法。本论文实现了在计算机后期生成莫尔条纹，滤除高频噪声得到折叠的相位图。最后，本论文讨论和比较了相位提取和解包裹的方法，并结合实际应用场景分析了相位分布图计算的途径。

关键词：三维测量；莫尔条纹；数字相移；平稳小波变换；图像处理

ABSTRACT

Digital Moiré profilometry uses the same measuring theory as the traditional Moiré Profilometry. Unlike the established method, however, it generates and analyses the Moiré patterns afterwards by exploiting computer algorithms. To re-build the 3D model for the object, this method captures only one image and can be applied to multiple measuring tasks. This benefit has driven researchers to improve and optimize the processing performance of digital Moiré. In this research, the fundamentals of ditial Moiré profilometry is explored by affirming that the phase map of the Moiré patterns is in direct propotion with the height map of the object. To verify the theory of digital Moiré profilometry, its key procedures and algorithms are implemented with a 3D modeling software, 3ds Max, and a scientific computing library, MATLAB. Specifically, the algorithms to genertate Moiré patterns and filter the high-frequency grids in it are investigated. Finally, this paper compares different methods to extract and unwrap the phase map. To illustrate the two steps, several applications are analyzed.

KEYWORDS: Profilometry; Moiré pattern; digital phase-shift; stationary wavelet transform; image processing

目　录

插图清单

表格清单

1 绪论

为了把握数字莫尔三维测量技术的研究背景，本章简要介绍了主流三维测量方法的实现原理和适用场景。在对比了两种方法的优缺点后，本章第二部分介绍了数字莫尔三维测量的具体步骤和专有名词。

1.1 数字莫尔三维测量介绍

三维测量，又称三维面形测量。根据其采用物理性质，实现方式不同，分为多种测量方案。从物理机制上分类，三维测量可分为光学三维面形测量，电磁学三维面形测量，超声波三维面形测量，和机械三维测量。前三种方案借助被测物体在光学，电磁或超声学上的物理特性，无需接触被测物体，就可得出物体三维形貌，因此对被测物体损伤较小。但相比之下，机械三维测量一般采用的是接触式测量方法。因此，机械三维测量，适合测量机加工件等不易变形的被测物体。该方法在工业生产上已有广泛应用。

表 1 三坐标测量机和光学三维测量对比

	三坐标测量机	光学三维测量
安装校准	一次性安装校准，但费用贵	无需安装，移动设备需校准
适用表面	坚硬	漫反射
量程	固定，由设备尺寸决定	较大，有投影图样和设备分辨率决定
算法难度	前期路径规划，NP 问题	后期图像处理，可借助 GPU 并行处理
成本	非常昂贵，维护成本高	较昂贵，但设备普及
便携	由于测量精度要求，无法随意移动	部分方案可随意移动
精度	高	还原三维模型，精度差
时间	前期路径规划时间长，测量时间长	测量时间短，处理时间较长

数字莫尔三维测量是光学三维测量中较为经济实用的方案。它采用的设备是投影仪和相机。但与其他光学三维测量的原理不同，数字莫尔三维测量并不直接处理经调图像，而是利用叠加条纹的方式，形成对应物体等高线的莫尔条纹，而后通过处理莫尔条纹得出物体高度分布。因此，数字莫尔三维测量具有非接触，

量程大，设备易携带等优点，并因此得到了较为广泛的应用[1-5]。例如，在文物保护方面，早在 2005 年，意大利学者就开始使用莫尔三维测量技术数字化意大利著名古建筑蒙特城堡[1]的泥砖外墙[6]。

数字莫尔三维测量，将传统莫尔三维测量中，产生莫尔图样的过程和由该莫尔图样得出所测表面三维模型的过程，转移到计算机后端处理[8]。在传统莫尔三

图 1 意大利学者利用莫尔三维测量技术部分数字化的古文物遗址蒙特城堡

维测量方法中，需将投影光栅和参考光栅叠加，在被测物体上形成与等高线对应的莫尔图样[2]。而在数字莫尔测量方法中，投影光栅被投影仪的数字光处理芯片代替，莫尔图样被直接投影到被测物体上，然后数码相机的 CCD 传感器捕捉图片，最后电脑程序将和投影条纹同频率的图样重合，形成莫尔条纹。使用后端处理算法，将投影条纹同周期条纹图样和被物体高度信息扭曲的条纹图样叠加产生莫尔条纹的过程，称为数字莫尔条纹生成[9]。经过该过程得到的莫尔图样，不仅包含有所需的等高线，还含有调制物体高度信息的高频条纹。为了得到单一高度信息，为下一步计算相位图分布做准备，所得莫尔图样要进一步滤波。去除由数字莫尔条纹生成得到图像中的高频噪声的方法叫做条纹去除[10]。经过这两步计算得出的单张莫尔图样，需要结合其他莫尔图样，利用三角函数关系，得出和被测面高度对应的相位分布。该过程成为相位提取。在相位提取过程中的多张莫尔图样，必须是投影条纹图样不同初始相位但同一条纹频率的条纹图样，经过数字

注1 图 3-b 是蒙特城堡的图片。这张图片属于公共领域，可自由使用，来源于意大利国家公园服务网站。

莫尔条纹生成得到。但由相位提取得出相位图像，并非和被测面高度直接对应，因此被称为折叠相位。由于所用三角函数关系具有 2π 整数的不确定性，提取的相位需要使用不同条纹频率得到的折叠相位作为参考，最终得出连续变化的，对应物体高度的展开相位。使用不同调制频率的条纹得到的折叠相位为参考相位，补充折叠相位和高度对应关系 2π 整数差异的过程，称为相位展开。至此，一个符合物体相对几何特征的相位分布已经得到。在实际测量中，为了得到准确符合被测物体相对几何特征的三维模型，数字莫尔三维测量法要求，在测量前通过系统校准得出莫尔波长和测量高度的关系。最后利用该关系，物体的绝对三维模型可以通过展开相位分布计算得到。使用平板，在不同高度位置，测量莫尔波长，最终得出莫尔波长和高度对应关系的过程称为为系统校准。利用相-高关系得出被测物体绝对三维模型的过程称为高度转换。

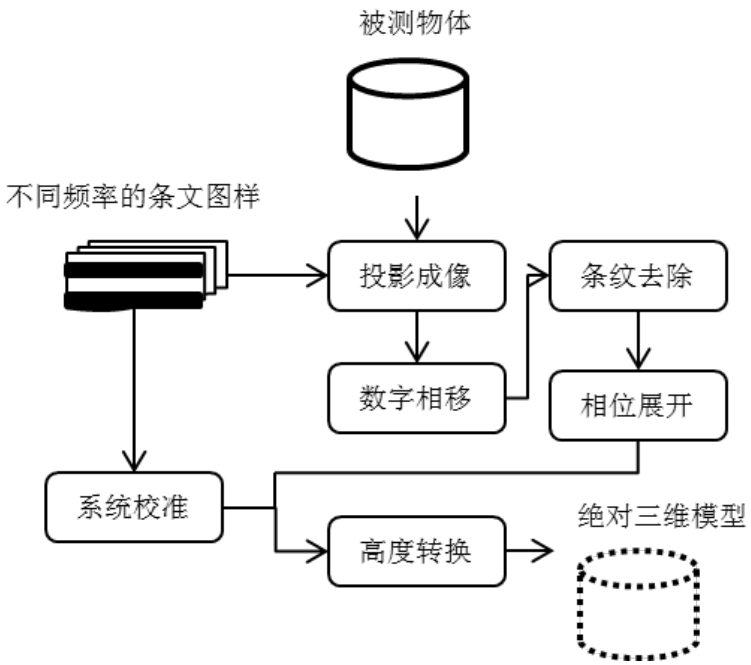


图 2 数字莫尔三维测量流程图

1.2 本文主要内容

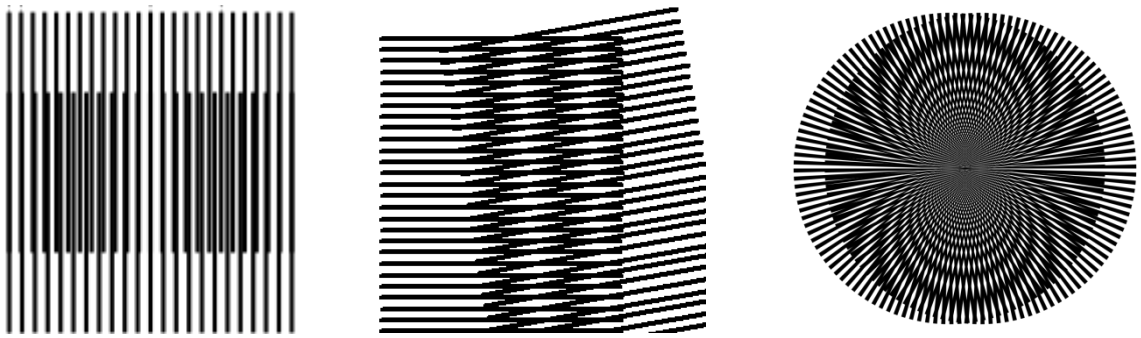
基于滑铁卢大学学者的工作[8]，本论文分析了数字莫尔三维测量过程中，近似处理，数字莫尔条纹生成过程中高频噪声，滤波改变图像细节等因素对数字莫尔三维测量误差产生的影响。结合数字莫尔三维测量理论基础，本论文研究中使用 3ds Max 三维建模仿真软件和 MATLAB 科学计算包，探索并验证了数字莫尔测量

过程中关键步骤的处理算法，为后期实际搭建数字莫尔 3D 测量系统做了可行性和难点分析。在第 2 章中，本论文将利用几何关系和傅立叶级数两种视角讨论莫尔条纹产生的原理，并将借助三角测量法的几何模型推导相位差和高度的对应关系。在第 3 章中，本论文将拓展第 2 章的相位差-高度关系，利用立体几何关系推导相位分布和高度分布的非线性关系和线性关系。在第 4 章，本论文将讨论叠加由物体扭曲的投影条纹和同周期不同初始相位条纹图样后所得强度分布的表达式，并将演示数字莫尔条纹的产生和平稳小波变换结合傅立叶频域的滤波方法。最后，在第 5 章中，本论文将修正第 4 章中的莫尔条纹灰度值与相位分布关系，并将借助外差法计算出折叠相位分布和介绍了相位展开法的原理。

2 相位-高度对应关系

传统莫尔三维测量和数字莫尔三维测量有三点不同。第一个区别是光栅的使用。传统莫尔三维测量，无论是投影莫尔法，还是阴影莫尔法，都需要使用至少一个光栅[11, 12, 1]。数字莫尔三维测量不再使用光栅，而是直接使用投影仪对物体投影条纹图样[8]。第二个区别是莫尔条纹产生的原理。传统莫尔三维测量法必须通过光栅和光栅的投影，或者光栅和另一个相同参数的光栅在被测物体上重叠，产生莫尔条纹。而数字莫尔三维测量则是在测量后，在电脑后端处理使用条纹和捕捉图像矩阵相乘，实现莫尔效应；第三个区别是后期处理。早期的莫尔测量由于信息技术的发展有限，只能得出莫尔条纹，并以此作为等高线，人工定标，操作复杂[11]。而随着电脑处理器速度的提高，数字莫尔三维测量使用后端处理算法，不仅能得到被测物体的数字三维模型，同时还可使用拟合等多种手段提高精度[13, 10, 3, 4]。

2.1 莫尔条纹产生的原理



(a) (b) (c)

图 3 莫尔条纹产生方式：(a)不同周期；(b)成角度；(c)不同周期和角度

莫尔条纹[2]形成的原因包括不同周期，不同重复周期函数，光栅有夹角，不同光栅材料折射率等[14]。讨论等间隔不同周期光栅叠的情况。如图 5，等间隔光

注[2] 图 4-a，b，c 来自于维基百科莫尔条纹词条，可直接使用。来源网址：图 4 来自维基百科莫尔条纹词条，可直接使用。来源网址：https://en.wikipedia.org/wiki/Moir%C3%A9_patternhttps://en.wikipedia.org/wiki/Moir%C3%A9_pattern

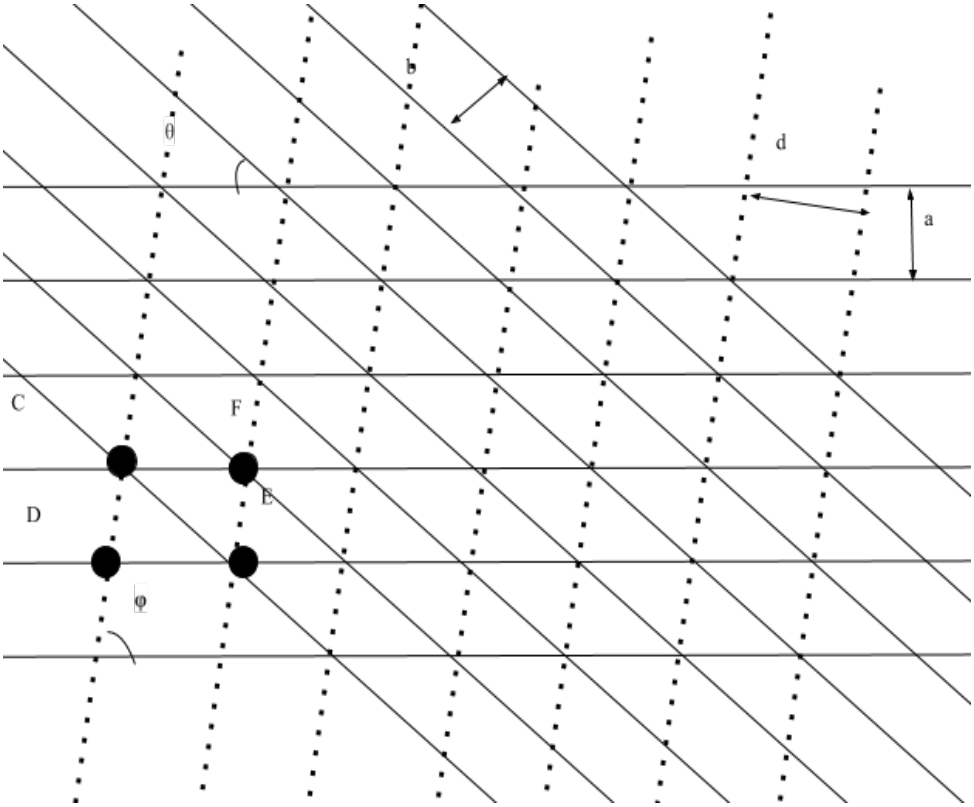


图 4 等间隔光栅以夹角θ重叠产生莫尔条纹

栅 *A*，间隔为 *a*；等间隔光栅 *B*，间隔为 *b*。当两光栅以θ角度重合时，出现明暗变化的图样就是莫尔条纹。上图中，黑色虚线为莫尔条纹，水平直线组代表光栅 *A*，倾斜直线组代表光栅 *B*。观察几何关系，可得，

$$S_{CDEF} = DE \times a = CD \times d = CE \times b \tag{2.1}$$

$$CD^2 = CE^2 + DE^2 - 2CE \times DE\cos\theta \tag{2.2}$$

将 *DE*，*CD*，*CE* 均用S_{CDEF}和*a*，*b*，*c* 等表示，约分化简可得，

$$D = \frac{ab}{\sqrt{a^2 + b^2 - 2ab\cos\theta}} \tag{2.3}$$

又可由，

$$\sin\varphi = \frac{d}{DE} \tag{2.4}$$

$$DE = \frac{b}{\sin\varphi} \tag{2.5}$$

可得，

$$\mathrm{Sin}\varphi=\frac{a\sin\theta}{\sqrt{a^2+b^2-2ab\cos\theta}} \tag{2.6}$$

可由重叠光栅的光栅间隔和两光栅交角计算出莫尔条纹的周期和相对于光栅的夹角。当a = b时，即使用相同间隔的光栅重叠时,可得，

$$d=\frac{a}{2\sin\left(\frac{\theta}{2}\right)} \tag{2.7}$$

$$sin\varphi = cos\frac{\theta}{2} \tag{2.8}$$

此时，莫尔条纹和两光栅夹角的平分线垂直。由于数字莫尔三维测量使用的相同间隔周期的条纹重叠形成莫尔条纹，可以将物体所扭曲的条纹处理解为两光栅出现一定夹角，而该夹角随着被测物体位置高度而变化。

使用简化几何关系分析莫尔条纹的产生和其参数，是解释莫尔现象的多种数学模型之一[15]。除此之外，也有学者通过傅立叶变换和光栅透过率函数来研究不同频率条纹间隔，即不同周期条纹叠加产生的莫尔条纹[16]。推导过程如下。

傅立叶级数$f_1(x,y)$和$f_2(x,\ y)$描述两不同频率光栅的透过率函数。描述一个只在一定范围内周期重复的光栅，需要无穷个不同系数的余弦函数分量求和。分量的系数决定了最后光栅的几何特征,而余弦函数的相位是关于位置坐标的函数，决定了具体以某一个光栅周期为中心展开。傅立叶级数表达光栅 1 和 2 的传递函数如下。

$$f_1(x,y)=a_1+\sum_{n=1}^{\infty}b_{1n}\cos[n\phi_1(x,y)] \tag{2.9}$$

$$f_2(x,y)=a_2+\sum_{m=1}^{\infty}b_{1m}\cos[m\phi_2(x,y)] \tag{2.10}$$

在上两式中，$n\phi(x,y)$对应表二中正弦函数中$\frac{n\pi x}{L}$。根据光栅间隔不同，展开周期具体级数不同，$n\phi(x,y)$取不同的值。而系数b_{ij}则对应着傅立叶系数，决定着最后求和所得到的光栅几何特征。上述二式可表述任意两不同光栅。

表 2 常见周期函数和其单周期傅立叶展开级数

函数名称	单个周期函数表达式	单周期傅立叶级数展开
方波	$2\left[H\left(\frac{x}{L}\right)-H\left(\frac{x}{L}-1\right)\right]-1$	$\frac{4}{\pi}\sum_{n=1,3,5\ldots}^{\infty}\frac{1}{n}\sin\left(\frac{n\pi x}{L}\right)$
锯齿波	$\frac{x}{2L}$	$\frac{1}{2}-\frac{1}{\pi}\sum_{n=1}^{\infty}\frac{1}{n}\sin\left(\frac{n\pi x}{L}\right)$
三角波	$T(x)$	$\frac{8}{\pi^2}\sum_{n=1,3,5\ldots}^{\infty}\frac{(-1)^{\frac{n-1}{2}}}{n^2}\sin\left(\frac{n\pi x}{L}\right)$

当这两个光栅重叠时，总体透过率函数为二者透过率函数之积,可表达如下，

$$
\begin{aligned}
&f_1(x,y)f_2(x,y)\\
&=\left(a_1+\sum_{n=1}^{\infty}b_{1n}\cos[n\phi_1(x,y)]\right)\left(a_2+\sum_{m=1}^{\infty}b_{1m}\cos[m\phi_2(x,y)]\right)\\
&=a_1a_2+a_1\sum_{m=1}^{\infty}b_{1m}\cos[m\phi_2(x,y)]+a_2\sum_{n=1}^{\infty}b_{1n}\cos[n\phi_1(x,y)]\\
&\qquad+\sum_{n=1}^{\infty}b_{1n}\cos[n\phi_1(x,y)]\sum_{m=1}^{\infty}b_{1m}\cos[m\phi_2(x,y)]\\
&=a_1a_2+a_1\sum_{m=1}^{\infty}b_{1m}\cos[m\phi_2(x,y)]+a_2\sum_{n=1}^{\infty}b_{1n}\cos[n\phi_1(x,y)]+\\
&\qquad\sum_{n=1}^{\infty}\sum_{m=1}^{\infty}b_{1n}b_{1m}\cos[n\phi_1(x,y)]\cos[m\phi_2(x,y)] \qquad (2.11)
\end{aligned}
$$

该式前三项为条纹本身的强度，对应着数字莫尔三维测量中需要去除的高频噪声。而第四项可以使用积化和差，计算出差频和和频两项。这两项为莫尔条纹所携带的信息，来源于物体高度扭曲了投影条纹。物体的高度信息使得原本等间距，零夹角的两幅条纹出现了夹角变化，周期变化。从莫尔条纹反推恢复物体高度是下一小节三角测量法，以及本论文的重点。

2.2 三角测量法

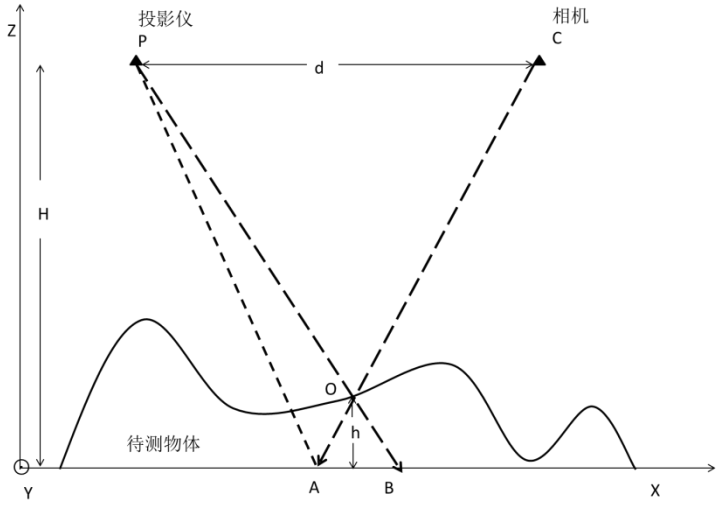


图 5 三角测量法几何关系

在数字莫尔三维测量中，莫尔条纹是由经待测表面各点不同高度扭曲后的条纹图样和相同周期条纹图样叠加而产生。所以由此生成的莫尔条纹含有物体的高度信息。利用三角测量法的数学模型可建立待测面高度和投影仪[12, 2, 4]，相机和被测点连线延长线与参考平面两交点之间的条纹相位差之间的联系。

如上图所示，在相机C和投影仪P(以下称$\mathrm{C}-\mathrm{P}$ 平面)和参考平面（坐标系 XY 平面）平行，以及投影仪投影光线和参考平面垂直的两个前提条件下，可根据几何条件计算出待测点O的高度h和A，B两点相位差之间的关系[1, 2, 4]。根据，

$$\Delta \mathrm{ABO} \sim \Delta \mathrm{CPO} \tag{2.12}$$

可得，

$$\frac{h}{H-h}=\frac{|AB|}{d} \tag{2.13}$$

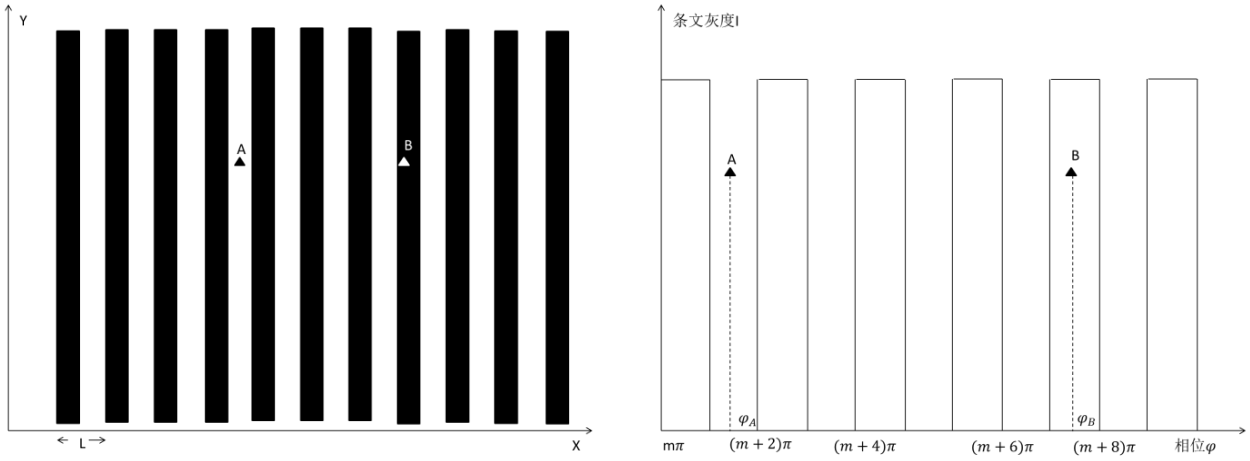
又由A，B两点相位和投影后周期长度 L 的关系，可得，

$$|AB|=\frac{\varphi_B-\varphi_A}{2\pi}L=\frac{\varphi_{BA}L}{2\pi} \tag{2.14}$$

带入(2.12)式，可得出待测点高度和相位之间的对应关系，可得，

$$\mathrm{H}=\frac{H}{1+\frac{2\pi d}{L\varphi_{BA}}} \tag{2.15}$$

其中相机和投影仪之间距离为d，$\mathrm{C}-\mathrm{P}$ 平面到参考平面之间的距离为 H，投影条纹在参考平面上的周期L以及A，B两点之间的相位差，在实际测量过程中，



(a) (b)

图 6 利用 AB 两点相位差得出被测点高度：(a) 参考平面单独投影条纹 AB 相对位置；（b)条纹灰度和 X 轴各点相位关系

并不是直接测量以上几何参数，并带入(2.15)式。实际测量过程的高度和相位关系在下一章节系统校准中会详细讨论。

3 实物系统校准

前一章定性分析了莫尔条纹和物体高度信息的对应关系，之后通过三角测量法的几何模型得出单一待测点高度和参考平面对应点相位差之间的关系。该关系式含有实际测量过程中，难以精确测量也不便测量的几何参数。为了这一解决问题，本章将拓展上一章得到的相位-高度关系，并根据文献，运用数学工具，提出实际测量系统校准的具体过程。

3.1 非线性校准原理

2007 年加拿大学者根据(2.14)式子中单一被测点的高度-相位差关系[17]，提出了非线性和线性校准两种方法，以下是非线性校准的具体推导过程[17]。在(2.14)式中，

$$\mathrm{h} = \frac{H}{1 + \frac{2\pi d}{L\varphi_{BA}}}$$

$\varphi_{BA} = \varphi_B - \varphi_A$，如图-7，根据费马定理，测量过程位于O点的投影条纹，应与无被测物体时位于参考平面上 B 点的投影条纹是同一级次。因此，B，O两点的

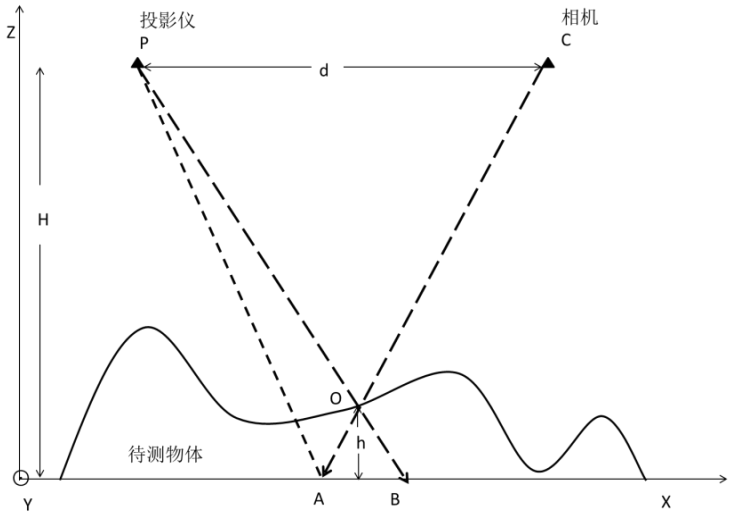


图 7 两点 O，相位等同

相位相等，即$\varphi_B = \varphi_O$。则，$\varphi_{BA} = \varphi_{OA}$，将此关系带入(2.14)中，可将该式关系变换为被测点高度与被测点和相机-被测点延长线与参考平面交点相位之差的关系。

$$\mathrm{h}=\frac{H}{1+\frac{2\pi d}{L\varphi_{OA}}} \tag{3.1}$$

考虑被测点位于$\mathrm{X}-\mathrm{Z}$平面之外的情况，即被测点位于纸面外的情况。假设被测点$\mathrm{X}-\mathrm{Y}$坐标为(x_0,y_0)。如下图，

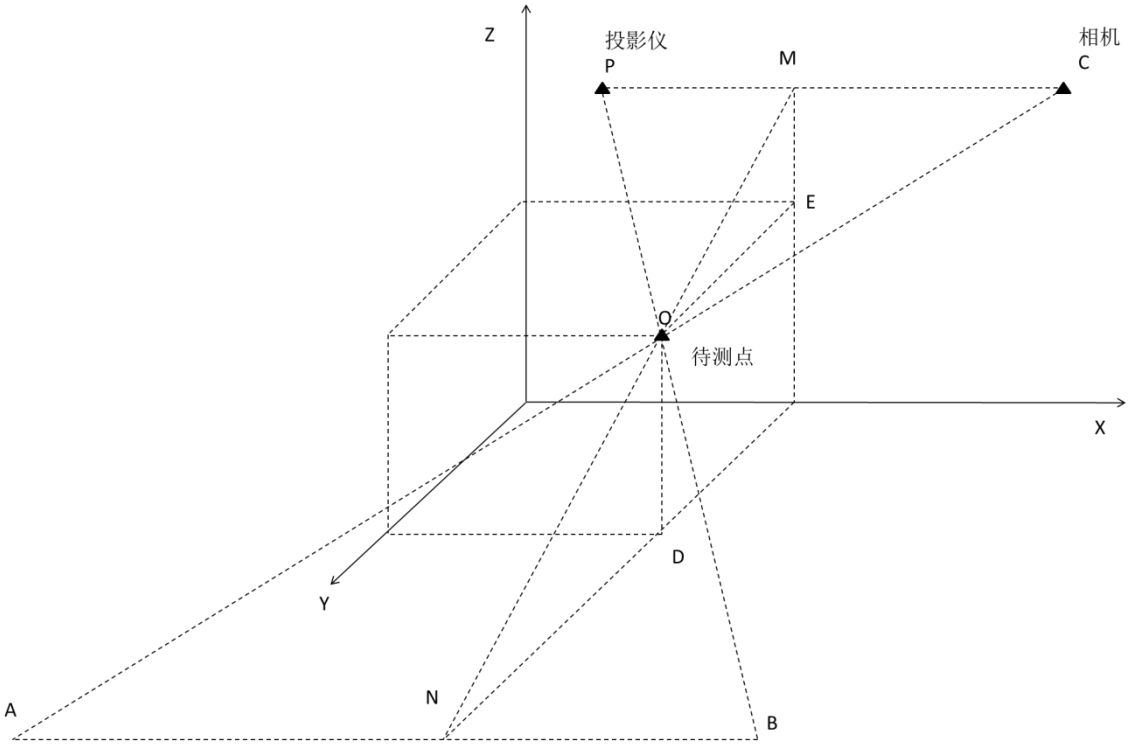


图 8 被测点在 X-Z 平面之外

在上图中同样存在$\Delta\mathrm{ABO}\sim\Delta\mathrm{CPO}$。则，

$$\frac{|CP|}{|AB|}=\frac{d}{\mathrm{L}\Delta\varphi(x_0,y_0)/2\pi}=\frac{|MO|}{|NO|} \tag{3.2}$$

又$\measuredangle\mathrm{ODN}\sim\Delta\mathrm{MEO}$，

$$\frac{|MO|}{|NO|}=\frac{|ME|}{|OD|}=\frac{H-h(x_0,y_0)}{h(x_0,y_0)} \tag{3.3}$$

由上两式可得，

$$\mathrm{H}(x_0,y_0)=\frac{H}{1+\frac{2\pi d}{L\Delta\varphi(x_0,y_0)}} \tag{3.4}$$

由于上式对任何被测点均成立，考虑整个待测平面，可有，

$$\mathrm{H}(\mathrm{x},\mathrm{y}) = \frac{H}{1 + \frac{2\pi d}{L\Delta\varphi(\mathrm{x},\mathrm{y})}} \tag{3.5}$$

将上式整理得，

$$\Delta\varphi(\mathrm{x},\mathrm{y}) = \frac{\frac{2\pi d}{L\mathrm{H}}\mathrm{h}(\mathrm{x},\mathrm{y})}{1 - \frac{1}{H}\mathrm{h}(\mathrm{x},\mathrm{y})} \tag{3.6}$$

该式描述了相位分布和和高度分布的非线性关系。在非线性校准中，需要确定的参数如下。

$$\mathrm{M} = \frac{2\pi d}{L\mathrm{H}} \tag{3.7}$$

$$\mathrm{N} = \frac{1}{H} \tag{3.8}$$

m和n均和测量装置的几何参数有关，可通过最小二乘方法处理已知$\mathrm{h}(\mathrm{x},\mathrm{y})$和$\Delta\varphi(\mathrm{x},\mathrm{y})$的数据，得出最大似然系数m和n的具体数值。然后，将拟合所得的参数带入相位-高度非线性关系(3.5)中，从而用于后续测量过程中。最后该式(3.5)可被用于将相位分布转换为待测表面高度分布。

3.2 线性校准原理

在 2007 年，加拿大学者提出了线性校准[17]。线性校准在莫尔三维测量早期已有应用[11]。线性校准和非线性校准同样使用最小二乘法估计参数，但线性校准了利用了近似关系，将相位-高度关系转变为线性方程。具体推导过程如下。

在 3.1 节中，高度相位关系为，

$$\mathrm{H}(\mathrm{x},\mathrm{y}) = \frac{H}{1 + \frac{2\pi d}{L\Delta\varphi(\mathrm{x},\mathrm{y})}} \tag{3.9}$$

当 H≫h 时，可将分母中的常数项忽略。整理可得，

$$\mathrm{H}(\mathrm{x},\mathrm{y}) = \frac{LH}{2\pi d}\Delta\varphi \tag{3.10}$$

即，

$$\mathrm{H}(\mathrm{x},\mathrm{y}) = \mathrm{K}\Delta\varphi \tag{3.11}$$

系数 K 同样与测量系统的几何参数有关，可使用已知位置分布和相位分布的数据估计。

3.3 莫尔波长与相位-高度转换

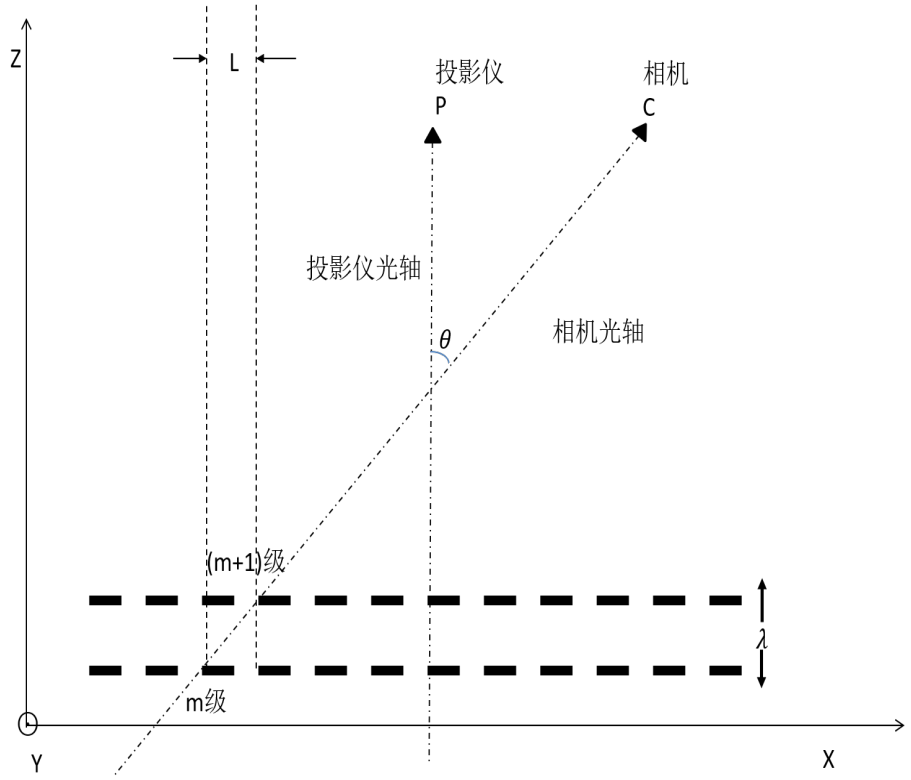


图 10 莫尔波长的计算

相位-高度线性关系(3.9)大大简化了系统校准的数学解释和处理过程。为了进一步理解系统校准的原理，莫尔波长作为过渡变量被引入[8, 18]。当投影平面垂直于投影方向靠近或远离投影仪时，条纹会向上或向下移动。虽然该过程会伴随投影条纹的放大和缩小，但当条纹移动一个周期时，由于投影仪投影张角较大且条纹周期尺度在毫米级别，可近似认为莫尔条纹大小不变。此时，投影平面移动的距离则为莫尔波长。

由图 10 显示的几何关系可知，在已知条纹周期和相机-投影仪光轴夹角 θ，可得的莫尔波长如下，

$$\Lambda = \frac{L}{tan\theta} \tag{3.12}$$

该公式表明物体高度每变化 λ，则投影条纹变化一个周期，此时莫尔条纹也变化一个周期，其相位变化 2π。则在相位-高度关系式(3.9)中，

$$K = \frac{\lambda}{2\pi} \tag{3.13}$$

也就是说，一个莫尔波长的高度变化，对应相位变化 2π。但该系数并不是常数。在被测高度变化处于周期尺度范围内，投影条纹可看作恒定大小，K 可看作常数。当高度变化范围更大时，投影仪对投射出的条纹图样具有放大或缩小作用，会使条纹周期 L 发生变化。随着高度增加，条纹图样缩小，条纹周期 L 减

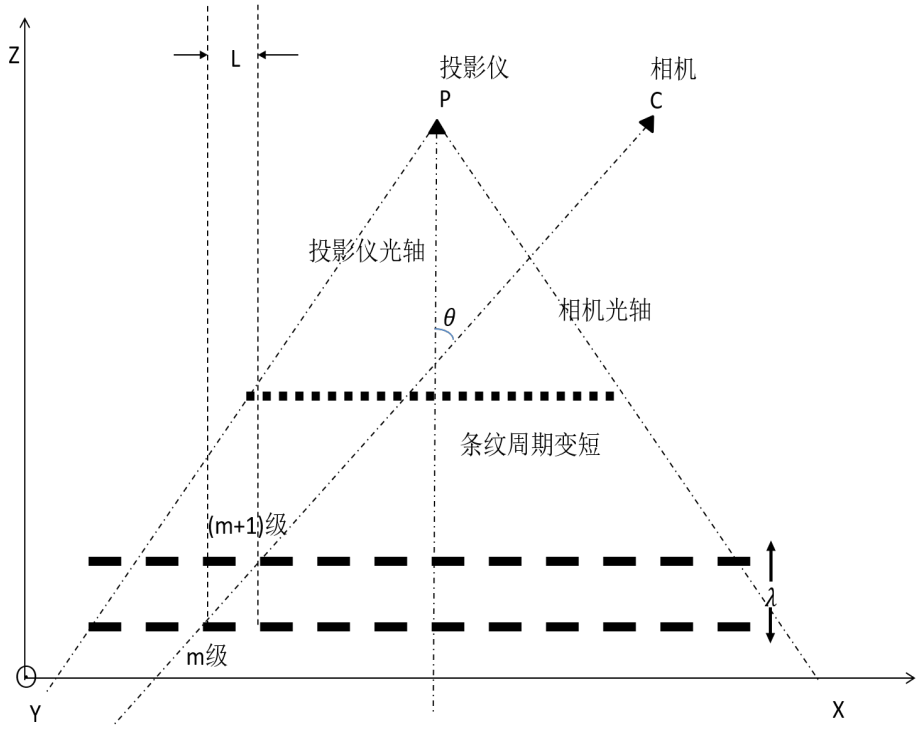


图 9 高度变化范围过大，条纹周期 L 减小

小，莫尔波长减小。反之，条纹周期 L 变大，莫尔波长变大。因此，在线性系统校准时，需要校准不同周期条纹图样的线性系数。

3.4 线性校准过程

为了建立被测面高度分布和相位分布的精准对应关系，需要得出莫尔波长和高度的对应关系。该关系需被用来计算适合物体高度分布的莫尔波长，以便完成相位-高度转换[12, 2, 4]。因此线性校准需要在一系列已知高度的位置，放置平板并投影捕捉条纹图像。经过数据采集后，莫尔波长和高度的线性变化关系可以根据相位信息和已知高度信息拟合得出。后期相位展开需要两个或三个不同莫尔波长的折叠相位图。对于同一测量物体有限的高度变化范围内，需通过投影产生不同周期的条纹图样，从而产生不同莫尔波长。因此，线性系统校准过程需要针对条纹周期和已知高度位置两个维度，完成相位-高度线性关系的拟合。以下是根据

文献[8]提出的线性校准步骤和具体测量要求，以及数字莫尔三维测量方法的系统校准过程。

(1) 为了同时满足尺寸较大和较小的待测物体的测量要求，测量范围需为$[0\ mm, 250\ mm]$。同时，采用的投影条纹的相对周期[3]分别为 6，8，10，12，18 像素。

(2) 在0 mm 到 250 mm范围内，以 0.5mm 为间隔，取 500 个位置，并分别在这 500 个位置，捕捉针对 5 个不同周期的条纹图样投影到平板上的图像，共 2500 张图像。

(3) 利用数字莫尔条纹生成和条纹去除，得出每一位置，每一周期的投影条纹的相位分布图。

(4) 针对同一周期的投影条纹得到的相位图的某一像素点，标出像素点灰度与高度变化的数据点，并拟合得出该像素点灰度和高度变化的关系。

(5) 针对（4）中得到的类正弦关系，将连续两个极大值高度之差作为远离投影仪-相机平面的极大值处高度位置的莫尔波长。

(6) 将（5），针对（4）中像素灰度和高度变化的类正弦关系的每组连续极大值，得到该像素莫尔波长和其对应高度的数据，即，

$$(\lambda_{i,j,k}, h_{i,j,k})$$

其中$\mathrm{i} = 1,2,3 \ldots\ldots \mathrm{m}$,为灰度极小处的级次，$h_{i,j,k}$为第i 级次极大值处，位于图像$(\mathrm{j}, \mathrm{k})$处对应的高度，可知，

$\lambda_{i,j,k} = h_{i,j,k} - h_{i-1,j,k}$则为第i 极大值处，位于图像$(\mathrm{j}, \mathrm{k})$处像素的莫尔波长。

(7) 将(4)(5)(6)针对（4）中同一周期条纹得到的相位图的每一点像素重复，并针对不同位置像素点，将同一高度得到的莫尔波长针对不同位置像素点平均，得到平均高度，该周期条纹所对应的莫尔波长。对于分辨率

注[3] 相对周期为条纹重复一个周期的像素个数，与投影到参考平面后重复一周期沿 X 轴经过的距离不同。

$\mathrm{M}\times\mathrm{N}$的相位图，

$$\lambda(h_i)=\frac{1}{MN}\sum_{j}\sum_{k}\lambda_{i,j,k}$$

(8) 将(7)中得到的高度-莫尔波长数据，拟合出线性关系，留作最后相位展开使用。

(9) 针对不同周期的条纹，重复（7）（8），并所得关系留作最后相位展开使用。

在系统校准过程中，有两个前提假设。第一个假设是可将在已知高度位置的拍摄的图像成功转换为相位分布。第二个假设是在物体的高度变化范围内可使用同一莫尔波长。前一个假设，是第 4 章重点。第二个假设，是第 5 章相位展开方法和高度转换的近似处理要求。

假设位置取样个数S，条纹周期个数P的情况下，系统校准方法的时间复杂度为$\mathrm{O}(S^2\mathrm{PMN})$。从此可看出，精准系统校准过程需要大量的时间投入，这一要求也限制了由分离投影仪器和相机组成的数字莫尔三维测量系统的便携性。但在商用设备中，由于系统的几何参数固定，该校准过程一般在出厂完成，对用户体验无严重影响。由于系统校准需要使用实物测量系统，本论文不着重讨论数据收集和处理过程。

4 生成数字莫尔图样

通过第 2 章和第 3 章的讨论，得出了相位分布和高度分布的线性关系。为了三维重建被测面，关键在于生成莫尔图样并计算出相位分布。而生成数字莫尔图样需要叠加和滤波两个过程。本章将讨论数字莫尔三维测量中叠加和滤除高频条纹。

4.1 叠加生成数字莫尔条纹

将二值条纹图样投影到待测物体表面后，捕捉到图像的强度分布函数为，

$$\mathrm{I}_o(\mathrm{x},\mathrm{y}) = b_o + \mathrm{rect}\big(\phi(x,y)\big) \tag{4.1}$$

其中，

$$\phi(x,y) = \frac{2\pi x}{L} + \Delta\varphi(\mathrm{x},\mathrm{y}) \tag{4.2}$$

最后的强度分布函数的相位来源与参考平面本身相位和由于物体高度变化而引起的相位变化之和。第 2，3 章已经得出Δφ(x,y)含有物体高度信息，因此数字相移法，最终要为得出Δφ(x,y)而服务。在上式子中(4.2)，b_o为背景光造成的灰度变化。

数字相移的做法是将一个同周期，但初始相位相差 δ 的条纹和所捕捉图像，灰度值矩阵每元素对应相乘，得到，

$$\begin{aligned} & I_{ps}(\mathrm{x},\mathrm{y}) \\ & = \mathrm{I}_o(\mathrm{x},\mathrm{y}) * \mathrm{I}_R(\mathrm{x},\mathrm{y}) \\ & = \left(b_o + \mathrm{rect}\left(\frac{2\pi x}{L} + \Delta\varphi(\mathrm{x},\mathrm{y})\right)\right) * \mathrm{rect}\left(\frac{2\pi x}{L} + \delta\right) \\ & = b_o * \mathrm{rect}\left(\frac{2\pi x}{L} + \delta\right) + \mathrm{rect}\left(\frac{2\pi x}{L} + \Delta\varphi(\mathrm{x},\mathrm{y})\right) * \mathrm{rect}\left(\frac{2\pi x}{L} + \delta\right) \end{aligned} \tag{4.3}$$

在上式中，第一项是高频条纹，第二项中利用傅立叶级数展开后相乘，基频分量积化和差后为

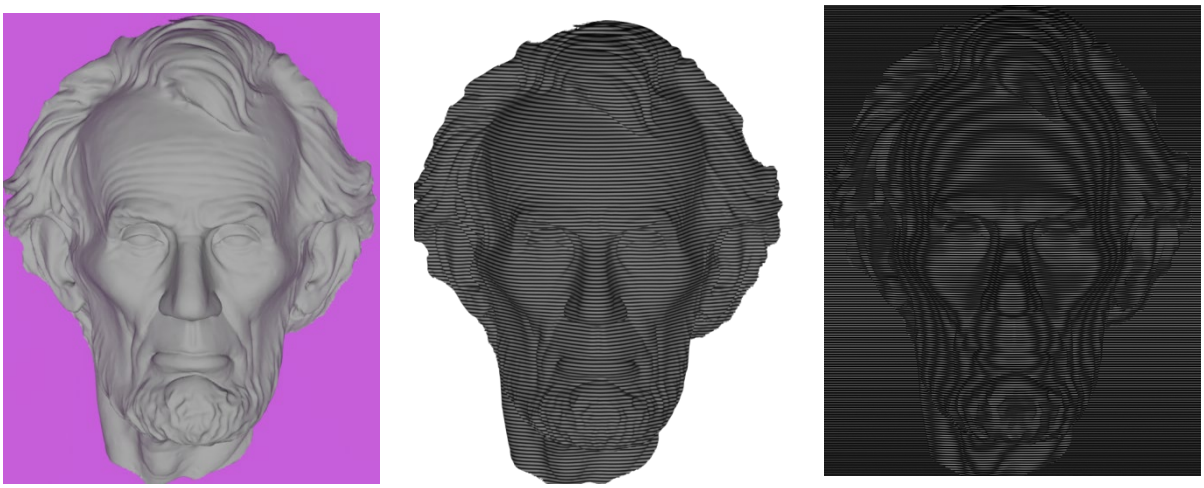
$$\cos\left(\frac{4\pi x}{L} + \Delta\varphi(\mathrm{x},\mathrm{y}) + \delta\right)$$
$$\cos(\Delta\varphi(\mathrm{x},\mathrm{y}) - \delta)$$

前者相对于$\cos(\Delta\varphi(x,y)-\delta)$同样属于高频变化成分，是被物体高度扭曲的投影条纹。假设通过滤波，可从上式子中分离出低频成分。

$$I_{filtered}=\cos(\Delta\varphi(x,y)-\delta) \tag{4.4}$$

则可通过相位提取公式，反解出$\Delta\varphi(x,y)$。具体推导过程详见第 5 章相位提取。

4.2 叠加结果分析

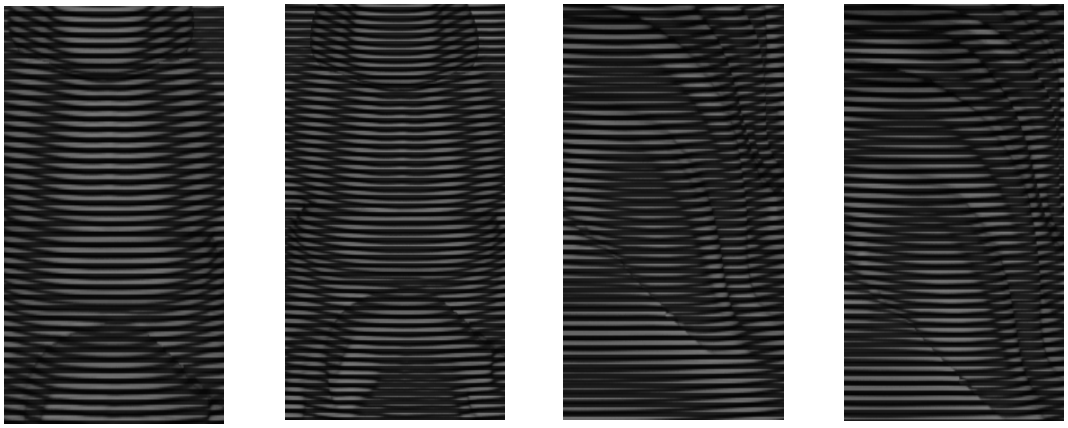


(a) (b) (c)

图 10 数字相移：(a)待测物体原型；(b)经过物体高度分布扭曲的投影条纹；(c)和同频率条纹叠加产生的含有高频噪声的莫尔图样

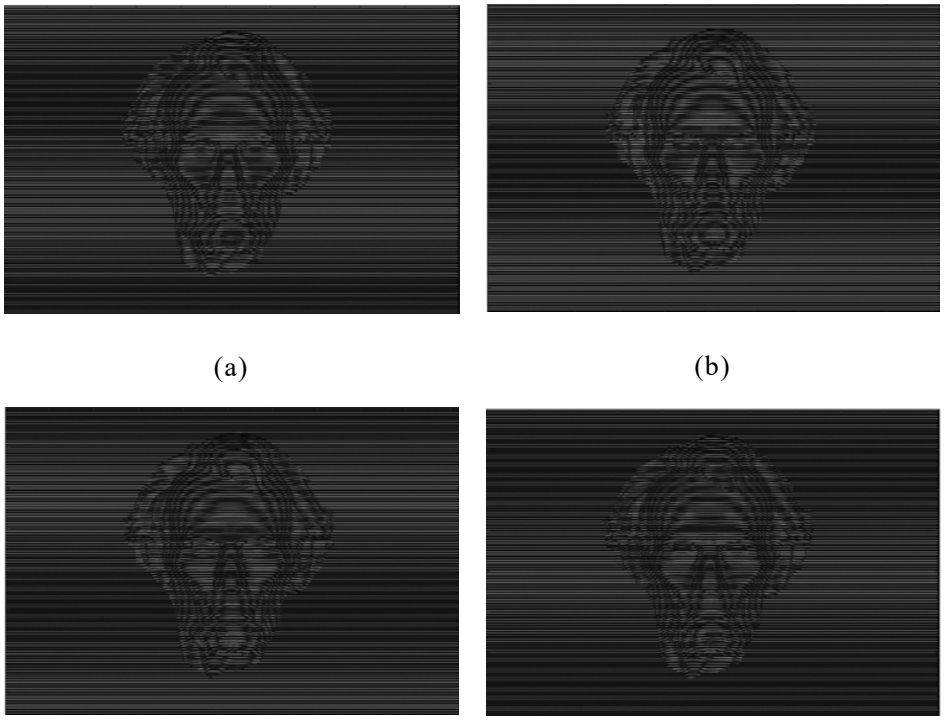
数字莫尔条纹生成需要同周期的条纹和捕捉到的图像这二者的灰度矩阵元素对应相乘，从而得出含有高频噪声的莫尔图样。在图 13(c)中，可观察到有与物体等高线对应的莫尔条纹。未出现莫尔现象处的相位与参考平面的相位差相同。

图 13 中，比较了条纹初始相位差为π，林肯脸三维模型发尖和鼻尖的相位对比。通过这两处的对比，根据相同强度的位置所对应的高度一致，则相位一致的条件，可以推断出，初始相位差导致莫尔条纹相位整体发生了移动。这一点验证了关于式子(4.3)中低频项$\cos(\Delta\varphi(x,y)-\delta)$对应莫尔条纹的假设。第 5 章将讨论如何过滤上图中的高频成分，得出对应被测面高度信息的低频成分，以便用于相位



(a) (b) (c) (d)

图 12 莫尔条纹和相位和初始相位关系：(a)δ = 0时，林肯脸鼻沟；(b)δ = π时，林肯脸鼻沟，与(a)图鼻沟处的莫尔条纹中有π的相位差；(c)δ = 0时，林肯脸发梢；(d)δ = π时，林肯脸发梢，与(c)图发梢处的莫尔条纹有π相位差；



(a) (b)

(c) (d)

图 11 不同初始相位的莫尔条纹

提取。在相位提取中，需要 4 张初始相位不同的莫尔图样，来计算折叠相位分布。但由于这 4 张莫尔图样，是在原捕捉图像上，通过叠加不同初始相位的同周期条纹得到。所以，数字莫尔三维测量只需捕捉一张图像，就能得到一张折叠相位图像。这是数字莫尔三维测量的一大优势，并为其提供了测量变形中，或运动

中物体的潜力。

4.3 滤除高频载波

根据前一节的讨论，为了得到去噪的莫尔条纹，以便于从其相位计算出物体高度分布，需要去除叠加的同周期、不同初始相位的高频条纹和被物体高度分布扭曲的投影条纹。为了达到这一目的，很多滤波方法被提出。在早期莫尔三维测量中，该工作是由测绘员勾勒莫尔等高线，费时复杂，误差大。随着计算机技术的发展，有学者利用快速傅立叶变换等算法，过滤噪声。但在傅立叶变换后，使用低通滤波器，也会使得图像细节模糊，信息丢失。为了解决这一问题，平稳小波变换结合低通滤波的方法被提出[8, 23, 24]。

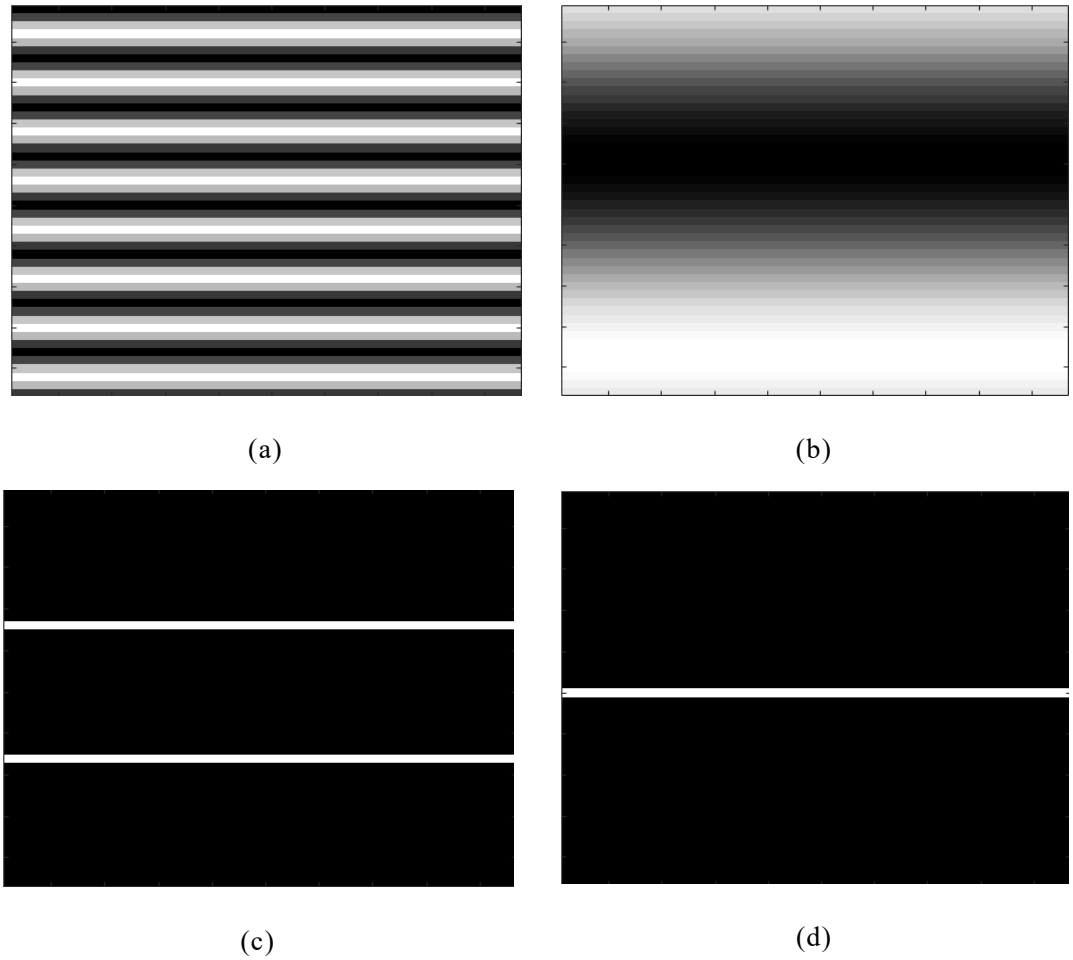
平稳小波变换，和离散小波变换不同，每经过高通滤波器或低通滤波器，都伴随一个上采样过程，最终得到的变换结果和原始数据个数一致。虽然实现平稳小波变化的方法较多，但均属于非抽取变换。在变化后，结果和原始数据维度相同[25][26]。同时，平稳小波变化具有时移不变性，即将信号 t(x)平移到$\mathrm{t}(\mathrm{x}-x_0)$经过平稳小波变换的结果相同。这一性质对于噪声消除极为重要。

在每一分解层，一维小波变换通过高通滤波器和低通滤波器，不断将信号二分成子带。二维小波变换则是分别在每一行和每一列，进行一维小波变换。

本论文去除高频信号的程序，参考了前人研究[8][24]。以下是滤波算法的具体步骤：

a) 根据分解层数 N，小波函数 W，利用 MATLAB 自带 swt2 函数得到近似系数 A，水平系数 H，竖直系数 V 和细节系数 D。四个系数的第三个纬度表示分解层数。例如，第三分解层的近似系数在 MATLAB 中表示为 A(:,:,3)。

b) 对于每一分解层：
   i. 只存在水平方向条纹，噪声集中在水平悉数中。因此，仅对水平系数进行傅立叶变换得到频域分布 FH。
   ii. 在水平系数矩阵中，只在列中出现高频率重复像素。因此，在傅立叶变换后，只需要针对竖直方向的空间频率低通滤波。



(a) (b)

(c) (d)

图 13 平稳小波傅立叶滤波条纹背景：(a)第三分解层水平系数；(b)第三分解层经过频域低通滤波后的水平系数；(c) 图(a)中系数傅立叶变化后的频谱幅度；(d)图(c)中频谱经过低通滤波后频谱幅度

iii. 在低通滤波后，傅立叶逆变换，得到滤波后水平系数矩阵。然后利用 MATLAB 自带平稳小波逆变换函数 iswt，带入未改变的近似系数 A, 竖直系数 V，和细节系数 D 以及滤波后的水平系数，可得到去除高频噪声的莫尔条纹图样。

假设向滤波函数，输入条纹背景灰度矩阵，并规定 N=3，W 为第五多贝西小波函数，得出的结果如上图。可由以下粗糙指数公式衡量滤波效果，即，

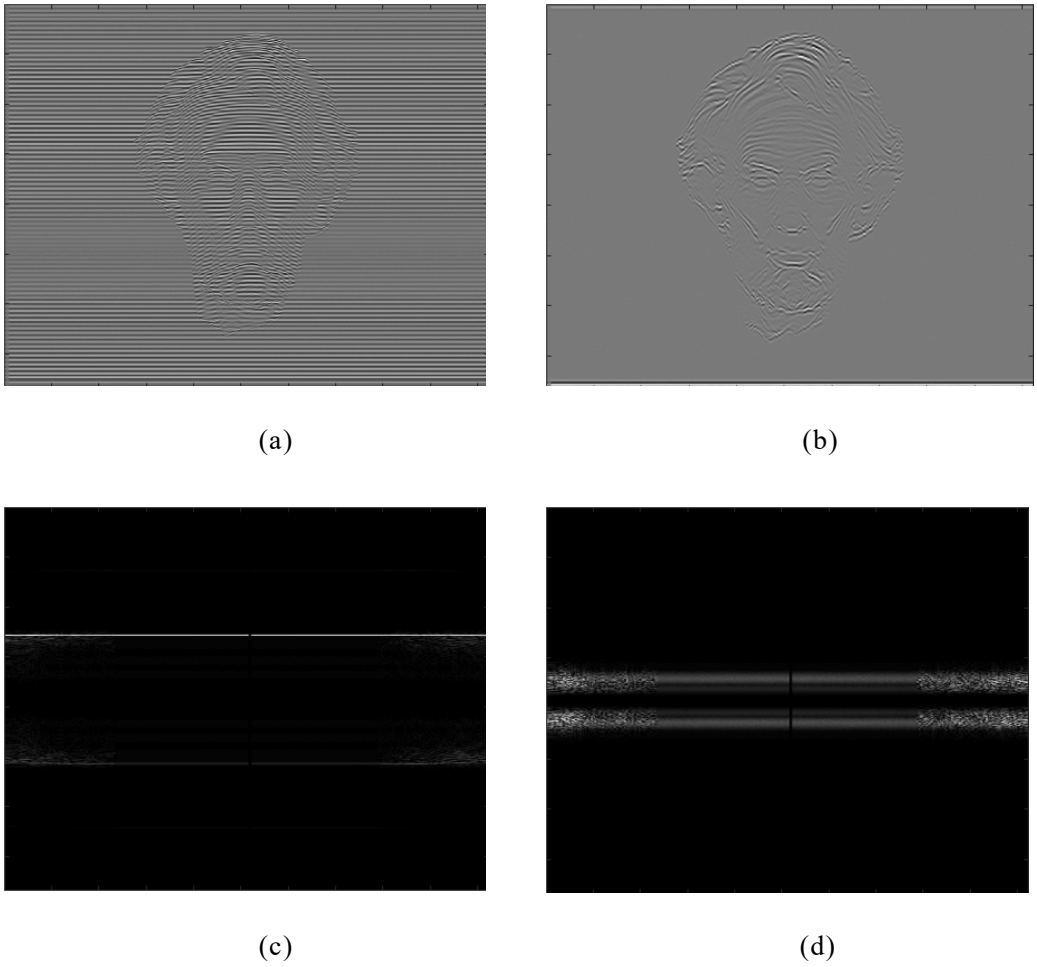
$$\rho = \frac{\|h \otimes X\|}{\|X\|} \tag{4.5}$$

其中，h 为索伯算子中检测横向边缘的卷积核。在$h \otimes X$中，横向条纹处，灰度值较大，而其他位置灰度值较小。

经过取模运算后，相对原图 X，该值越大，则横向条纹越多，滤波效果越差，这一标准也可用来优化多贝西小波函数，低通滤波带宽，平稳小波变换分解

层数等滤波参数。图 13(a)的粗糙指数为 6.3426，图 13(c)的粗糙指数为 0.3333。滤波后，粗糙指数相对降幅 94.74%。

如图 15 所示，图 15(a)是由平稳小波变换中第三分解层的水平系数，可见经过三次滤波后，叠加和物体高度分布扭曲的高频条纹集中水平系数中。图 15(c)是该水平系数经傅立叶变化后，频谱的幅度分布。在图 15(c)中，只在竖向频率出现



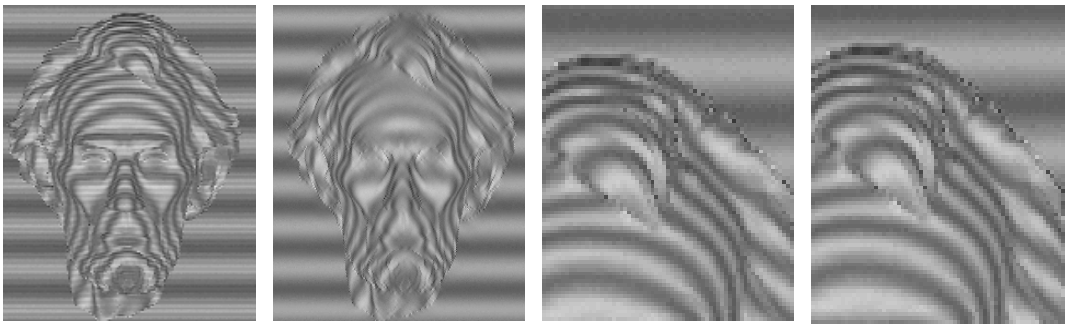
(a) (b)

(c) (d)

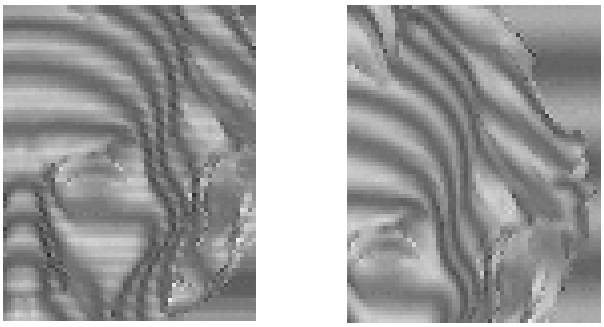
图 14 平稳小波傅立叶滤波林肯脸条纹叠加图：(a)第三层水平分解系数；(b)滤波后第三层分解层水平系数；(c)图(a)中系数傅立叶变化后的频谱幅度；(d)图(c)中频谱经过低通滤波后的频谱幅度

较高幅度分布，符合低通滤波算法部署时的假设——无需对水平空间频率滤波。在图 15(c)对应频率分布每一列乘以标准差为 10，平均值为 0 的高斯函数取样序列便得到了图 15(d)的频率分布。图 15(d)是经低通滤波后的频谱幅度图。 在图 15(b)中，可观察到莫尔等高线。最后，将经过低通滤波后的水平系数和其他系数，回带到 MATLAB 平稳小波逆变化中，得到最终强度分布结果，如图 16。

在优化滤波参数过程中，低通滤波的器标准差σ确定了低通带的带宽。σ太小，对高频过滤作用有限。σ太大，无法让莫尔等高线的频谱通过。分解层数 N 决定了经过子带二分数量，当分解层数恰好能分离出水平参数里的高频信号而保留莫尔图样，滤波效果最小。该参数过小，高频成分过滤效果差；该参数过大，得到的图像会严重变形。由于小波变换和傅立叶变化不同，完备正交对是具有局部特性的小波函数系，不同小波函数也会对图像细节产生影响。确定滤波参数的优化过



(a) (b) (c) (d)



(e) (f)

图 15 滤波参数优化，除特殊表明，参数为分解层数 3，小波函数为 db5，高斯低通滤波器标准差为 10：(a)分解层数为 2，仍有高频噪声；(b)分解层数为 5，细节模糊；(c)标准差为 1000，物体边缘变形；(d)标准差为 1000，物体边缘变形；(e)小波函数为 db1，保留了大量高频条纹；(f)小波函数为 db8，边缘变形

程一般可采取网格搜索优化，或先大致确定参数范围。

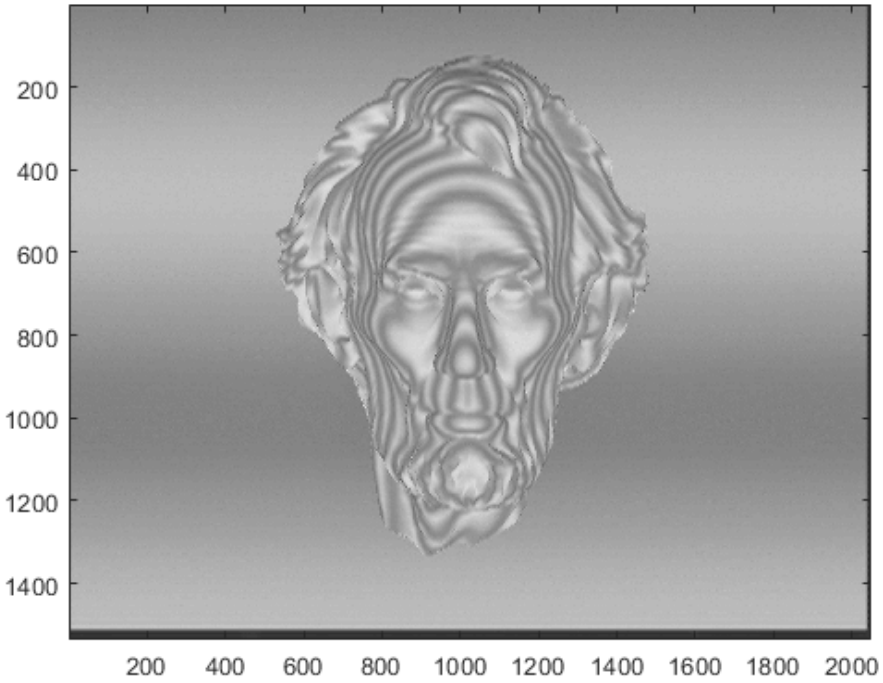
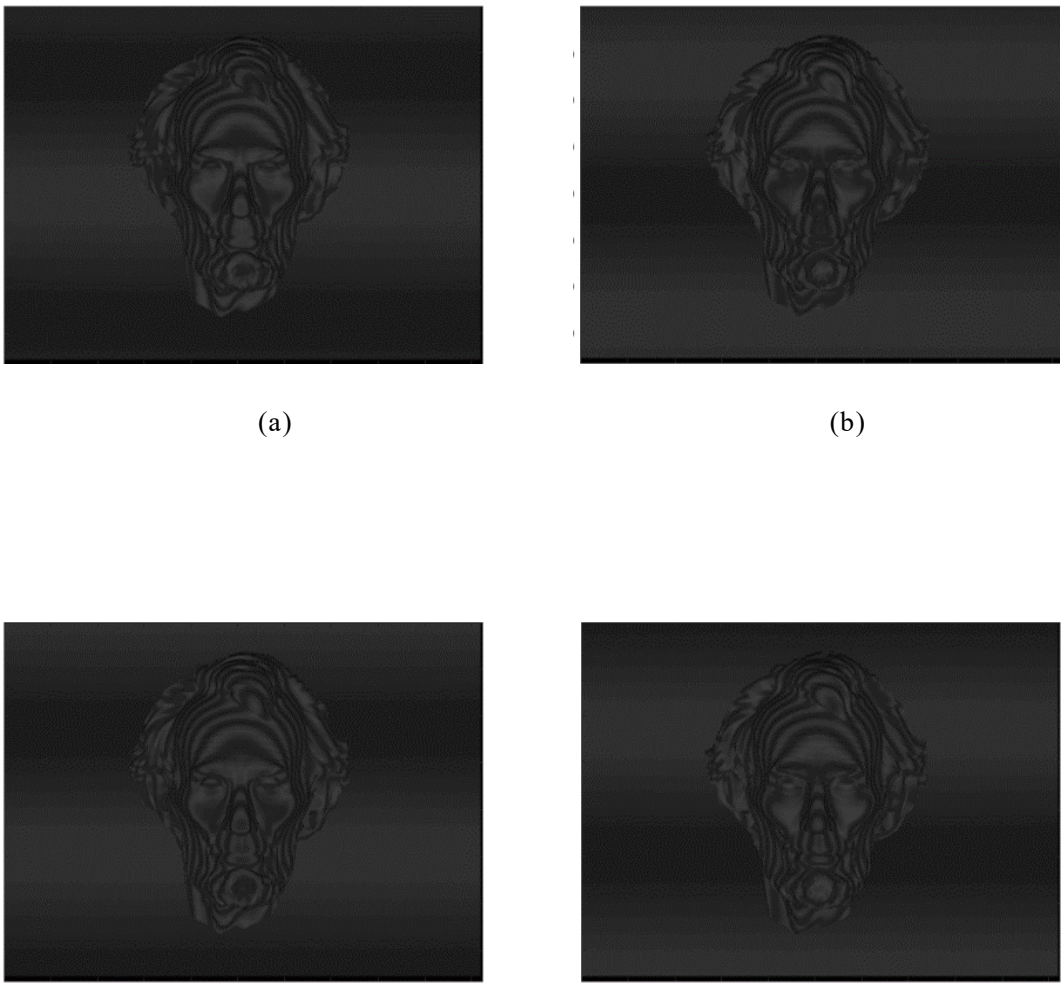


图 16 经平稳小波傅立叶滤波后的强度分布图

5 计算相位分布



(a) (b)

(c) (d)

图 17 滤波后的莫尔图样

4.1 节中，假设了通过滤波得到的强度分布可由下式(4.4)表述

$$I_{filtered} = \cos(\Delta\varphi(x,y) - \delta)$$

5.1 相位提取

观察图 21 不同初始相位的滤波后的莫尔图样，发现，有背景噪声和调制系数项，对(4.4)式进行修正，得，

$$I(x,y) = a(x,y) + b(x,y)\sin(\Phi(x,y) + \delta) \tag{6.1}$$

根据相位提取公式[18, 19]，

$$\Phi(\mathrm{x},\mathrm{y}) = -tan^{-1}\left(\frac{\sum_{\mathrm{i}=1}^{\mathrm{N}} \mathrm{I_i}(\mathrm{x},\mathrm{y})\sin\delta_{\mathrm{i}}}{\sum_{\mathrm{i}=1}^{\mathrm{N}} \mathrm{I_i}(\mathrm{x},\mathrm{y})\cos\delta_{\mathrm{I}}}\right), \mathrm{i} = 1,2,\dots,\mathrm{N} \tag{6.2}$$

在本论文中，采用了四个不同初始相位0，π，δ，δ + π 的叠加图，带入计算过滤后的强度分布公式(6.1)，得,

$$I_1(x,y) = a(x,y) + b(x,y)\sin\Phi(x,y) \tag{6.3}$$
$$I_2(x,y) = a(x,y) + b(x,y)\sin(\Phi(x,y) + \pi) \tag{6.4}$$
$$I_3(x,y) = a(x,y) + b(x,y)\sin(\Phi(x,y) + \delta) \tag{6.5}$$
$$I_4(x,y) = a(x,y) + b(x,y)\sin(\Phi(x,y) + \delta + \pi) \tag{6.6}$$

利用和差化积公式和相位反解公式可得，

$$\Phi(\mathrm{x},\mathrm{y}) = \arctan\left(\frac{\big(I_2(x,y) - I_1(x,y)\big)sin\delta}{\big(I_4(x,y) - I_3(x,y) - I_2(x,y) + I_1(x,y)\big)sin\delta}\right) \tag{6.7}$$

该公式是莫尔三维测量外差法求相位分布的方法[18, 20]。其中δ为平移一个像素，所对应的相位。将经过滤波的强度分布矩阵带入上式，可得相位分布。该

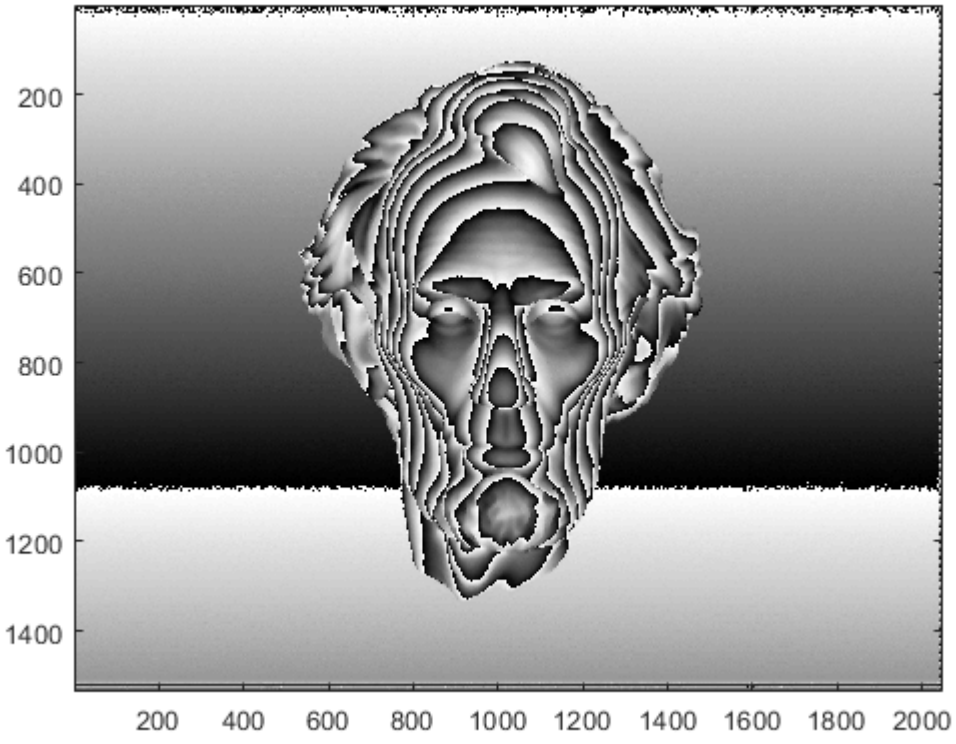


图 18 相位提取

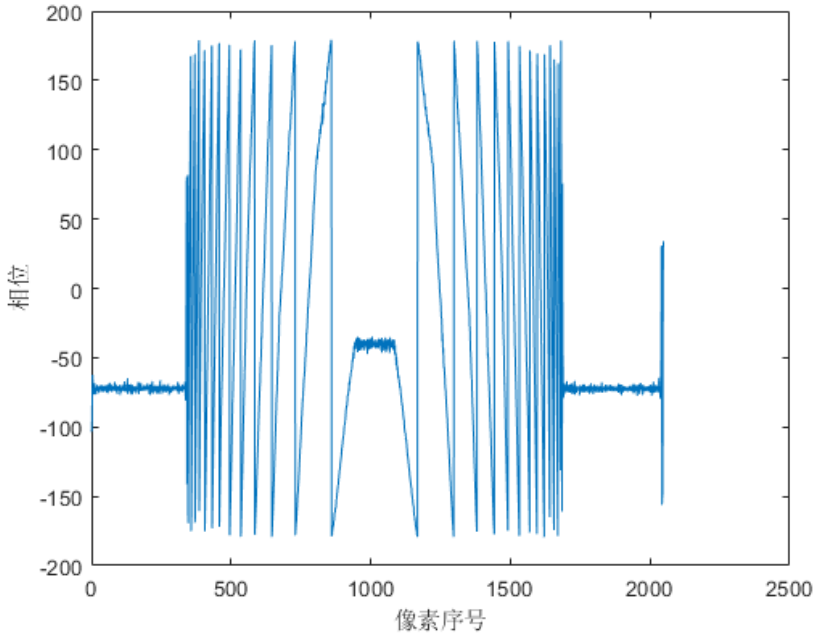
相位分布和被测物体高度并不是直接对应，并出现多个断带(不连续处)而被测物体高度无此突变。在利用(6.7)式，得出相位分布时，由于arctan函数所的求出相

位范围处于

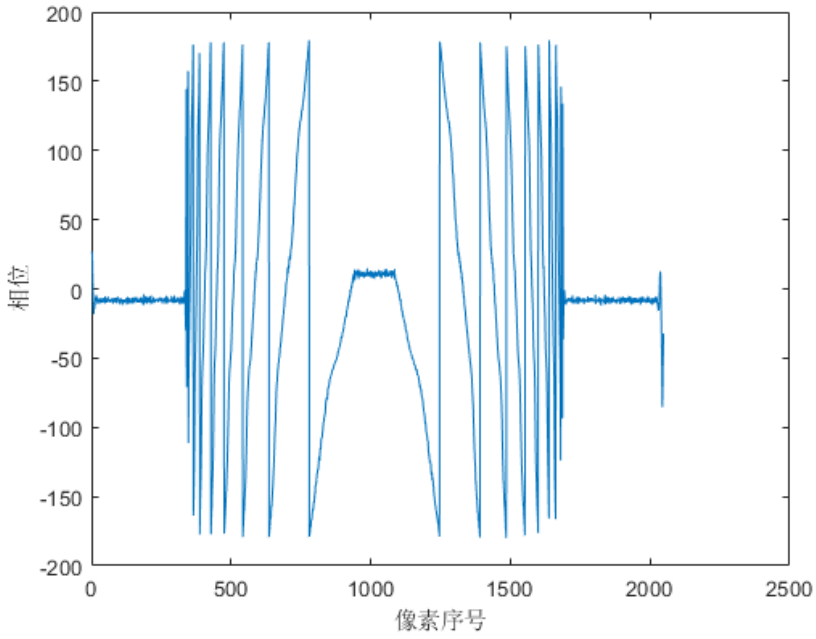
$$\left[-\frac{\pi}{2},\frac{\pi}{2}\right]$$

通过判断(6.7)式中分子分母符号，可将上述范围延拓到$[-\pi,\pi]$。但此范围仍不能和物体高度分布一一对应。这一相位分布称为折叠相位，需经过相位展开，和高度转换得出最终被测物体的三维模型。

5.2 相位展开



(a)



(b)

图 18 球体中截面折叠相位：(a)投影条纹周期为 6 个像素；(b)投影条纹周期为 10 个像素；

在经过了投影条纹图样，数字生成莫尔条纹，滤除高频噪声，和相位提取后，利用莫尔条纹的相位分布重建物体三维高度分布的问题转化成为如何消除折叠相位2π整数倍不确定性的问题。

时间相位展开[8]是一种利用莫尔波长展开折叠相位的方法，分为双波长和三波长时间相位展开方法。双波长时间相位展开方法通过比较不同周期条纹投影得到的折叠相位图，确定出参考相位分布。然后通过参考相位分布作为参考，在波长较小的折叠相位图上加减2π的整数倍。

首先计算参考相位，

$$\Phi_{12}\begin{cases}\Phi_1(x,y)-\Phi_2(x,y), & \Phi_1>\Phi_2\\ \Phi_1(x,y)-\Phi_2(x,y)+2\pi, & \text{其他}\end{cases} \tag{6.8}$$

在上式子中，Φ_1，Φ_2分别对应这莫尔波长λ_1，λ_2,而且$\lambda_1<\lambda_2$。相对与同一高度分布，小莫尔波长的相位分布应超前于大莫尔波长的相位分布。但在折叠相位分布里，相位限制在一个周期内。因此，当小莫尔波长的相位分布相对与大莫尔波长的相位分布出现延迟时，既是相位出现折叠时。此时，参考相位加一个周期。如图 19，对同一测量物体圆球，根据 3.1 式，在同一相机光轴-投影仪光轴夹角下，

$$\lambda=\frac{L}{tan\theta} \tag{6.9}$$

投影周期越大，莫尔波长越大。因此，图 19(a)对应Φ_1，图 19(b)对应Φ_1。当图 19(a)中的相位拖后与图 19(b)的时候，对二者相位差，补偿2π。以此参考相位展开小莫尔波长的相位分布，得

$$\varphi(x,y)=\Phi_1(x,y)+(2\pi)Round\left(\frac{\left(\frac{\lambda_{12}}{\lambda_1}\right)\Phi_{12}(x,y)-\Phi_1(x,y)}{2\pi}\right) \tag{6.10}$$

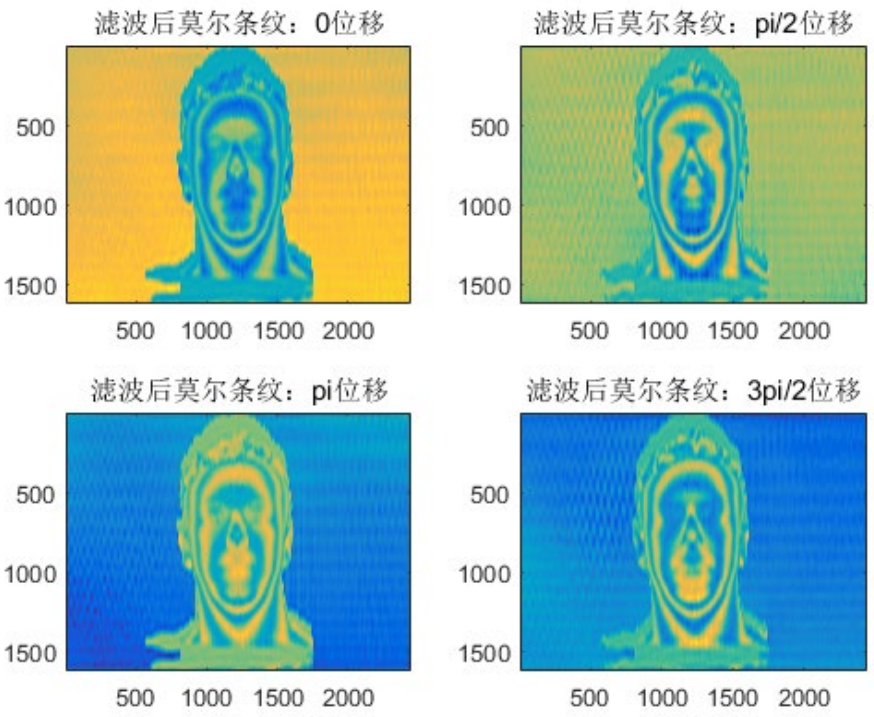
该展开相位分布，对应物体真实高度分布。而物体的真实高度分布为

$$\mathrm{H(x,y)}=\frac{\lambda_1}{2\pi}\varphi(x,y) \tag{6.11}$$

时间相位展开需要利用在系统校准时得到的对应不同条纹周期得到的莫尔波长和高度的关系。它使用物体高度范围中值处的莫尔波长作为整个物体的莫尔波长。由于本论文仅在仿真环境下实验数字莫尔方法的处理算法，对于相位展开的

结果[4]不作详细讨论。可从最终的展开相位中看到，根据人脸高度的不同，出现不同颜色的分布，相位展开方法完成从图 20(c)含有2π不确定性的折叠的相位分布到对应物体高度信息的连续相位分布。除了时间相位展开法，还有其他基于邻域突变判断的相位展开[21, 22]。但此类方法都无法保证物体本身突变细节的保留。

注 4 实物测量结果由本论文导师袁自钧教授和其研究生闫邵华先生提供，使用已征得同意。



(b)　　(c)

图 19 实物测量系统结果：(a)滤波后不通过初始相位的莫尔条纹；(b)由(a)中莫尔条纹计算出的折叠相位；(c)展开后的相位分布

6 结论

本论文基于加拿大学者关于数字莫尔三维测量的精度分析的文献，讨论了主流数字莫尔三维测量的实现方法（数字莫尔条纹的生成，高频噪声的滤除，相位分布的提取），并使用 MALTAB 科学计算环境和 3ds Max 三维建模软件验证。

根据第 2，3 章的讨论，数字莫尔三维测量技术是在传统莫尔三维形貌测量基础上，将一定的图像处理过程转移到计算机上。为了得到待测物体准确的三维形貌，一个较为完备的系统校准方法被提出。该方法基于传统莫尔相位和高度分布的关系。

数字莫尔三维测量技术的优势来源数字莫尔条纹产生的方法——将电脑合成的条纹与捕捉图像叠加。这一方法不但能满足后期相位提取需要多张不同初始相位的莫尔条纹的要求，而且能在只捕捉一张图像的情况下，就能得到折叠相位。如果使用自动相位展开方法，由一张捕捉图像得到的这一折叠相位可以转换为展开相位。对于不需要物体实际大小的三维测量任务，该方法具有明显优势。如果被测物体高度范围不超过一个莫尔波长，例如生物细胞，或 PCB 电路板等，则不需要展开相位[23]。

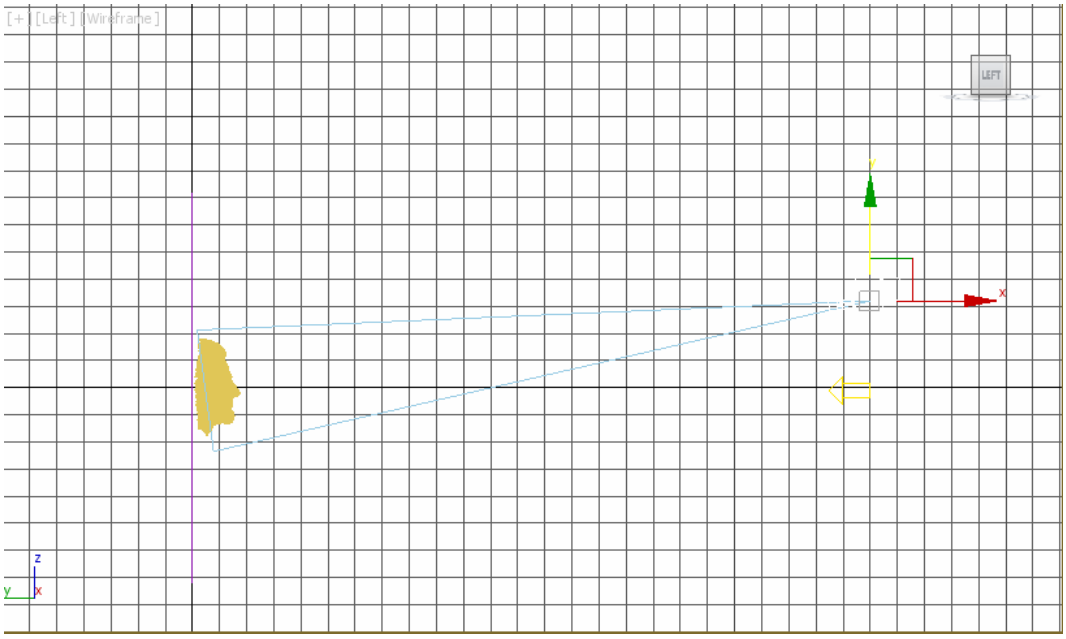


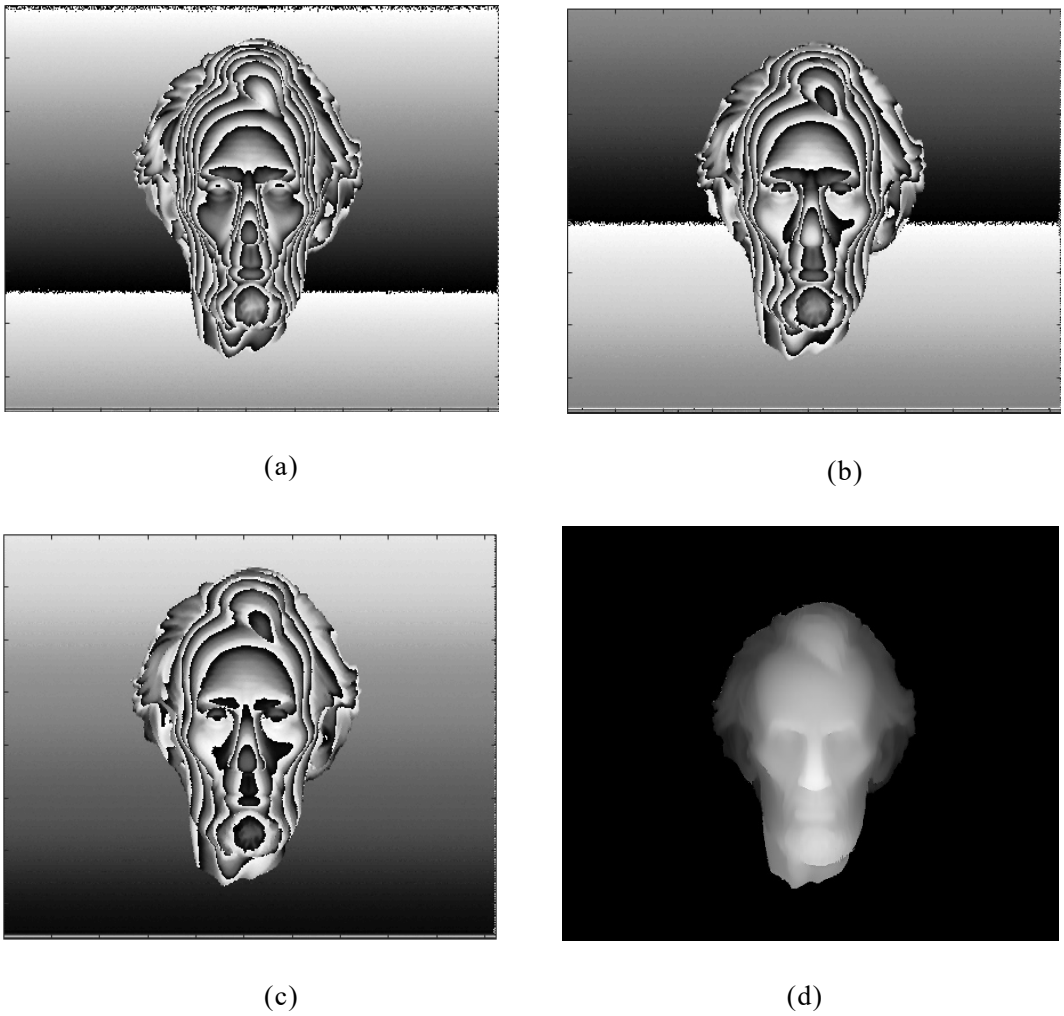
图 20 3ds Max 三维建模软件仿真环境设置

本论文使用了 3ds Max 三维仿真软件和 MATLAB 科学计算环境产生莫尔条纹。如图，在场景导入或创建被测物体。然后，离物体 2.5m 处，设置摄像机和方向灯。然后，调节摄像机的视场角和焦距，使视场完全覆盖所测物体。调节方

向灯的光阑参数，使投影范围大致和摄像机成像范围重合。在方向灯的高级选项中的映射选项，选择相位图，选择投影条纹投射。最后渲染得到捕捉图像。本论文仿真环境下的几何参数：相机-投影仪光轴夹角为 7°，相机-投影仪距离为 32 厘米，距参考平面距离为 2.5 米。渲染得到图像导入到 MALTAB，灰度化处理后，作为矩阵保存。

数字莫尔三维测量核心步骤是高频噪声的滤除。这一步骤关系着折叠相位和展开相位的计算。值得注意的是，无论是在 3ds Max 三维仿真软件中，还是在实际测量过程中，需检查捕捉图像和叠加条纹的周期相同。二者不相同时，会产生低频的背景条纹，滤除较困难。高频噪声的滤除，除了选择合适的算法外，还需要对滤波参数进行调优。这一过程需针对具体被测物体，大大减小了该三维测量方法的实用性。这也是本文的缺陷这一，未能针对高频噪声的滤除提出使用范围更广但参数更少的算法。目前已经有学者，将深度学习技术应用到莫尔测量上，取得了初步的成果[24, 25]，深度学习模型由于具有大量参数，可根据数据优化模型等特点，可作为未来滤除高频噪声的研究方向。

相位提取已有成熟理论支持。但相位展开，仍然需要复杂的系统校准过程和处理算法。相位提取推荐使用后期叠加生成的 4 张不同的相位的莫尔条纹计算折叠相位。相位展开需根据效果选择表现较为满意的算法。图 21 中使用了 6 像素，8 像素，10 像素为周期的条纹投影到被测物体，林肯脸雕塑上。图 21(a)(b)(c)是有投影条纹后捕捉的三张图像得到的折叠相位。对比这三张图，可以看出，当条纹周期较小，得出的莫尔条纹圈数较多。这意味着利用周期短条纹投影，可以得到更高的灵敏度。但同时需注意，莫尔波长的相对波长能覆盖被测物



(a) (b)

(c) (d)

图 21 林肯脸测量结果：(a)条纹周期 6 个像素的折叠相位；(b)条纹周期 8 个像素的折叠相位；(c)条纹周期 10 个像素的折叠相位；(d)利用莫尔波长得到的展开相位；

体高度范围。该被测物体使用的滤波参数：6 像素，8 像素，10 像素的投影条纹，滤波参数为$\sigma = 50$，第 5 号小波多贝西函数，3 分解层数。最后得出的相位分布如图 25(d)。图 26 是林肯脸的三维重建模型，可以观察到相机光轴平行的部分，无法得到准确的高度模型，这也是在固定方向上使用数字莫尔三维测量的一个缺陷。

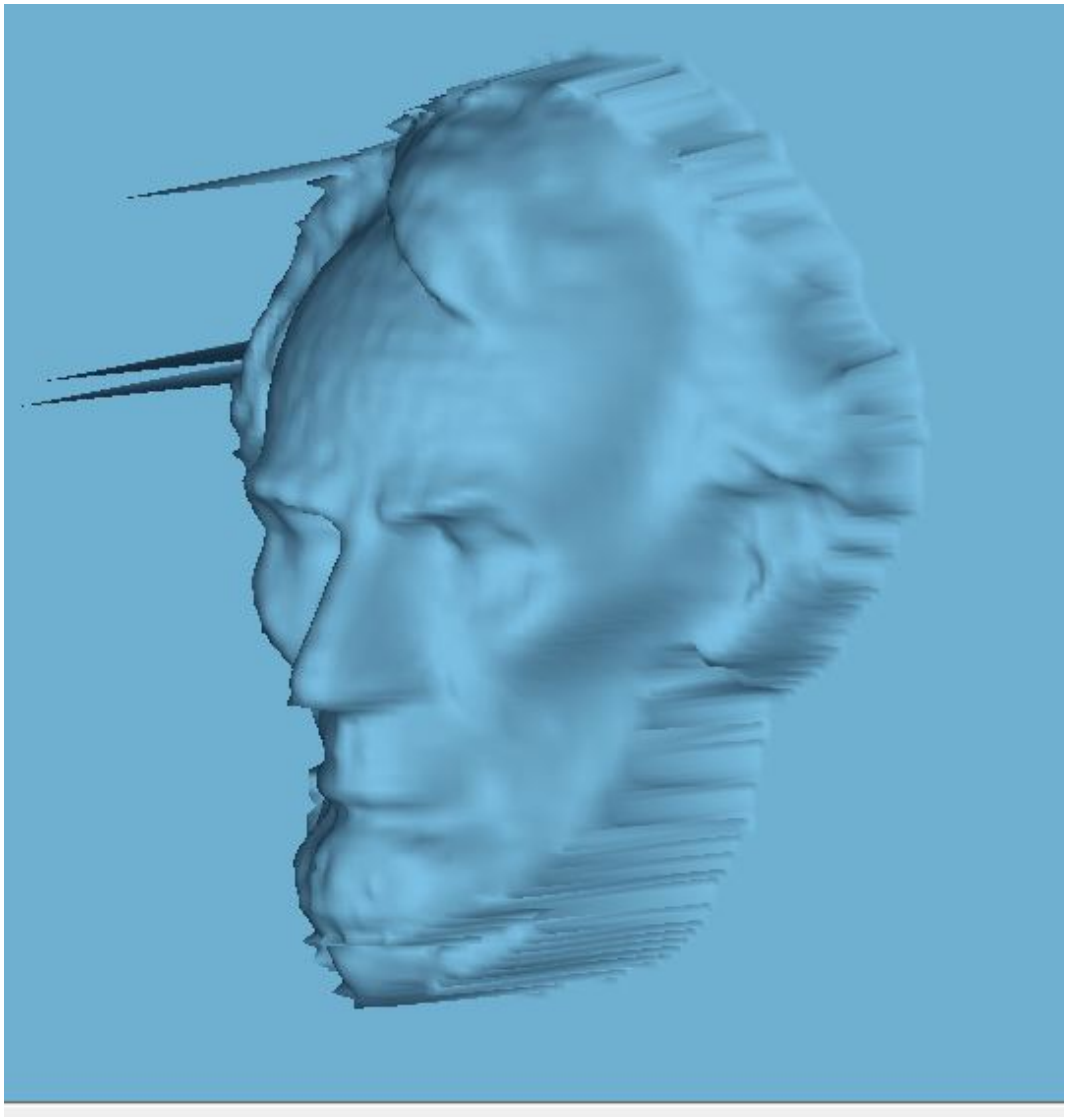


图 22 林肯脸三维重建模型

参考文献

[1] 曹向群, 黄维实 1990. 莫尔技术的现状和展望. 光电工程 [J]: 48-56.

[2] 丁一飞 2016. 数字光栅投影测量关键技术研究 [M]. 合肥工业大学.

[3] 朱丽君, 王玉荣, 孟祥锋, et al. 2015. 数字合成莫尔条纹的频谱分析与滤波处理. 中国激光 [J], 42: 256-264.

[4] 朱丽君 2016. 数字莫尔条纹三维面形测量技术研究 [M]. 山东大学.

[5] 鹿丽华, 胡摇, 王劭溥, et al. 2018. 数字莫尔移相干涉仪误差多点标定与修正研究. 仪器仪表学报 [J], 39: 77-84.

[6] WARDEN R, AL RATROUT S 2005. Moiré Contours for Documenting Petroglyphs at Montezuma Castle.

[7] GORTHI S S, RASTOGI P 2010. Fringe projection techniques: whither we are? Optics and lasers in engineering [J], 48: 133-140.

[8] MOHAMMADI F 2017. 3D optical metrology by digital moiré: Pixel-wise calibration refinement, grid removal, and temporal phase unwrapping.

[9] MOHAMMADI F, MADANIPOUR K, REZAIE A H 2010. Application of digital phase shift moiré to reconstruction of human face [C] //, IEEE; City. 306-309.

[10] MOHAMMADI F, KOFMAN J 2016. Improved grid-noise removal in single-frame digital moiré 3D shape measurement. Optics and lasers in engineering [J], 86: 143-155.

[11] TAKASAKI H 1970. Moiré topography. Applied Optics [J], 9: 1467-1472.

[12] 李恩泽 2016. 光栅投影法三维轮廓测试关键技术研究 [M]. 西安工业大学.

[13] ZHOU C, SI S, LI X, et al. 2018. Dynamic 3D shape measurement based on the phase-shifting moir\'e algorithm. arXiv preprint arXiv:1807.01399 [J].

[14] NISHIJIMA Y, OSTER G 1964. Moiré patterns: their application to refractive index and refractive index gradient measurements. JOSA [J], 54: 1-5.

[15] AMIDROR I, HERSCH R D 2010. Mathematical moiré models and their limitations. Journal of Modern Optics [J], 57: 23-36.

[16] CREATH K, WYANT J 1992. Moiré and fringe projection techniques. Optical shop testing [J], 2: 653-685.

[17] JIA P, KOFMAN J, ENGLISH C E 2007. Comparison of linear and nonlinear calibration methods for phase-measuring profilometry. Optical Engineering [J], 46: 043601.

[18] DIRCKX J J, DECRAEMER W F 1990. Automatic calibration method for phase shift shadow moiré interferometry. Applied Optics [J], 29: 1474-1476.

[19] SURREL Y 1996. Design of algorithms for phase measurements by the use of phase stepping. Applied Optics [J], 35: 51-60.

[20] MOHAMMADI F, KOFMAN J 2019. Multi-Wavelength Digital-Phase-Shifting Moiré Based on Moiré Wavelength. Applied Sciences [J], 9: 1917.

[21] VENEMA T M, SCHMIDT J D 2008. Optical phase unwrapping in the presence of branch points. Optics express [J], 16: 6985-6998.

[22] KARASEV P A, CAMPBELL D P, RICHARDS M A 2007. Obtaining a 35x speedup in 2d phase unwrapping using commodity graphics processors [C] //, IEEE; City. 574-578.

[23] LIU Y 2011. Accuracy improvement of 3D measurement using digital fringe projection.

[24] FENG S, CHEN Q, GU G, et al. 2019. Fringe pattern analysis using deep learning. Advanced Photonics [J], 1: 025001.

[25] XIE J, XU L, CHEN E 2012. Image denoising and inpainting with deep neural networks [C] //; City. 341-349.

致谢

本论文是在指导老师袁自均副教授的悉心指导和严格要求下完成的。袁老师学识严谨，认真负责。袁老师耐心给予我专业的指导，启发我将实践和理论结合。在专业学习上，老师不断鼓励我，多尝试，多思考。同时，在学习之余，袁老师还给予我很多无私的帮助、鼓励和温暖。非常感谢袁老师的指导！

特别感谢高伟清老师。高老师给予我许多宝贵的建议，使我更好地思考论文。

同时，还要谢谢同门的闫邵华等学长。学长热心帮助我，给了我很多启发。

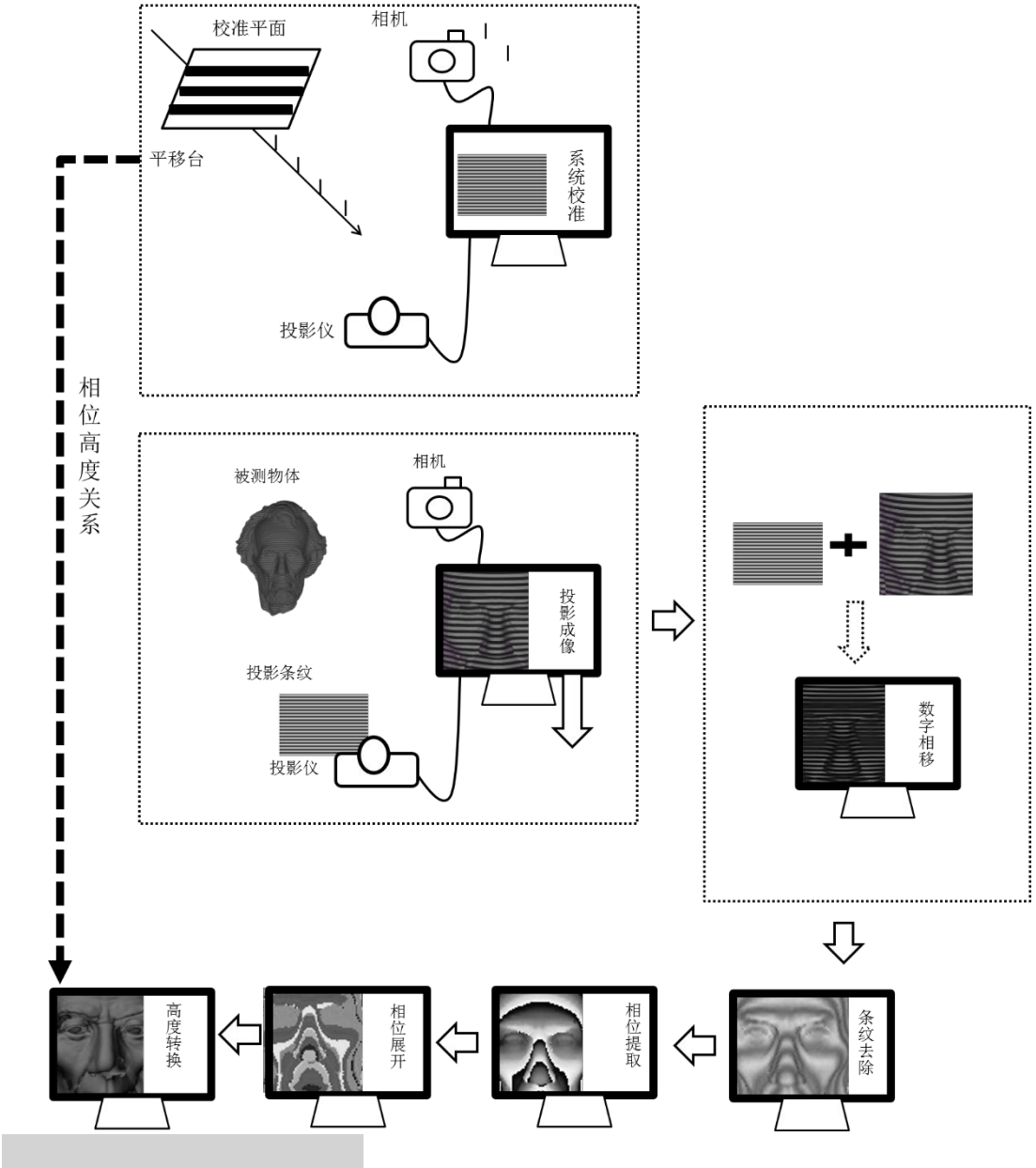
最后，我还要谢谢我的父母。没有他们默默的关心和支持，我不可能顺利完成学业。

作者：张凡

2019 年 05 月 30 日

附录

附录-1. 数字莫尔三维测量流程图



数字莫尔三维测量 MATLAB 程序

After capturing a single frame of image, shift the pattern by a phase of pi, delta, and delta + pi; Then superimpose the two phase-shifted pattern with the captured image

```
% Generate virtually two phase-shifted images and display them
% Input:
%     figPath - the directory path of the captured image
%     patternPath - the directory path of used pattern
%     deltaPixel - the smallest pixel moved
%     isDisplay - logic true
% Output：
%     figZeroPS - the dfig of grayscalized caputured image
%     figPiPS - the fig of grayscalized image with the phase
%     shift of pi
%     figDeltaPS - the fig of grayscalized image with the phase delta;
%     figDeltaPiPS - the fig of grayscalized image with the phase delta + pi
function [figZeroPS, figPiPS, figDeltaPS, figDeltaPiPS] =
digitalMorieSuperimpose(figPath, patternPath, deltaPixel, isDispaly)
    currentPath = pwd();
    cd("..\Patterns");
    directoryPath = pwd();
    patternPath = char(patternPath);
    patternName = patternPath(length(directoryPath)+1:end); % add the / simbol
    cd(currentPath);
    % take out the width
    [startIndex, endIndex] = regexp(patternName, 'w[\d]*_');
    width = str2num(patternName(startIndex+1:endIndex-1));

    % take out the height
    [startIndex, endIndex] = regexp(patternName, 'h[\d]*_');
    height = str2num(patternName(startIndex+1:endIndex-1));
```

```
% take out the minimum gray value
[startIndex, endIndex] = regexp(patternName, 'g[\d]*_');
minGray = str2num(patternName(startIndex+1:endIndex-1));

% take out the maxium gray value
[startIndex, endIndex] = regexp(patternName, '_[\d]*_');
maxGray = str2num(patternName(startIndex+1:endIndex-1));

% take out the wavelength
[startIndex, endIndex] = regexp(patternName, 'wl[\d]*_');
pitch = str2num(patternName(startIndex+2:endIndex-1)); %#ok<*ST2NM>

% take out the phase
[startIndex, endIndex] = regexp(patternName, 'p[\d]*.[\d]*_');
phase = str2num(patternName(startIndex+1:endIndex-1));

% take out the binary or sinuidal signal
isBinary = (patternName(end-4) == 'B');

% new phases
phaseZero = phase;
phasePi = phase + pitch / 2;
phaseDelta = phase + deltaPixel;
phaseDeltaPi = phase +    pitch / 2 + deltaPixel;

[pattern0, ~] = generatePattern(width, height, minGray, maxGray, pitch,
phaseZero, isBinary);
[pattern1, ~] = generatePattern(width, height, minGray, maxGray, pitch, phasePi,
isBinary);
[pattern2, ~] = generatePattern(width, height, minGray, maxGray, pitch,
phaseDelta, isBinary);
```

```
[pattern3, ~] = generatePattern(width, height, minGray, maxGray, pitch,
phaseDeltaPi, isBinary);

figCaptured = inputDeformedImage(figPath);

% save generated figures
[figZeroPS, figPath0] = superimposeSingle(pattern0, figCaptured ,figPath,
phaseZero);
[figPiPS, figPath1] = superimposeSingle(pattern1, figCaptured ,figPath, phasePi);
[figDeltaPS, figPath2] = superimposeSingle(pattern2, figCaptured, figPath,
phaseDelta);
[figDeltaPiPS, figPath3] = superimposeSingle(pattern3, figCaptured ,figPath,
phaseDeltaPi);

% display the generated figures path
disp("Figure on " + figPath0 + " is generated.");
disp("Figure on " + figPath1 + " is generated.");
disp("Figure on " + figPath2 + " is generated.");
disp("Figure on " + figPath3 + " is generated.");


% display the image
if isDispaly
    figure('Position', [100, 100, 2048, 1536]);
    colormap('gray');

    subplot(2, 2, 1);
    imagesc(figZeroPS, [0, 1]);
    title("Zero Phase-shift");

    subplot(2, 2, 2)
```

```
            imagesc(figPiPS, [0, 1]);
            title("\pi Phase-shift ");

            subplot(2, 2, 3)
            imagesc(figDeltaPS, [0, 1]);
            title("\delta Phase-shift ");

            subplot(2, 2, 4)
            imagesc(figDeltaPiPS, [0, 1]);
            title("\delta + \pi Phase-shift")
        end
end

% Superimpose the phase-shifted pattern and captured image
% Input:
%     pattern - the phase-shifted the pattern matrix
%     figCaptured - the captured frame matrix, which is already grayscalized
%     figPath - the directory path of captured image
%     phase - the phase value from the pattern matrix
% Output:
%     newFig - the phase-shifted image matrix
%     newFigPath - the phase-shifted image directory path
function [newFig, newFigPath] = superimposeSingle(pattern, figCaptured, figPath,
phase)
    figPath = char(figPath);
    newFigPath = figPath(1:end-6) + "p" + num2str(phase) + ".bmp";
    newFig = figCaptured .* pattern;
    imwrite(newFig, char(newFigPath));
end
```

Generate binary or sinusoidal patternwith specific paramters: the min and max grayvalue, wavlength(have to be perfect divided by the width), and image height and

```
width
    % Generate the pattern array and save the image according to specific parameters
function [pattern, figPath] = generatePattern(w, h, minGray, maxGray, pitch, phase,
isBinary)
% Check whether the image is already generated
    figName = "w" + int2str(w) + "_h" + int2str(h) + "_g" + int2str(minGray) + "_" +
int2str(maxGray) + "_wl" + int2str(pitch) + "_p" + num2str(phase, '%0.2f\n');
    if isBinary
        figName = figName + "_B";
    else
        figName = figName + "_S";
    end
    figName = figName + ".bmp";
    current_dir = pwd();
    cd("..\Patterns");
    if(exist(figName))
        disp('Pattern ---- ' + figName + ' is already generated!');
        pattern = imread(char(figName)); % save for future use
        pattern = mat2gray(pattern, [0, 225]);
    else
        if isBinary
            pattern = generateBinaryPatternArray(w, h, minGray, maxGray, pitch,
phase);
        else
            pattern = generateSinPatternArray(w, h, minGray, maxGray, pitch,
phase);
        end
        imwrite(pattern, char(figName), 'BMP');
        disp('Pattern ---- ' + figName + ' is generated!');
    end
    figPath = pwd() + "\" + figName;
```

```
    cd(current_dir);
end

% Generate the sinusoidal pattern array
function pattern = generateSinPatternArray(w,h,minGray, maxGray, wavelength, phase)
    positions = linspace(0, h-1, h)';
    amplitude = (maxGray - minGray) / 2;
    standard = (maxGray + minGray) / 2;
    sinWaveMask = standard + amplitude * sin(2 * pi * positions / wavelength +
phase);
    pattern = sinWaveMask * ones([1, w]); % expand as horizontal stripes
    pattern = mat2gray(pattern, [0, 255]); % grayscalize
end

% Generate the binary pattern array
function pattern = generateBinaryPatternArray(w, h, minGray, maxGray, wavelength,
phase)
    positions = linspace(1 + phase, h    + phase, h)';
    amplitude = (maxGray - minGray) / 2;
    standard = (maxGray + minGray) / 2;
    squareWaveMask = standard + amplitude * squareIntergerWave(positions,
wavelength);
    pattern = squareWaveMask * ones([1, w]); % expand as horizontal stripes
    pattern = mat2gray(pattern, [0, 255]); % grayscalize
end

% generate a square wave signal with respect to a single pixel
% Input:
%     positions - series of pixels positions, which is already added the
%     phase shift, starting with one
%     wavelength - period of the recurring pattern
```

```
function signal = squareIntergerWave(positions, wavelength)
    halfWavelength = floor(wavelength/2);
    signal = -1 * ones(size(positions));
    for i = 1:length(positions)
        r = mod(positions(i), wavelength);
        if r < halfWavelength
            signal(i) = 1;
        end
    end
end
    This file contains functions that help to extract phase from 4 captured deformed
and grid removed image;
    % Etract WrappedPhase and display the image according to paramter
%
% Input:
%    figZeroPS - the fig matrix with zero phase shift
%    figPiPS - the fig matrix with pi phase shift
%    figDeltaPS - the fig matrix with delta phase shift
%    figDeltaPiPS - the fig matrix with delta    + pi phase shift
%    ddecNum - the optimized decomposition number
%
%
% Output:
%   wrappedPhase - the wrapped phase matrix
function wrappedPhase = extractWrappedPhaseSWTFFT(figZeroPSFiltered,
figPiPSFiltered, figDeltaPSFiltered, figDeltaPiPSFiltered, delta, isDisplay)
    figZero =    (figPiPSFiltered - figZeroPSFiltered) /2;
    figDelta = (figDeltaPiPSFiltered - figDeltaPSFiltered) / 2;
    wrappedPhase = atan2d(figZero * sin(delta), figDelta - figZero * cos(delta));
    if isDisplay
        displayFig(wrappedPhase, "test for extracting wrapped phase")
```

```
    end
end
    This file display the image under a gray colormap
    %Input:
% figMatrix - the Matrix that    represent the grayscalized image
% titleString - the fig title
function displayFig(figMatrix, titleString)
    figure
    colormap('gray');
    imagesc(figMatrix);
    title(titleString);
end
    This file contains function that can calculate the Morie Wavelength according to
the system geometric parameters
    function lambda = getMorieWavelengthGeo(theta, pitch)
    lambda = pitch / tan(theta);
end
    This file contains function that calculates Rough Index for all four deformed
images
    % Calculate the Rough Index for one image by convolving the image with
% Sobel vertical edge detector filter
% Input:
%     fig - the fig matrix
% Output:
%     roughIndex - the rough Index of the image

function roughnessIndex = getRoughnessIndex(fig)
    h = [-1 -2 -1; 0 0 0; +1 +2 +1];
    verticalEdge = conv2(h, fig);
    roughnessIndex = norm(verticalEdge, 1) / norm(fig, 1);
end
```

```
    This file contains the function    that repeat the whole process to extract the
wrapped phase for a specific morie Wavelength
    % getSingleUnwrappedPhase, this function calculate a single wrapped phase
% use previous functions
% Input:
%     pitch - the pitch of the fringe pattern
%     lambda - the wavelength of the morrie pattern
% Output:
%     wrappedPhase - the wrapped Phase

function wrappedPhase = getSingleWrappedPhase(pitch)
    [~, patternPath] = generatePattern(2048, 1536, 50, 200, pitch, 0, 1);
    prompt = 'the pattern path is ' + patternPath + '.\nPlease paste the captured image
directory path below:\n';
    figPath = input(char(prompt));
    [figZeroPS, figPiPS, figDeltaPS, figDeltaPiPS] =
digitalMorieSuperimpose(figPath, patternPath, 1, false);
    decum = input(char("Please enter the decomposition level: \n"));
    dampingFactor = input(char("please enter the damping factor: \n"));
    [figZeroPSFiltered, figPiPSFiltered, figDeltaPSFiltered, figDeltaPiPSFiltered] =
removeGridSWTFFT(figZeroPS, figPiPS, figDeltaPS, figDeltaPiPS, decum, 'db5',
dampingFactor, false);
    delta = 1 / pitch * 2 * pi;
    wrappedPhase = extractWrappedPhaseSWTFFT(figZeroPSFiltered,
figPiPSFiltered, figDeltaPSFiltered, figDeltaPiPSFiltered, delta, false);
end
    This helper grayscalizes the captured image
    % Read and grayscalize the image file
%
% Input:
%     figPath0 - the directory path of image file
```

```
%
% Output:
%      fig - the grayscalized pattern matrix
function fig = inputDeformedImage(figPath)
      figPath = char(figPath); % the url have to a characters
      fig = imread(figPath);
      figSize = size(fig);
      if figSize(end) == 3
            fig = rgb2gray(fig); % grayscale
      end
      fig = mat2gray(fig, [0, 255]); % normalized
end
      This file contains function that removes the Grid for four different images with
SWTFFT method
      %Remove the grid noises according to SWTFFT method
% Input:
%      figZeroPS - the fig matrix with zero phase shift
%      figPiPS - the fig matrix with pi phase shift
%      figDeltaPS - the fig matrix with delta phase shift
%      figDeltaPiPS - the fig matrix with delta    + pi phase shift
%      decNum - the optimized decompostion level, an interger
%      wName - the wavlet function that performs the best
%      sigma - the optimized the damping factor
%      isDisplay - the boolean variable; when it is true, the result will be
%      shown
%
%
% Output:
%      figZeroPSFiltered - the filtered fig matrix with zero phase shift
%      figPiPSFiltered - the filtered fig matrix with pi phase shift
%      figDeltaPSFiltered - the filtered fig matrix with delta phase shift
```

```
%     figDeltaPiPSFiltered - the filtered fig matrix with delta    + pi phase shift
function [figZeroPSFiltered, figPiPSFiltered, figDeltaPSFiltered, figDeltaPiPSFiltered]
= removeGridSWTFFT(figZeroPS, figPiPS, figDeltaPS, figDeltaPiPS, decNum,
wName, sigma, isDisplay)
    figZeroPSFiltered = SWTFFT(figZeroPS, decNum, wName, sigma);
    figPiPSFiltered = SWTFFT(figPiPS, decNum, wName, sigma);
    figDeltaPSFiltered = SWTFFT(figDeltaPS, decNum, wName, sigma);
    figDeltaPiPSFiltered = SWTFFT(figDeltaPiPS, decNum, wName, sigma);

    if isDisplay
        figure('Position', [100, 100, 2048, 1536]);
        colormap('gray');

        subplot(2, 2, 1);
        imagesc(figZeroPSFiltered, [0, 1]);
        title("Zero Phase-shift Filtered");

        subplot(2, 2, 2)
        imagesc(figPiPSFiltered, [0, 1]);
        title("\pi Phase-shift Filtered ");

        subplot(2, 2, 3)
        imagesc(figDeltaPSFiltered, [0, 1]);
        title("\delta Phase-shift Filtered");

        subplot(2, 2, 4)
        imagesc(figDeltaPiPSFiltered, [0, 1]);
        title("\delta + \pi Phase-shift Filtered")
    end
end
```

Remove stipes and other noises in the superimposed images by SWT-FFT filtering

```
method.
    % Input:
% fig       - the superimposed figure matrix
%   decNum - the decomposition level number, an interger
%   wName    - the wavelt function name, 'db5', 'db12'
%   sigma    - the gaussian damping factor
function figFiltered = SWTFFT(fig, decNum, wName, sigma)
    % wavelet decompostion
    [A, H, V, D] = swt2(fig, decNum, wName);

    % fourier transform
    HDamped = zeros(size(H)); % only need to look at the horizontal coeffecient

    for i = 1:decNum
        % FFT
        HFFTed = fft(H(:, :, i));
        HFFTed = fftshift(HFFTed);
        [M, N] = size(HFFTed);

        % damping of horizontal stipe
        gaussianDampingX = -floor(M / 2) : -floor(M / 2) + M    - 1;
        gaussianDampingY = normpdf(gaussianDampingX, 0, sigma);
        gaussianDamping2D = repmat(gaussianDampingY', 1,    N);

      HFFTed = HFFTed .* gaussianDamping2D;
      %inverse FFT
      HFFTed = ifftshift(HFFTed);
      HDamped(:, :, i) = ifft(HFFTed);
    end

    % wavelet reconstruction
```

```
    figFiltered = iswt2(A, HDamped, V, D, wName);
 end
    This file contains function that can unwrap the phase
    % unwrapPhase3, unwrap the phase by three-wavelength phase unwrapping
% Input:
%     wrappedPhase1 - wrapped phase according to lambda1
%     wrappedPhase2 - wrapped phase according to lambda2
%     wrappedPhase3 - wrapped phase according to lambda3
% Output:
%     unwrappedPhase - the unwrappedPhase
function unwrappedPhase =    unwrapPhase3(wrappedPhase1, wrappedPhase2,
wrappedPhase3, lambda1, lambda2, lambda3)
    lambda12 = getRelativeWavelength(lambda1, lambda2);
    lambda23 = getRelativeWavelength(lambda2, lambda3);
    lambda123 = getRelativeWavelength(lambda12, lambda23);
    unwrappedPhase12 = unwrapSinglePhaseAB(wrappedPhase2, lambda2, lambda12,
wrappedPhase1);
    unwrappedPhase23 = unwrapSinglePhaseBA(wrappedPhase2, lambda2, lambda23,
wrappedPhase3);
    unwrappedPhase = unwrapSinglePhaseABC(wrappedPhase2, lambda2,
lambda123, unwrappedPhase12, unwrappedPhase23);
end

% getRelativeWavelength
% Input:
%     lambdaA - the first wavelength
%     lambdaB - the second wavelength
% Output:
%     lambdaAB - the relative wavelength
function lambdaAB = getRelativeWavelength(lambdaA, lambdaB)
    lambdaAB =    lambdaA * lambdaB / abs(lambdaA - lambdaB);
```

```
end

% getRelativeWavelength
% Input:
%      phaseA - the first phase
%      phaseB - the second phase
% Output:
%      phaseAB - relative phase
function phaseAB = getRelativePhase(phaseA, phaseB)
      phaseAB = phaseA - phaseB;
      mask = sign(-sign(phaseAB) + 1) * 2 * pi;
      phaseAB = phaseAB +    mask;
end

% getPhaseFiltered
% Input:
%      phase - the phase to unwrap
%      lambda - the wavelenght of corresponding morrie pattern
%      relativePhase - the calculated relative phase
%      relativeLambda - the calculated relative wavelength
function phaseFiltered = getPhaseFiltered(phase, lambda, relativePhase,
relativeLambda)
      phaseFiltered = phase +    2 * pi * round( (relativeLambda / lambda * relativePhase
- phase) / (2 * pi));
end

% unwrapSinglePhaseAB
% Input:
%      phaseB - the phase to unwrap
%      lambdaB - the wavelenght of corresponding morrie pattern
%      relativeLambda - the calculated relative wavelength
```

```
%      phaseA - the wrapped phase for referrence
function phaseUnwrapped = unwrapSinglePhaseAB(phaseB, lambdaB, relativeLambda,
phaseA)
      relativePhase = getRelativePhase(phaseA, phaseB);
      phaseUnwrapped = getPhaseFiltered(phaseB, lambdaB, relativePhase,
relativeLambda);
end

% unwrapSinglePhaseBA
% Input:
%      phaseB - the phase to unwrap
%      lambdaB - the wavelenght of corresponding morrie pattern
%      relativeLambda - the calculated relative wavelength
%      phaseA - the wrapped phase for referrence
function phaseUnwrapped = unwrapSinglePhaseBA(phaseB, lambdaB, relativeLambda,
phaseA)
      relativePhase = getRelativePhase(phaseB, phaseA);
      phaseUnwrapped = getPhaseFiltered(phaseB, lambdaB, relativePhase,
relativeLambda);
end

% unwrapSinglePhaseABC
% Input:
%      phaseB - the phase to unwrap
%      lambdaB - the wavelenght of corresponding morrie pattern
%      relativeLambda - the calculated relative wavelength
%      phaseAB - the unwrapped phase for referrence
%      PhaseBC - the unwrapped phase for referrence
function phaseUnwrapped = unwrapSinglePhaseABC(phaseB, lambdaB,
relativeLambda, phaseAB, phaseBC)
      relativePhase = getRelativePhase(phaseAB, phaseBC);
```

```
    phaseUnwrapped = getPhaseFiltered(phaseB, lambdaB, relativePhase,
relativeLambda);
end
```