

第十一届数学中国数学建模网络挑战赛

地址：数学中国数学建模网络挑战赛组委会
电话：0471-4969085

邮编：010021

网址：www.tzmcm.cn
Email: service@tzmcm.cn

第十一届“认证杯”数学中国

数学建模网络挑战赛

承 诺 书

我们仔细阅读了第十届“认证杯”数学中国数学建模网络挑战赛的竞赛规则。

我们完全明白，在竞赛开始后参赛队员不能以任何方式（包括电话、电子邮件、网上咨询等）与队外的任何人（包括指导教师）研究、讨论与赛题有关的问题。

我们知道，抄袭别人的成果是违反竞赛规则的，如果引用别人的成果或其他公开的资料（包括网上查到的资料），必须按照规定的参考文献的表述方式在正文引用处和参考文献中明确列出。

我们郑重承诺，严格遵守竞赛规则，以保证竞赛的公正、公平性。如有违反竞赛规则的行为，我们接受相应处理结果。

我们允许数学中国网站(www.madio.net)公布论文，以供网友之间学习交流，数学中国网站以非商业目的的论文交流不需要提前取得我们的同意。

我们的参赛队号为： 5163

参赛队员（签名）：

队员 1： 王浅宁

队员 2： 郭子君

队员 3： 许子涵

参赛队教练员（签名）： 张贺祥

参赛队伍组别（例如本科组）：本科组

第十一届数学中国数学建模网络挑战赛

地址：数学中国数学建模网络挑战赛组委会
电话：0471-4969085

邮编：010021

网址：www.tzmcm.cn
Email: service@tzmcm.cn

第十一届“认证杯”数学中国

数学建模网络挑战赛

编号专用页

参赛队伍的参赛队号：（请各个参赛队提前填写好）：

5163

竞赛统一编号（由竞赛组委会送至评委团前编号）：

竞赛评阅编号（由竞赛评委团评阅前进行编号）：

第十一届数学中国数学建模网络挑战赛

地址：数学中国数学建模网络挑战赛组委会
电话：0471-4969085

邮编：010021

网址：www.tzmcm.cn
Email: service@tzmcm.cn

2017 年第十一届“认证杯”数学中国 数学建模网络挑战赛第一阶段论文

题 目 基于盲去卷积原理的动态模糊图像复原算法

关 键 词 动态模糊 盲去卷积 参数估计 图像复原

摘 要

图像退化的成因众多，动态模糊是造成图像降质的主要原因之一。在曝光时间内，景物与摄像机之间的相对运动产生了像移量，导致动态模糊。为了解决动态模糊导致的图像退化问题，本文提出了基于盲去卷积原理的图像复原算法。

本文将 Brenner 梯度函数作为动态模糊图像复原评价标准。即计算图像中相邻两个像素灰度差的平方，其函数定义为： $D(f) = \sum_y \sum_x |f(x+2, y) - f(x, y)|^2$ ，取阈值 $T = 0$ 。

当 $D(f)$ 数值越大则图像越清晰，复原效果最好。本文首先对典型复原算法进行优化，降低噪声对算法复原能力的影响。然后运用 Brenner 梯度函数对逆滤波算法、维纳滤波算法、最小二乘滤波算法进行评价。其值分别为 4.82×10^7 、 4.76×10^7 和 5.50×10^5 。由此可知逆滤波复原效果较好。

但由于使用上述算法进行图像复原后，视觉效果并不十分理想。我们在其原理上进行优化，提出了基于盲去卷积的匀速直线运动模糊图像复原算法。动态模糊的实质是清晰图像在一段距离上的叠加，因此建立匀速直线运动退化模型

$$g(x, y) = \frac{1}{a} \sum_{i=0}^{a-1} f(x-i, y) \Delta t。$$

接下来需要确定点扩散函数 PSF 的主要参数：模糊角度 θ 、模糊长度 L 。首先在频域中，将模糊图像频谱图进行二值化及 Sobel 边缘检测得到较清晰暗纹；再运用 Hough 变换提取图像中亮度为 255 的像素点，转换为另一个域中的曲线，通过曲线分析相交次数确定对应直线信息，得到模糊角度；将原图像转化为水平方向的运动模糊，统计频谱图中暗条纹个数即为模糊长度。以题目所给模糊图像为例，其模糊角度为 88° ，模糊长度为 31 像素。最终通过盲去卷积将图像进行还原处理、USM 锐化以及色相饱和度还原得出最终复原图像，图中可见字母为 ENDING。

参赛队号： 5163

所选题目： B 题

参赛密码 _____
(由组委会填写)

第十一届中国数学建模网络挑战赛

地址：数学中国数学建模网络挑战赛组委会
电话：0471-4969085

邮编：010021

网址：www.tzmcm.cn
Email: service@tzmcm.cn

Abstract

There are many reasons of image degradation. Dynamic blurring is one of the reasons for the reduction in image quality. During the exposure time, the relative motion between scene and camera produces an image shift, which results in dynamic blur. In order to solve the problem of dynamic blurring, we propose a dynamic fuzzy image restoration algorithm based on the Blind Deconvolution principle.

We use the Brenner Gradient Function, which calculates the square of the gray difference of two adjacent pixels, as the evaluation criterion for dynamic blurred image restoration. It is defined as $D(f) = \sum_y \sum_x |f(x+2, y) - f(x, y)|^2$. And its threshold is

$T = 0$. The larger $D(f)$, the sharper image, the better the effect of restoration. We optimize the typical restoration algorithm to reduce the effect of noise on the resilience of the algorithm. Then we use the Brenner Gradient Function to evaluate the Inverse Filter Algorithm, the Wiener Filter Algorithm, and the Least Squares Filter Algorithm. The results are 4.82×10^7 , 4.76×10^7 and 5.50×10^5 . Obviously, the Inverse Filter Algorithm is better.

Due to the visual effect is not very satisfactory. We optimize the above principles and propose a blurred motion image restoration algorithm based on Blind Deconvolution. The essence of dynamic blurring is the superposition of clear images over a distance. Establishing

an uniform linear motion degradation model as $g(x, y) = \frac{1}{a} \sum_{i=0}^{a-1} f(x-i, y) \Delta t$. Then we need to

determine the main parameters of the point spread function PSF: fuzzy angle θ , fuzzy length L . First, in the frequency domain, the binarization of the blurred image spectrum map and Sobel Edge Detection result in clearer dark lines. Then we use the Hough Transform to extract pixels with a brightness of 255 in the image and convert it to a curve in another domain. The corresponding straight line information is determined by analyzing the number of intersections of the curve, and the fuzzy angle is obtained. Next, the original image is transformed into motion fuzzy in the horizontal direction. The number of dark stripes in the statistical spectrum map is the fuzzy length. Take the blurred image given by the title as an example, the fuzzy angle is 88° , and the fuzzy length is 31 pixels. Finally, we use blind deconvolution, USM sharpening, and hue saturation reduction to obtain the final restored image. The letter in the picture is “ENDING”.

Key words: Dynamic Blurring, Noise Processing, Parameter Estimation, Image Restoration

1 模糊图像的成因分析

1.1 图像退化及其分类

造成图像退化的原因有很多，典型原因表现为：一，成像系统的象差、畸变、带宽有限等造成的图像失真；二，由于成像器件拍摄姿态和扫描非线性引起的图像几何失真；三，动态模糊，成像传感器与被拍摄景物之间存在相对运动，引起所成图像的动态模糊；四，灰度失真，光学系统或成像传感器本身特性不均匀，造成同样亮度景物成像灰度不同；五，辐射失真，由于场景能量传输通道中的介质特性如大气湍流效应，大气成分变化引起图像失真；六，图像在成像、数字化、采集和处理过程中引入的噪声。而本文所研究内容为动态模糊，故下文均研究动态模糊图像复原方法。

1.2 动态模糊原理

图像在获取的过程中，可能出现畸变、模糊、失真或混入噪声等问题，使所成图像降级，称为图像的退化。而引起图像退化的原因有很多。成像系统的象差、畸变、带宽有限会造成图像失真；成像器件拍摄姿态和扫描的非线性会引起几何失真；成像传感器与被拍摄景物之间存在相对运动，而引起所成图像的动态模糊；光学系统或成像传感器本身因特性不均匀，造成同样亮度景物成像灰度失真；由于场景能量传输通道中的介质特性，如大气湍流效应，大气成分变化等，而引起的辐射失真；图像在成像、数字化、采集和处理过程中引入的噪声。为了尽可能改善图像质量，最大程度恢复退化图像，本文以动态模糊为方向研究图像复原的有效方法。

动态模糊的产生是由于成像传感器与被拍摄景物之间存在相对运动。在用摄像机获取景物图像时，如果在相机曝光时间内，景物与摄像机之间存在相对运动，所获得的图像可能存在模糊的现象。在日常生活中动态模糊是相当常见的。对电影的截图来说，动态画面的每一帧都是模糊的；高速飞行中的拍摄设备俯拍到的路面字迹也是模糊的；交通路口的电子眼所拍摄的违章行为车辆也容易形成动态模糊，导致很难准确获取车辆信息。

我们以高速运动的摄像机为例讲述动态模糊的基本原理。

当摄像机以速度 V 在空中飞行时，如图所示，地面景观 A 点相对摄像机向后移动到 A' 。通过光学系统成像于 a' 点，在 CCD 靶面上像移速度为：

$$V' = \frac{V}{H} f'_{\max} \quad (1-1)$$

式中， V 为摄像机飞行速度； H 为摄像机高度； f'_{\max} 为光学系统最大焦距。

在 CCD 摄像机每场积分时间内像移量为：

$$\Delta l = V' t(mm) \quad (1-2)$$

式中， t 为 CCD 摄像机的场积分时间。

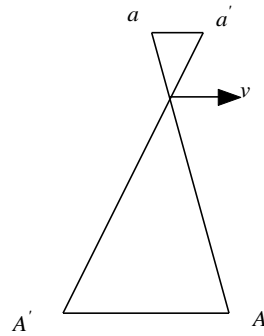


图 1-1 运动示意图

像移量的存在导致图像模糊，为得到清晰图像，必须要对像移进行控制。在实际工程中，*CCD*的积分时间不能无限的缩小，而且高帧频*CCD*的价格很贵。积分时间缩短后，为了保证图像质量，所需的地面照度就越大，这就限制了相机的工作条件，在许多情况下是不能接受的。

目前解决动态模糊的主要手段是通过了解图像的退化过程，建立运动图像的复原模型，通过数学模型来解决图像的复原问题。

2 动态模糊图像复原方法及比较

2.1 图像复原评价体系的建立

在无参考图像的质量评价中，图像的清晰度是衡量图像质量优劣的重要指标，它能够较好的与人的主观感受相对应，图像的清晰度不高表现出图像的模糊。本文针对无参考图像质量评价应用，给出一个较为常用的、具有代表性清晰度算法进行解释分析，为实际应用中选择清晰度算法提供依据。

Brenner 梯度函数是最简单的梯度评价函数，它只是简单的计算相邻两个像素灰度差的平方，该函数定义如下：

$$D(f) = \sum_y \sum_x |f(x+2, y) - f(x, y)|^2 \quad (2-1)$$

其中： $f(x, y)$ 表示图像 f 对应像素点 (x, y) 的灰度值， $D(f)$ 为图像清晰度计算结果。上式表明， $D(f)$ 所得数值越大图像越清晰且取阈值 $T = 0$ 。下文所述方法均可由该式计算，且结果可由数值比对分析得出。

2.2 不同图像复原方法原理分析

动态模糊图像的恢复是图像复原的主要分支之一，它的恢复算法有很多种。有些算法虽然有很好的恢复效果，但算法复杂，恢复时间比较长（如最大熵法）。有些算法虽然计算速度较快，但恢复效果不尽人意（如空间域逆向恢复）。

下面介绍逆滤波算法、维纳滤波算法和最小二乘滤波算法三种恢复方法的原理。

2.2.1 逆滤波算法原理

恢复退化图像最简单的方法是直接逆滤波。在该方法中，用退化函数除退化图像的傅里叶变换 $F(u,v)$ 来计算原始图像的傅里叶变换估计 $\hat{G}(u,v)$ ，由式(2-2)可以得到逆滤波退化公式：

$$\hat{G}(u,v) = \frac{F(u,v) - N(u,v)}{H(u,v)} \quad (2-2)$$

该公式说明逆滤波对于无噪声污染的图像很有效，这里不考虑当 $H(u,v)$ 接近 0 时可能遇到的计算问题，幸运的是忽略这些点在恢复结果中并不会产生可感觉到的影响。但是，如果出现噪声就会引起几个问题：第一，对于 $H(u,v)$ 幅值比较小的频率处噪声的影响可能变得显著起来。这种状况通常对于高频 u,v 。在实际中，通常 $H(u,v)$ 幅值衰减得比 $N(u,v)$ 快得多，因此噪声的影响可能支配整个复原结果。将复原限定在 $H(u,v)$ 足够大得 u,v 原点处得一个小邻域中，可以克服这个问题。为了克服 $H(u,v)$ 接近 0 所引起的问题，在分母中加入一个小的常数 k ，将式(2-2)修改为：

$$\hat{G}(u,v) = \frac{F(u,v) - N(u,v)}{H(u,v) + k} \quad (2-3)$$

2.2.2 维纳滤波算法原理

一般地，噪声源往往具有平坦的功率谱，即使不是如此，其随着频率的升而下降的趋势也要比典型图像的功率谱慢得多。因此，可以料想功率谱的低频分以信号为主，然而高频部分则主要被噪声所占据。由于逆滤波滤波器的幅值常随着频率的升高而升高，因此会增强高频部分的噪声。为了克服以上缺点，出了采用最小均方误差的方法（维纳滤波）进行模糊图像恢复。

维纳(wiener)滤波可以归于反卷积(或反转滤波)算法一类，它是由 Wiener 首提出的，并应用于一维信号，并取得很好的效果。以后算法又被引入二维信号理，也取得相当满意的效果，尤其在图像复原领域，由于维纳滤波器的复原效良好，计算量较低，并且抗噪性能优良，因而在图像复原领域得到了广泛的应用并不断得到改进发展，许多高效的复原算法都是以此为基础形成的。

$$Q = R_g^{-\frac{1}{2}} R_n^{-\frac{1}{2}} \quad (2-4)$$

R_g 和 R_n 分别是图像和噪声的自相关矩阵。即 $R_g = E\{gg^T\}$ ， $R_n = E\{nn^T\}$ ，并且都是正定对称矩阵，则有

$$\hat{g} = (H^T H + \gamma R_g^{-1} R_n)^{-1} H^T f \quad (2-5)$$

$\hat{Q}g = R_g^{1/2} R_n^{1/2} \hat{g}$ 的模方最小，实际上就意味着使噪声和信号的比对复原图像影响最小。因为图像和噪声的相关矩阵都是把图像当作随机过程来研究，从而描述其统计特性的量，在这里最小二乘方的最佳已经演变成均方误差最小准则下的最佳。

同样根据式(2-5)可求得频域维纳滤波公式如下

$$\hat{G}(u,v) = \frac{1}{H(u,v)} \cdot \frac{|H(u,v)|^2}{|H(u,v)|^2 + \gamma \frac{S_n(u,v)}{S_g(u,v)}} \cdot F(u,v) \quad (2-6)$$

$\gamma=1$ 时，为标准维纳滤波器； $\gamma \neq 1$ 时，为含参维纳滤波器。

实际应用中必须调节 γ 以满足上式。因为 $S_n(u,v), S_g(u,v)$ 实际很难求得因此，可以用一个比值 k 代替两者之比，从而得到简化的维纳滤波公式

$$\hat{G}(u,v) = \frac{1}{H(u,v)} \cdot \frac{|H(u,v)|^2}{|H(u,v)|^2 + k} \cdot F(u,v) \quad (2-7)$$

2.2.3 最小二乘滤波算法

由于大多数图像恢复问题都不具有唯一解，或者说恢复具有病态特征。为了克服这一问题，通常需要在恢复过程中对运算施加某种约束。

设对图像施加某一线性运算 Q ，求在约束条件下，使 $\left| Q \hat{g} \right|^2$ 为最小的 \hat{g} 作为原图 g 的最佳估计。

$$\left| f - H \hat{g} \right|^2 = |n|^2 \quad (2-8)$$

我们可以直接从空间域的有约束最小二乘方恢复式(3-10)得到它的频域解：

$$\hat{G}(u,v) = \frac{1}{H(u,v)} \cdot \frac{|H(u,v)|^2}{|H(u,v)|^2 + \gamma |C(u,v)|^2} \cdot F(u,v) \quad (2-9)$$

应用有约束最小二乘方恢复方法时，只需有关噪声均值和方差的知识就可对每幅给定的图像给出最佳恢复结果。

2.3 三种滤波方法应用对比

上文所述三种动态模糊图像复原方法均可在 *matlab* 环境下实现，下面将对每一种方法举例实践，并对结果做出合理性分析。

下图为一全局模糊图像，采用上述三种方法对图像进行处理，可得到以下结果，其中详细运行代码见附录。

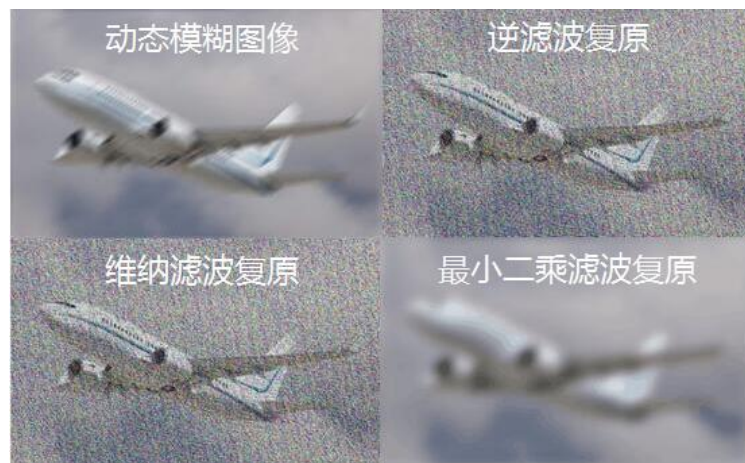


图 2-1 结果对比图

综合以上图像对比分析可知：在图像上视觉直观感觉逆滤波、维纳滤波更能还原主

体，但整体清晰度较差，最小二乘滤波整体一般，但主体没有清晰化，综合图像还原的目的和视觉直观体验得，逆滤波、维纳滤波优于最小二乘滤波。

3 基于盲去卷积原理的动态模糊图像复原算法

3.1 匀速直线运动退化模型

动态模糊是由于拍摄物体在相机快门打开期间与相机的相对运动导致物体图像的平滑，减少动态模糊的方法有两种，第一种是减少曝光时间，但是曝光时间越短，采集到的图像信噪比也越低，且曝光时间减少到一定数值便很难再继续减小，这就限制了通过控制曝光时间来改善动态模糊的应用。第二种方法是通过图像恢复算法来进行模糊图像复原，这种方法局限性小，且效果明显，因此也成为改善动态模糊图像的主要途径。由于变速、曲线运动可看作是匀速直线运动的叠加，因此研究由匀速直线运动造成的图像模糊恢复更加具有普遍性和代表性。

在不考虑噪声的情况下，动态模糊模型如下图所示：

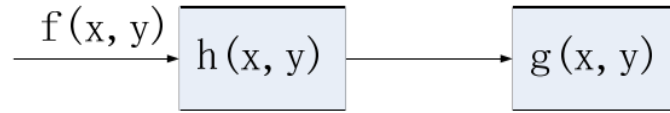


图 3-1 无噪声动态模糊模型

$f(x, y)$ 为原始图像， $h(x, y)$ 为退化函数， $g(x, y)$ 为模糊图像，图像在 x 和 y 方向上的位移分别为 $x_0(t)$ 和 $y_0(t)$ ，相机运动时间为 T ，则由相对运动产生的模糊图像可表示为：

$$g(x, y) = \int_0^T f[x - x_0(t), y - y_0(t)] dt \quad (3-1)$$

对上式进行傅里叶变换由傅里叶逆变换性质可推出：

$$G(u, v) = F(u, v) \int_0^T \exp[-j2\pi(ux_0(t) + vy_0(t))] dt \quad (3-2)$$

将 $H(u, v)$ 定义为：

$$H(u, v) = \int_0^T \exp[-j2\pi(ux_0(t) + vy_0(t))] dt \quad (3-3)$$

则上式可简化为：

$$G(u, v) = F(u, v) \cdot H(u, v) \quad (3-4)$$

其中 $H(u, v)$ 为动态模糊的点扩散函数。

若考虑的是水平方向的相对模糊运动，即只存在 x 方向运动， y 方向相对运动幅度为 0，则问题可简化为下式：

$$g(x, y) = \int_0^T f[x - x_0(t), y] dt \quad (3-5)$$

设在曝光时间 T 内水平方向移动的像素个数为 a ，则有水平匀速直线运动速度为 a/T ，在时刻 t 的水平位移 $x_0(t) = aT/t$ ，则上式可转换为：

$$g(x, y) = \int_0^T f(x - aT/t, y) dt \quad (3-6)$$

上面研究的是连续模式下的动态模糊，在离散模式下，上式可变为：

$$g(x, y) = \sum_{i=0}^{a-1} f(x-i, y) \Delta t \quad (3-7)$$

其中， a 为模糊运动距离对应的像素个数的整数近似， Δt 为每个像素对模糊产生影响的时间因子。从物理角度来看动态模糊，其实质是原始清晰图像在一段距离上的延迟并叠加的过程，叠加获得的图像就是动态模糊图像，则水平匀速动态模糊模型可由下式描述：

$$g(x, y) = \frac{1}{a} \sum_{i=0}^{a-1} f(x-i, y) \quad (3-8)$$

若用函数的卷积来描述，如下式所示：

$$g(x, y) = f(x, y) \cdot h(x, y) \quad (3-9)$$

$$h(x, y) = \begin{cases} 1/L, & 0 \leq x \leq L-1 \\ 0, & \text{其他} \end{cases} \quad (3-10)$$

其中， $h(x, y)$ 为退化函数， $f(x, y) \cdot h(x, y)$ 表示原始图像函数与退化函数的卷积运算。

3.2 模糊运动参数的确定

常用的模糊图像复原方法有很多，包括逆滤波、维纳滤波算法、盲解卷积算法、Lucy-Richardson 算法等，不同的算法效果和使用范围各不相同，但是都有一个共同点，那就是需要预先确定点扩散函数 PSF，在不知道点扩散函数的情况下，进一步的复原工作无法进行。而对于一般的模糊图像，都并没有直接给出点扩散函数，必须通过已有的模糊图像信息估计点扩散函数。

动态模糊图像复原中，点扩散函数的参数主要包括：模糊角度 θ 和模糊长度 L ，模糊角度 θ 表示图像进行模糊运动的方向，模糊长度 L 表示像素点在模糊运动中移动的相对长度。由于这两个参数都是未知的，我们必须通过对模糊图像的处理来推断出它们值的大小。

3.2.1 动态模糊角度的确定

动态模糊角度的确定对于整个点扩散函数的确定是十分重要的，确定了模糊角度之后就能够将非水平方向的模糊运动转化为水平方向的匀速直线运动，降低了动态模糊长度和图像复原的难度。

对于匀速直线动态模糊而言，其点扩散函数具有零点，这就导致模糊图像的频谱也具有零点，在相应的频率处，频谱上会出现一系列平行的暗纹。若动态模糊角度为 θ ，则模糊图像频谱图上 $\theta+90^\circ$ 方向会出现平行暗纹。因此，只需要鉴别出频谱图上暗纹的

方向，即可得到模糊模糊角度值。

3.2.2 Hough 变换

*Hough*变换提取直线是将直线上的点坐标变换到过点的直线系数域，利用共线和直线相交的关系，将提取直线的问题转化为计数，且这种算法最大的优点是受图像中直线的间断与噪声影响较小。

设平面上有直线： $y = ux + v$ (3-11)

直线斜率为 u ，截距为 v ，对于每一条直线都有且只有唯一的数组 (u, v) 与之对应，而对每一个数组 (u, v) 也有唯一的直线 $y = ux + v$ 与之对应，*Hough*变换的原理正是基于这种 O_{xy} 平面上的直线与 O_{uv} 平面上的点一一对应的关系，同理， O_{uv} 平面上的一条直线与 O_{xy} 平面上的点也是一一对应的。

因此，对于 O_{xy} 平面上待检测的直线 $y = ux + v$ ，由于该直线上的每一个点都对应着 O_{uv} 平面上的一条直线，所有的这些直线必会相交于点 (u, v) ，只需要检测出这个相交点就能够获得关于待测直线的相关信息。

由于 O_{xy} 平面上存在着垂直与 x 轴的直线，这样的直线斜率为无穷大，无法用 $y = ux + v$ 的方式表达，因此常用直线的法线式表示：

$$x \cos \theta + y \sin \theta = \rho \quad (3-12)$$

式中， ρ 为原点到直线的垂直距离， θ 为 x 轴与直线法线的夹角， O_{xy} 平面中一条直线对应于 $O_{\theta\rho}$ 平面中一点， O_{xy} 平面中一点对应于 $O_{\theta\rho}$ 平面中的一条曲线，则 O_{xy} 平面中一条直线上的所有点对应的 $O_{\theta\rho}$ 平面中的曲线相交于同一点，因此只需检测到 $O_{\theta\rho}$ 平面各点信息，就可以确定 O_{xy} 平面中待测直线的位置，若对 $O_{\theta\rho}$ 平面上相交点处曲线相交次数进行统计，其统计值等于经过该点的曲线条数，也就等于 O_{xy} 平面中待测直线上像素点的个数，这样一来，每一个统计值的峰值就对应 O_{xy} 平面上的一条直线，这种检测 O_{xy} 平面上直线的方法就是*Hough*变换直线检测法。

3.2.3 Sobel 边缘检测算子

边缘检测是图像处理中的一个重要的应用方面，检测出边缘才能够准确的识别出目标，在图像处理问题中，边界代表一个区域与另一区域的临界点的集合，不同区域的特征和属性不相同，边缘提取技术正是基于这种图像背景与景物在数字图像特征上的差异来进行的，这些特征主要有灰度和纹理等，边缘检测其实就是确定图像特征变化的像素点位置。

边缘检测算子是一种简易可行的边缘提取方法，通过考察图像中各像素点在邻域内灰度变化情况，计算其一阶或二阶方向导数值来确定边缘点。若某像素点位于边界处，则其相邻像素的灰度值变化就较大，对这种灰度值的变化进行量化统计就能够提取出图像的边界，常用的边缘检测算子有 Robert 算子、Sobel 算子、Canny 算子等，在这里我们使用 Sobel 算子进行边缘提取。

Sobel 算子从不同的方向对图像进行边缘检测，其本身是一组方向算子的集合，它提高了被考察像素点上下左右六个像素点的权重，其输出为原图像的边缘图像。该算子表达式如下：

$$f'_x(x, y) = f(x-1, y+1) + 2f(x, y+1) - f(x-1, y-1) - 2f(x, y-1) - f(x+1, y-1) \quad (3-13)$$

$$f'_y(x, y) = f(x-1, y-1) + 2f(x-1, y) + f(x-1, y+1) - 2f(x, y-1) - f(x+1, y-1) \quad (3-14)$$

$$G[f(x, y)] = |f'_x(x, y)| + |f'_y(x, y)| \quad (3-15)$$

式中 $f'_x(x, y)$ 、 $f'_y(x, y)$ 分别表示 x 、 y 方向的一阶导数， $G[f(x, y)]$ 是 Sobel 算子的梯度值， $f(x, y)$ 代表输入图像，在计算出像素点的梯度之后，若其值大于事先设定的阈值 T ，则标记此点为边界点。

3.2.4 模糊模糊角度检测实验及结果

试验流程图如下：

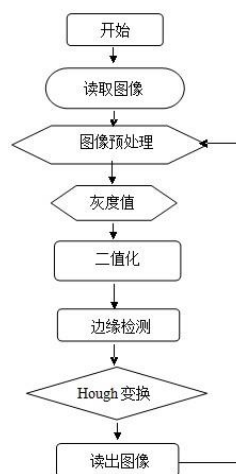


图 3-2 流程图

首先研究下面的动态模糊图像。在假定未知模糊角度大小的前提下，通过实验来确

定其模糊角度值，并与真实值进行比较。然后对模糊图像进行傅里叶变换，得到频谱图如下：



图 3-3 动态模糊图像

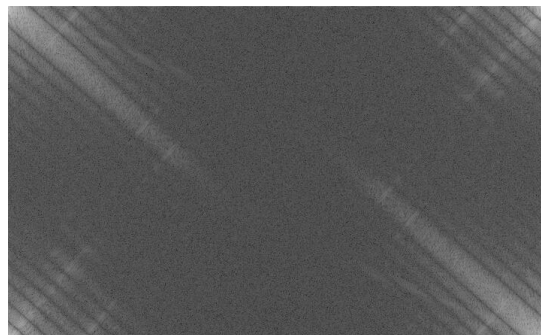


图 3-4 模糊图像频谱图

可以看出，频谱图中有一系列平行暗纹，只需确定暗纹方向即可得到动态模糊角度，为使图中条纹更加清晰，先将灰度图像转化为二值图像。由于需要检测的直线是暗条纹，为方便起见，先将二值图像反转，将待测直线变为亮条纹。

使用变换对上图进行直线方向确定，Hough 变换检测图像中亮度为 255 的像素点，将每一个这样的点转换为另一个域中的一条曲线，共线的所有像素点对应的变换域中的曲线交于一点，只需要在变换域中确定出相交次数最多的点，就能获得原图像中对应直线的信息。

从图中可以看到，中间部分存在大片的白色亮区，如果直接将整幅图像用来做 Hough 变换，则中间的亮区将会影响边缘亮线角度的检测，使结果误差增大。因此，只截取图像边角上线条明显的区域进行处理，由于图中所有的直线都是平行的，这样做并不会引入误差，下图是截取左上角的图像：

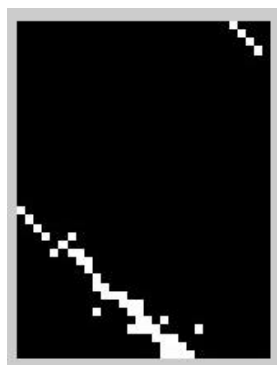


图 3-5 截取图像

对上图进行 Hough 变换，检测出变换域中各点叠加次数的峰值，得到峰值处对应的直线角度，即可计算出模糊模糊角度大小。

Matlab 运行结果：对应的动态模糊角度 $\theta = 88^\circ$ ，误差为 1° 。由此可以看出用此方法对模糊角度进行估计，结果比较准确，但是不是对于任意角度的模糊图像都可以得到

比较精确的结果。

为了验证此方法是否具有普遍适用性,按照上述方式对不同动态模糊角度的图像进行角度的确定,实验结果如下表:

表 3-1 模糊角度估计实验结果

真实模糊角度 (度)	50	60	70	80	90	100
估计模糊角度 (度)	50	59	70	90	99	101
误差 (度)	0	1	0	0	1	-1

从实验结果可以看出,本算法的准确度较高,误差很小。

3.3 动态模糊长度的确定

对于一幅动态模糊图像,其频谱图上一系列暗线的个数即是原图像模糊运动的长度,可以通过统计频谱图上暗条纹个数来获得模糊长度 L 的值。在前面已经估计出了模糊模糊角度值,可以通过将图像往相反的方向旋转相同角度值,将任意方向的动态模糊转化为水平方向,这样,问题就转化为模糊角度为零的动态模糊长度的确定。

下面考察一幅动态模糊图像:



图 3-6 动态模糊图像

其频谱图如下:

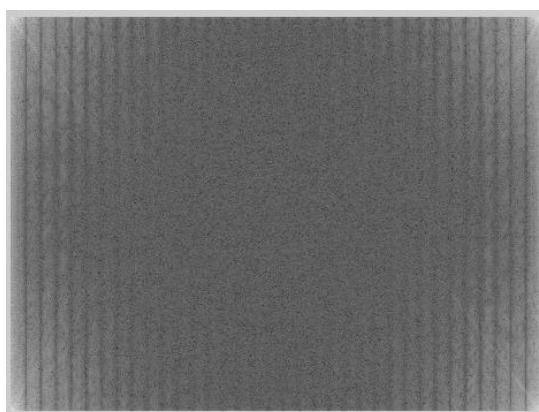


图 3-7 频谱图

为统计上图中暗纹条数,将频谱图视为二维矩阵,求矩阵各列的和,得到一个行向量,该向量的每个元素值代表原频谱矩阵中各列元素之和,暗纹条数即对应于这个向量中极小值的个数。用曲线表示该向量如下:

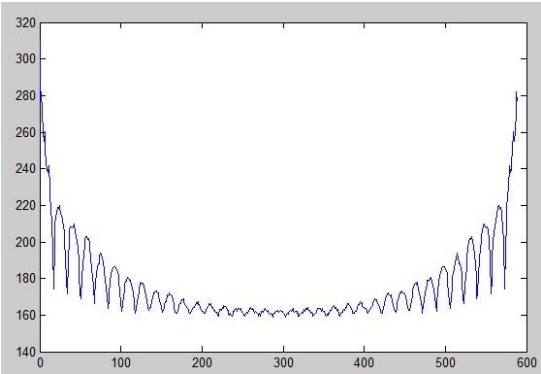


图 3-8 暗条纹统计曲线

从上面的曲线可以看出，极小值的个数一共为 34，则对原图像的模糊长度估计为 34。按此方法进行多组试验以便得到准确的结论，实验结果如下表：

表 3-2 暗条纹统计法模糊长度实验结果

实际模糊长度（像素）	6	22	45	80	110	125	168	180	200
估计模糊长度(像素)	5	21	44	78	107	123	166	176	195
误差值(像素)	1	1	1	2	3	2	2	4	5

上面的方法在模糊模糊长度不是很大时效果很好，而在模糊长度较大时，由于暗纹统计结果图中条纹个数太多，这将使极小值的统计难度增大，当条纹数足够多时，有可能无法准确的统计出暗纹条数，因此，这种模糊模糊长度统计的方法仅适用于模糊长度较小的情况。模糊长度较大的情况在此不作研究。

3.4 题目的求解

应用本文上述方法，计算出来原图动态模糊角度 88 度，模糊长度 31 像素。通过上述流程处理本题所给模糊图像可得出以下结果。

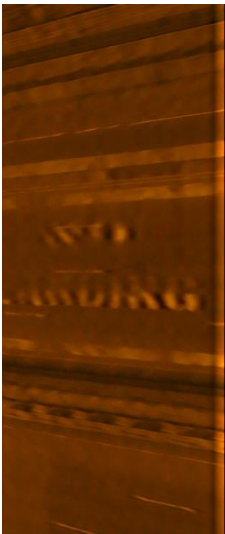


图 3-9 还原处理

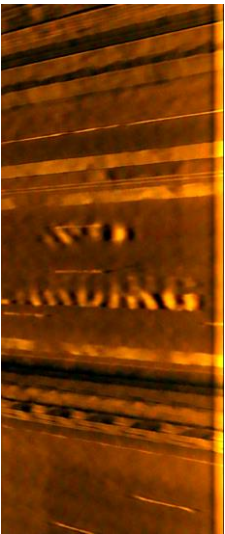


图 3-10 USM 锐化

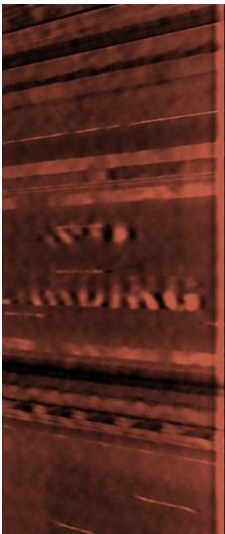


图 3-11 色相饱和度还原

由上图可以得出该图清晰度已达标准，其中图中可见字母为“ENDING”。

4 复原方法性能测试与评价

4.1 算法总结

本文通过对三种滤波复原图像方法的研究,提出基于盲去卷积原理的动态模糊图像复原算法。其中以图 2-1 为例,求解出三种方法复原程度。其中,逆滤波复原函数值为 4.82×10^7 , 维纳滤波复原函数值为 4.76×10^7 , 最小二乘吕伯父元函数值为 5.50×10^5 。

综合以上数据及图像对比分析可知:在图像上视觉直观感觉逆滤波、维纳滤波更能还原主体,但整体清晰度较差,最小二乘滤波整体一般,但主体没有清晰化,综合图像还原的目的和视觉直观体验得,逆滤波、维纳滤波优于最小二乘滤波。

其次通过研究盲去卷积原理,确定了所需的两个参数——运动模糊尺度和运动模糊角度的计算方法。运动模糊角度是通过计算模糊图像频谱图条纹方向来确定的,在频域中截取条纹明显的边缘区域,然后使用 Sobel 算子进行边缘提取,再进行 Hough 变换得到模糊运动方向;运动模糊长度的确定主要介绍了频谱图暗条纹统计法,这种方法适用于运动尺度较小的情况。

最终对本题所给动态模糊图像给出较好复原。

4.2 算法展望

(1) 本文研究的是匀速直线运动模糊图像的复原,而实际中存在这变速、曲线的运动模糊,这些条件下的图像复原更为复杂,对这些情况本文缺乏研究。

(2) 在使用维纳滤波法进行图像复原时,复原图像中或多或少的存在着振铃现象,估计参数与真实参数差距越大,振铃效果越明显,因此还需要进一步研究如何优化算法,减小振铃效果。

(3) 本文对图像的评价采取主观评价法,严格意义上讲应该采取主客观相结合的方式对图像质量进行评价,才具有一定的说服力。一次还需要进一步改善图像质量评价方式,使之更严谨,提高说服力。

5 参考文献

- [1] 李桐. 单幅图像全局运动去模糊研究[D]. 北京交通大学, 2016.
- [2] 寇杰. 图像的运动模糊复原算法研究[D]. 中国科学院研究生院(光电技术研究所), 2016.
- [3] 刁兆静. 基于变分方法的单幅运动模糊图像盲复原[D]. 青岛大学, 2016.
- [4] 刘洋. 基于运动模糊图像的超分辨率复原方法研究[D]. 哈尔滨工业大学, 2009.
- [5] 彭青建. 运动模糊图像复原算法研究[D]. 电子科技大学, 2010.
- [6] 李祚林, 李晓辉, 马灵玲, 胡玥, 唐伶俐. 面向无参考图像的清晰度评价方法研究[J]. 遥感技术与应用, 2011, 26(02): 239-246.
- [7] 曹慧. 基于单幅运动模糊图像复原算法研究[D]. 西安理工大学, 2017.
- [8] 徐贵力, 刘小霞, 田裕鹏, 程月华, 李鹏. 一种图像清晰度评价方法[J]. 红外与激光工程, 2009, 38(01): 180-184.
- [9] 赵雪青. 降质图像复原方法研究[D]. 陕西师范大学, 2013.

- [10]任金凡. 运动模糊图像复原算法的研究[D]. 电子科技大学, 2014.
- [11]石明珠. 运动模糊图像全变分复原理论及关键技术研究[D]. 北京理工大学, 2014.
- [12]赵阳. 运动图像去模糊技术研究[D]. 西安电子科技大学, 2014.
- [13]李红阳. 运动图像恢复[D]. 清华大学, 2002.

6 附录

6.1 MATLAB 代码汇总

6.1.1 图像复原程度评价标准—Brenner 梯度函数代码

```
function D=Brenner_algorithm(f, T)
% Brenner_algorithm
%
if nargin < 2
    T = 0;
end
[m,n] = size(f);
tmp1=abs(f(3:m,:) - f(1:m-2,:));
tmp1 = tmp1(:);
D = sum(tmp1(abs(tmp1)>T));
end
```

6.1.2 运动模糊方向长度检测的代码

```
close all;
clear all;

%% 读入并显示图像
filename = 'E:\pic\plane.png';
I = imread(filename);

%figure
%imshow(uint8(I));
%title('原图');

%% 生成运动模糊图像
PSF = fspecial('motion',30, 35);
g = imfilter(I, PSF, 'circular');
%figure
%imshow(uint8(g));
%imwrite(g,'plane3035.png')
g=imread('E:\pic\road.png');
```

```
%title('运动模糊图');

%% 对运动模糊图像进行灰度化，并进行二维快速傅里叶变换，生成其频谱图
gb = rgb2gray(g);
%figure
%imshow(uint8(gb));
PQ = paddedsize(size(gb));
F = fft2(gb, PQ(1), PQ(2));
%figure
%imshow(uint8(F));

%% 作出倒频谱
F1 = log(1+abs(F));
F2 = abs(F1).^2;
F3 = real(ifft2(F2));
%figure
%imshow(uint8(F3));

%% 将倒频谱压缩，居中
H = log(1+abs(F3)); % 将倒频谱动态范围进行压缩
Hc = fftshift(H); % 将压缩结果进行循环移位，使低频成分居中
%figure
%imshow(uint8(Hc));

%% 通过阈值处理，边缘检测“canny”算子二值化倒频谱
T = graythresh(Hc);
bw=edge(Hc, 'canny', T);
%figure
%imshow(bw);

%% 对倒频谱从 1 到 180 作 radon 变换，以求出模糊角度
theta = 1:180;
R = radon(bw, theta);
%figure
%imshow(R);

%% 计算出通过倒频谱 radon 变换估计出的模糊角度
MAX = max(max(R));
[m, n] = find(R == MAX);
if 90 < n <= 180
    beita = n - 90;
elseif 0 < n < 90
    beita = n + 90;
```

```
elseif n == [90;90] | n == [180;180]
    beita = n(1);
end;
beita
```

6.1.3 频谱图代码

```
X=imread('E:\pic\plane3035.png');
X=rgb2gray(X);
subplot(221);
imshow(X);
title('原始图像');
% 生成含噪图像并图示

%用小波函数 coif2 对图像 XX 进行 2 层
% 分解
[c,l]=wavedec2(X,2,'coif2');
% 设置尺度向量
n=[1,2];
% 设置阈值向量 , 对高频小波系数进行阈值处理
p=[10.28,24.08];
nc=wthcoef2('h',c,l,n,p,'s');
% 图像的二维小波重构
X1=waverec2(nc,l,'coif2');
subplot(223);
imshow(uint8(X1));
%colormap(map);
title(' 第一次消噪后的图像 ');
%再次对高频小波系数进行阈值处理
mc=wthcoef2('v',nc,l,n,p,'s');
% 图像的二维小波重构
X2=waverec2(mc,l,'coif2');
subplot(224);
imshow(uint8(X2));
title(' 第二次消噪后的图像 ');

X=rgb2gray(X);
B=X;
B1=fft2(double(B));
B2=mat2gray(log(abs(B1)));
figure,imshow(B2);%模糊图像的频谱图
C=sum(B2,1);%对频谱图求列和
[m,n]=size(C);
figure,plot(C);%绘制频谱列和曲线
```

6.1.4 三种滤波复原总代码

```

I=imread('E:\pic\plane3035.png');
figure(1)
imshow(I,[])
title('原图像');
PSF=fspecial('motion',30,35);%生成滤波掩模 fspecial('motion',len,theta)len 个像素的
图像线性运动，运动方向为 theta
MF=imfilter(I,PSF,'symmetric');
noise=imnoise(zeros(size(I)),'gaussian',0,0.000001);% 添 加 噪 声
imnoise(f,'gaussian',m,var) 均值 m，方差 var 的噪声添加到图像上
MFN=imadd(I,im2uint8(noise));
NSR=sum(noise(:).^2)/sum(MFN(:).^2);
figure(3);
imshow(deconvwnr(MFN,PSF),[]);
title('逆滤波复原');
figure(4);
imshow(deconvwnr(MFN,PSF,NSR),[]);
title('维纳滤波复原');
NP=0.002*prod(size(I));
[reg1 LAGRA]=deconvreg(MFN,PSF,NP/3.0);
figure(5);imshow(reg1);
title('最小二乘滤波复原');

```

6.1.5 盲去卷积代码

```

clear
I=imread('E:\pic\plane3035.png');
I=im2double(I);
%subplot(121)
%imshow(I)
%subplot(122)
PSF=fspecial('motion',30,35);
SD=0.0001;
g=imnoise(I,'gaussian',0,SD^2);
DAMPAR=sqrt(SD);
LIM=ceil(size(PSF,1)/2);
WEIGHT=zeros(size(g));
WEIGHT(LIM+1:end-LIM,LIM+1:end-LIM)=1;
NUMIT=50;
f5=deconvlucy(g,PSF,NUMIT,DAMPAR,WEIGHT);
imshow(f5)

```