

多帧图像超分辨率修复

华诚 卢红旺 华淼

September 18, 2011

Abstract

本文针对超分辨率多帧图像修复理论进行总结说明。本文主要针对两方面问题进行说明：

1. 多帧数字图像超分辨率数据融合过程。该过程主要是确定其他帧与参考帧之间的相对运动，估算出相对运动后将各个像素映射到高分辨率图像的网格中。即完成数据融合过程。
2. 去模糊过程。模糊可能有多方面的因素造成，物理因素：相机运动，离焦；数学因素：数据插值造成类高斯扩散过程。

多帧数据融合与去模糊过程构成超分辨率思想的主体。文章主要将会以此两部分展开。文章组织如下：**第一部分**，介绍超分辨率重构的基础理论；**第二部分**，我们目前实现的多帧数据融合理论；**第三部分**，我们目前实现的去模糊算法；**第四部分**，总结及相关理论发展方向。

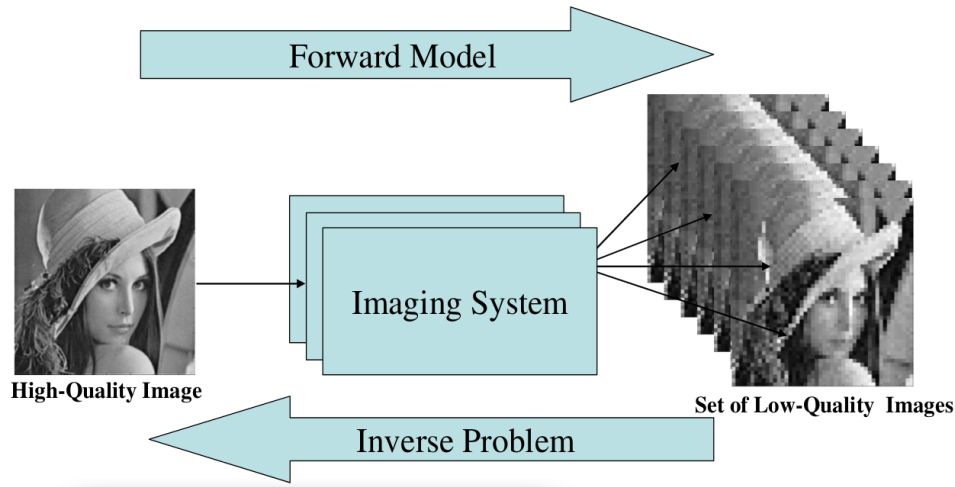


Figure 1: 超分辨率重构过程数学描述

1 超分辨率(Super-Resolution)重构基础理论

1.1 分辨率的概念

超分辨率(SR)之所以称之为超分辨率(SR)是相对于一张低分辨率(LR)的图像而言的，假设LR的分辨率是 $M \times N$ ，放大系数为2，则对应的高分辨率(HR)图像点阵数量应该是 $2M \times 2N$ 。但是点阵密度的增加并不能等同于分辨率的提升。以线性插值法为例，在2倍的放大过程中，一个像素要被扩展成四个像素，若是使用相邻两个元素来重构中间缺失的像素。实际是LR图中 (i, j) 像素信息向 $(2i - 1, 2j - 1)$, $(2i - 1, 2j)$, $(2i - 1, 2j + 1)$, $(2i, 2j - 1)$, $(2i, 2j + 1)$, $(2i + 1, 2j - 1)$, $(2i + 1, 2j)$, $(2i + 1, 2j + 1)$ 。这样的信息各向同性的扩散会造成模糊，进而使放大后的HR图像细节上反而没有LR丰富。因为线性插值算法是类高斯的模糊过程。SR过程的目的是在增加点阵数量的同时也增加细节。分辨率实际表征的是细节丰富的程度。

1.2 SR问题的基本思想

如图fig1之中，高清的图像经成像系统处理生成一系列的低分辨率的图像过程称为向前模型，也叫正问题，即已知输入求输出。反之，由输出求输入，即是反问题。不论是求解反问题还是正问题最核心的问题还是确定成像系统的模型。成像系统模型确定后，需要进行的工作就是模型参数估计。

1.2.1 超分辨率重构过程数学模型

如图fig2所示，真实世界的高分辨率场景 $X(x, y)$ 经过大气模糊 H_{atm} ，像素相对运动 F ，相机运动模糊 H_{cam} ，下采样 \downarrow 及加性噪声 V ，最终生成含噪模糊下采样图像 $Y[1]$ 。

$$Y(m, n) = [H_{cam} \otimes F(H_{atm} \otimes X)] \downarrow + V(m, n) \quad (1)$$

其中， \otimes 表示卷积运算， H_{atm} 和 H_{cam} 形成模糊的去卷积核，大小均为 $p \times p$ ，此处均以 X, Y, V 二维形式表示。 Y 的阶数： $M \times N$ ， X 的阶数： $rM \times rN$ ， r 分辨率图像相对低分辨率图像的尺度大小。记 X^\dagger 为 X 首尾相接的一维形式， $r^2MN \times 1$ 。 \downarrow 记做 D 。

$$Y_k^\dagger = D_k H_k^{cam} F_k H_k^{atm} X^\dagger + V_k \quad k = 1 \dots N \quad (2)$$

k 表示同一个场景图像序列中的第 k 帧， X^\dagger 为 $r^2MN \times 1$ 的列矢量， H_k^{atm} 和 H_k^{cam} 分别是相应模糊核 H_{atm} 和 H_{cam} 的Toeplitz形式矩阵， $r^2MN \times r^2MN$ ， F_k 也是 $r^2MN \times r^2MN$ 。 D_k 是 $MN \times r^2MN$ 的形式。

调换模糊顺序，则有

$$\begin{aligned} Y_k^\dagger &= D_k H_k^{cam} F_k H_k^{atm} X^\dagger + V_k^\dagger \\ &= D_k F_k H_k^{cam} H_k^{atm} X^\dagger + V_k^\dagger \end{aligned} \quad (3)$$

$$= D_k H_k^{cam} H_k^{atm} F_k X^\dagger + V_k^\dagger \quad (4)$$

$$k = 1 \dots N$$

此处我们使用(3)式模型来计算。同时令 $H_k = H_k^{cam} H_k^{atm}$ ，则有

$$Y_k^\dagger = D_k F_k H_k X^\dagger + V_k^\dagger \quad k = 1 \dots N \quad (5)$$

此处再令 $Z_k^\dagger = H_k X^\dagger$

$$Y_k^\dagger = D_k F_k Z_k^\dagger + V_k^\dagger \quad k = 1 \dots N \quad (6)$$

$$Z_k^\dagger = H_k X^\dagger \quad (7)$$

如此由一系列相同场景的低分辨率图像 Y_k 重构出该场景的高分辨率图像 X 的过程就可以分为两步：

1. 借助像素运动估计，计算出 X 的模糊版本 Z ，对应(6)式表示过程，称为数据融合过程；
2. 由 Z 借助盲去模糊，计算出 X ，对应(7)式表示过程，称为去模糊过程。

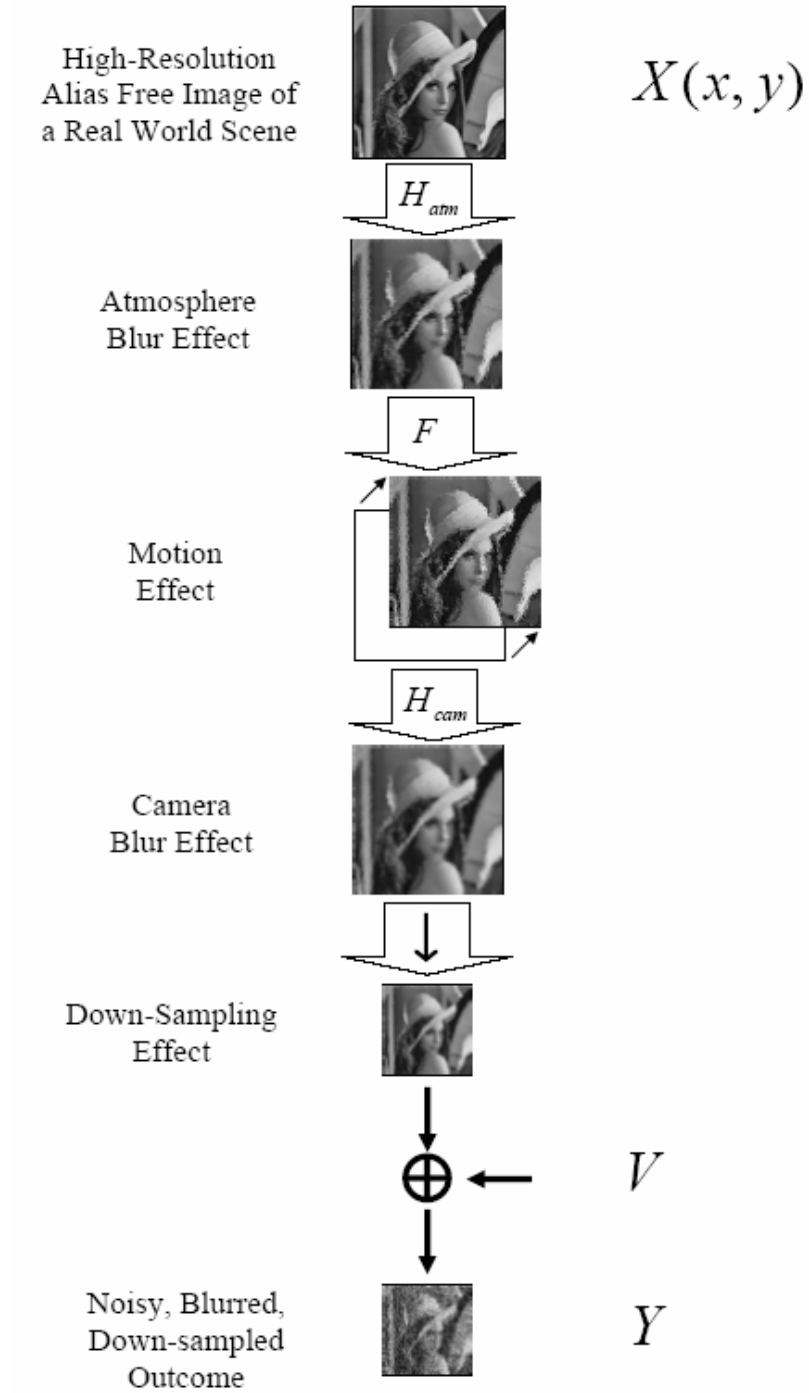


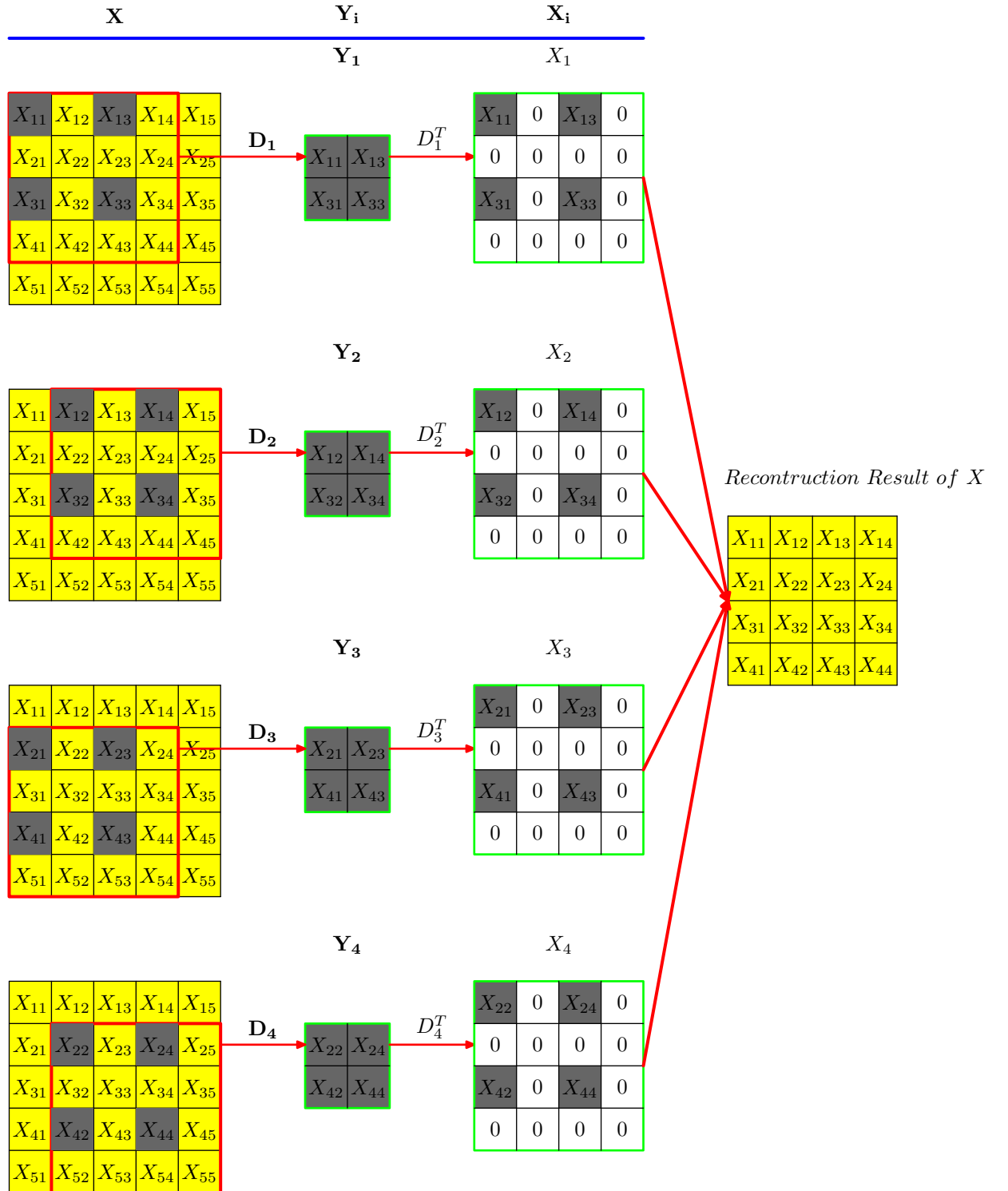
Figure 2: 超分辨率重构过程数学模型

目前文献[1, 2, 3, 4, 5, 6]中，都是利用(3)式的结果进行处理，但是我想用(4)式处理应该会有优势。因为在反问题计算过程中，若利用(3)式，则是先进行运动估计，后进行去模糊；因为模糊的存在造成运动估计过程无法对于局部细节的运动进行估计，而先去模糊则可以，恢复细节再进行像素运动估计。上述利用(4)式来计算的理由也不充分，因为低分辨率的图像 Y_k 是模糊的，若是先对参考帧插值放大后就去模糊，而其他帧不去模糊，运动估计也还是不准确，所以如此看来，对于其他帧 $Y_j, (j \neq k)$ 也应该进行去模糊操作。利用(4)式计算是不是好，我也吃不准，我想可以试一试。

目前仍按(3)式的思想来处理。

1.2.2 数据融合过程基本思想示例

如下图所示，用 5×5 的矩阵表示某个场景 S 的高清图像 X ，该场景的低分辨率图像序列为 Y_1, Y_2, Y_3, Y_4 。图中红色边框表示相机捕获有效场景的区域。该场景经过下采样过程(D_i 表示)，生成一系列的低分辨率图像序列。红色边框所捕获场景不同内容的原因，可以是相机不动，场景发生运动造成的结果；也可以是场景不动，相机发生小幅运动。总之，是相机与场景发生小幅的相对运动。注意：这种相机与场景的相对运动不会造成 Y_i 运动模糊，理由是这种相对运动是亚像素级的，造成 Y_i 模糊的运动必须是像素级的。以 Y_1 作为参考帧 F_{ref} ，将其经上采样映射到 X_i 网格上去。从图中我们可以看出， Y_2 图像是相机相对场景向右移动1像素，经 $1/2$ 下采样 D_2 的结果，即 Y_2 相对 Y_1 向右移动了 $1/2$ 像素。同理 Y_3 相对 Y_1 向下移动 $1/2$ 像素； Y_4 则是相对 Y_1 向下移动 $1/2$ 像素，同时再向右移动 $1/2$ 像素。在估计出低分辨率图像 Y_k 之间的相对运动，我们也就能计算出其上采样网格 X_k 之间的相对运动。由此重构出一副高分辨率的图像 X 。同时在此例子中我们也看到，对于场景的运动估计(场景相对相机运动)，必须非常准确，对于低分辨率图像序列要达到亚像素级别。若是运动估计错误，重构的图像可能还没有还没有低分辨率图像好。



在实际问题之中，场景的运动可能不是均一的。以一段演讲视频为例，其场景的运动就是不均匀的，通常演讲者嘴巴运动明显，而鼻子则运动的较小。所以仅有全局的场景运动估计还不够，还要有局部场景运动估计。高精度的局部场景运动估计→称为光流(Optical Flow)，但是其计算非常的复杂，具体可以参看M.D.Robinson的博士论文[7]。fig3是其计算的一个局部运动估计结果。

2 实际场景中的数据融合

我们现在所使用的针对实际场景的运动估计来源于Elad[8]的文章。Elad在文章中提出了复杂的局部场景运动是由简单运动线性叠加而成，在此架构下Elad提出了概率运动估计模型。由

$$Y_k^\dagger = D_k F_k Z_k^\dagger + V_k^\dagger \quad k = 1 \dots N \quad (8)$$

$$Z_k^\dagger = H_k X^\dagger \quad (9)$$

利用Bayes定理则有

$$E(X^\dagger) = \frac{1}{2} \sum_{t=1}^N \|D_k F_k H_k X^\dagger - Y_k^\dagger\|_2^2 + \lambda R(X^\dagger) \quad (10)$$

$$E(Z^\dagger) = \frac{1}{2} \sum_{t=1}^N \|D_k F_k Z_k^\dagger - Y_k^\dagger\|_2^2 \quad (11)$$

$$E(X^\dagger) = \frac{1}{2} \sum_{t=1}^N \|H_k X^\dagger - Z_k^\dagger\|_2^2 + \lambda R(X^\dagger) \quad (12)$$

此处引入一个假设，假设 H_k 是相同的，这个假设的基础是因为 N 帧图像序列，在捕获时时间非常短，一般 $\leq 0.5s$ (以一秒三十帧， N 通常不大于15计)。在这么短的时间内，假设其相机运动状态和大气运动状态保持不变，是一个合理的假设，所以此处有 $Z^\dagger = H X^\dagger$ 。同时从上面数据融合的示例之中，可以看出 D_k 彼此也是相等的。综上，

$$Y_k^\dagger = D F_k Z^\dagger + V_k^\dagger \quad (13)$$

所以，(11)(12)式可以简化成

$$E(Z^\dagger) = \frac{1}{2} \sum_{k=1}^N \|D F_k Z^\dagger - Y_k^\dagger\|_2^2 \quad (14)$$

$$E(X^\dagger) = \frac{1}{2} \|H X^\dagger - Z^\dagger\|_2^2 + \lambda R(X^\dagger) \quad (15)$$

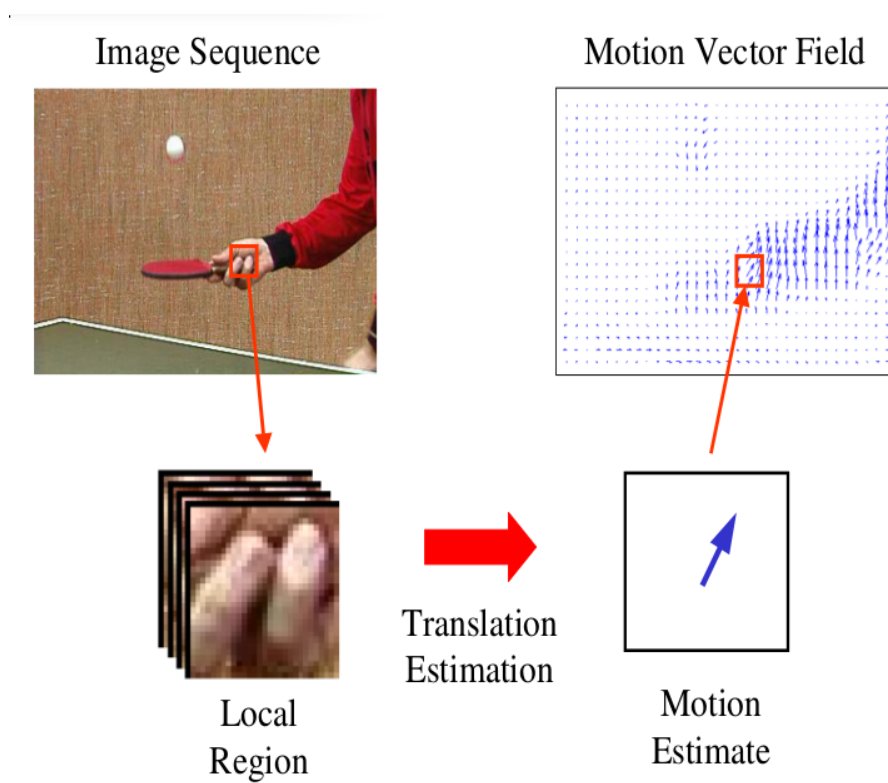


Figure 3: 局部场景运动估计[7]

此处首先针对(14)式处理，对于 $F_k Z^\dagger$ ， F_k 表示灰度级空间全局运动算符，实际是使灰度级在空间发生运动。假设 $F_k = \sum_{m=1}^M Q_{m,k} F_m$ ， F_m 也是全局运动算符。此式表达的是一个复杂的灰度级空间运动是有 M 个简单灰度级空间运动按系数 $Q_{m,k}$ 线性叠加而成。若位于空间位置 (i,j) 的灰度级 $X_{i,j}$ 在水平方向上空间移动范围是 $(i - \Delta M, i + \Delta M)$ ，垂直方向空间移动范围是 $(j - \Delta M, j + \Delta M)$ ，由此 $M = (2 * \Delta M + 1)^2$ 种简单运动形式合成复杂灰度级空间运动。

所以，

$$F_k Z^\dagger = \sum_{m=1}^M Q_{m,k} F_m Z^\dagger \quad (16)$$

$$E(Z^\dagger) = \frac{1}{2} \sum_{k=1}^N \sum_{m=1}^M \|DF_m Z^\dagger - Y_k^\dagger\|_{W_{m,k}}^2 \quad (17)$$

对于(16)(17)式的推导，我目前还未想明白，存有疑问？

其中， $\|a\|_W^2 = a^T W a$ ，求出使 $E(Z^\dagger)$ 最小的 Z^\dagger ，即求出 Z^\dagger 。对 $E(Z^\dagger)$ 相对 Z^\dagger 求一阶导数。

$$\frac{\partial E(Z^\dagger)}{\partial Z^\dagger} = \sum_{k=1}^N \sum_{m=1}^M F_m^T D^T W_{m,k} (DF_m Z^\dagger - Y_k^\dagger) \quad (18)$$

令 $\frac{\partial E(Z^\dagger)}{\partial Z^\dagger} = 0$ ，同时令

$$W_m^* = \sum_{k=1}^N W_{m,k} \quad (19)$$

$$Y_m^{\dagger*} = \sum_{k=1}^N W_{m,k} Y_k^\dagger \quad (20)$$

所以有

$$\left\{ \sum_{m=1}^M F_m^T D^T W_m^* DF_m \right\} Z^\dagger = \sum_{m=1}^M F_m^T D^T Y_m^{\dagger*} \quad (21)$$

先分析等号右侧表达式 $RightSide(RS) = \sum_{m=1}^M F_m^T D^T Y_m^{\dagger*}$ ： F_m 的作用是使 (i,j) 位置灰度级移动 $(dx(m), dy(m))$ 到 $(i + dx(m), j + dy(m))$ 空间位置。 F_m^T 的作用就是移动 $(-dx(m), -dy(m))$ 。 D^T 的作用选择部分位置的灰度级映射到上采样网格上。所以 F_m^T, D^T 两个算符作用到 $Y_m^{\dagger*}$ 上不改变灰度级的值，而仅改变其空间位置。

所以，

$$RS(i, j) = \sum_{(p,q) \in N(i,j)} Y_m^{\dagger*}(p, q) \quad (22)$$

其中， $N(i, j)$ 表示 (i, j) 的近邻，实际表示着 (i, j) 位置的灰度级经过下采样之后在低分辨率图像区域空间运动的范围。定义如下：

$$N(i, j) = \{(p, q) | \forall m \in [1, M], s * p = i + dx(m), s * q = j + dy(m)\} \quad (23)$$

s ：表示放大的系数。 M ：表示 M 种可能的运动。以 $s = 2, i, j$ 均为偶数为例， M 种运动中当且仅当 $dx(m), dy(m)$ 都是偶数时，才有整数的 p, q 出现，所以 $N(i, j)$ 实际表示着 M 种运动之中的有效运动。

再看等式左侧表达式，同样根据上面 $RS(i, j)$ 分析的思想，

$$LS(i, j) = \sum_{m=1}^M W_m^* Z(i, j) \quad (24)$$

$$= \sum_{(p, q) \in N(i, j)} W^*(p, q, i, j) Z(i, j) \quad (25)$$

$W_{p, q, i, j}^*$ 实际表示着从低分辨率图像的 (p, q) 位置的灰度级 $Y(p, q)$ 经过上采样过程运动到 (i, j) 的概率。**注意：此时 $Z(i, j)$ 是二维形式，而不是一维形式 $Z^\dagger(i)$**

此即是Elad提出的概率运动模型。

所以，有 $Z^\dagger(i, j)$

$$Z(i, j) = \frac{\sum_{(p, q) \in N(i, j)} Y_m^*(p, q)}{\sum_{(p, q) \in N(i, j)} W^*(p, q, i, j)} \quad (26)$$

$$= \frac{\sum_{(p, q) \in N(i, j)} \sum_{k=1}^N W_k(p, q, i, j) Y_k(p, q)}{\sum_{(p, q) \in N(i, j)} \sum_{k=1}^N W_k(p, q, i, j)} \quad (27)$$

其中，对于 $W_k(p, q, i, j)$ 的表达式并没有什么明确的要求。 $W_k(p, q, i, j)$ 一般利用Bilateral Filter 和Non-Local Means Filter之中确定权重的方法进行。

在Elad[8]的文章之中，其使用的形式是

$$W_k(p, q, i, j) = \exp\left(\frac{-\|DF_m R_{i, j} Z^\dagger(i, j) - R_{p, q} Y(p, q)\|_2^2}{2\sigma^2}\right) f(p, q, i, j, k) \quad (28)$$

$R_{i, j}, R_{p, q}$ 表示以 $(i, j), (p, q)$ 为中心，选定的一个窗口 p ， p 的大小可以是 $1 \times 1, 3 \times 3, 5 \times 5$ 等等。 p 实际表示的是一个局部场景。实际计算实验表明， p 的大小选择 1×1 最好，此是最小场景，最容易找到相似场景。Elad采用此形式的权重或是概率形式，实际是认为灰度级的分布是正态分布形式。也可选择灰度级的分布是指数分布。 $f(p, q, i, j, k)$ 是其空间距离，对于权重的影响，一般不考虑。即可以令 $f(p, q, i, j, k)$ 为常数。

p 选择 1×1 时，上式可以简化成

$$W_k(p, q, i, j) = \exp\left(\frac{-\|Z^\dagger(i, j) - Y(p, q)\|_2^2}{2\sigma^2}\right) f(p, q, i, j, k) \quad (29)$$

此处我借用了量子力学之中波函数的概念，定义波函数 ψ ，将波函数的平方定义成权重， p 的尺寸定义为 1×1

$$\psi(p, q, i, j) = \exp\left\{\frac{\sqrt{-1} \times [Z(i, j) - Y(p, q)]}{\sigma}\right\} \quad (30)$$

$$W_k(p, q, i, j) = \psi^2 \times f(p, q, i, j, k) \quad (31)$$

针对数据融合过程，我想有以下方面可以改进的地方。

1. 对于 ψ 我目前使用的是 $\exp\left\{\frac{-[Z(i, j) - Y(p, q)]}{\sigma}\right\}$;
2. 目前在权重计算过程中计算量还非常大，除了现在已经实现查表法之外，我想目前C语言程序之中是RGB三通道分别计算，将RGB转成YUV，其中Y通道还是用数据融合思想处理。而颜色通道(UV)可以先试试直接使用WAZ插值看看结果。同时还可以试试Liu[9]提出的彩色化的方法;
3. 将多帧尝试使用单帧计算看看结果，即单帧超分辨率[10]。
4. 再有就是使用GPU进行进行并行计算的问题了。

Elad 的概率运动估计思想最核心的公式就是(27)式的内容。

3 盲去模糊算法

在解释我们使用的盲去模糊算法之前先引用一下CSDN上一篇简介性的文章：图像去模糊之初探-Single Image Motion Deblurring

<http://blog.csdn.net/dcraw/article/details/5849538>

此处我想解释的是：都是相机和场景之间的相对运动什么情况形成的是运动模糊，什么情况形成不会造成运动模糊。其实很简单，在相机一次曝光的时间内，场景相对相机的运动会造成运动模糊，而在两次曝光时间之间的相对运动不会形成运动模糊。所以上面在数据融合一节中说的场景相对相机发生亚像素级运动不会形成运动模糊，像素级运动会造成运动模糊是错误的。上面数据融合过程所作的运动估计实际是估计两次曝光时间之间场景相对相机发生的运动。而在盲去模糊中则是估计一次曝光时间内场景相对相机发生的运动。

虽说两个都是做场景相对相机的运动估计，但是估计的物理过程是不一样的。

3.1 模糊核的概念

模糊核物理意义表示的是相机相对场景的运动轨迹。形成模糊的原因也是相机在曝光时相对场景运动，这就使得某个传感器在一次RGB三通道信息积累的过程中是综合了多个来自不同点的反射光线。一个灰度级的值是多个不同点信息的平均。若是在一次RGB通道信息积累的过程中只综合了来自一个点的信息，就不会产生运动模糊。

在一次曝光时间内，场景和相机发生相对运动。以场景静止，相机运动为例，这种情况若相机是在平行于镜头的平面上仅发生平移运动，此时运动模糊就是均一的，仅用一个模糊核表示即可。模糊核没有空间相关性。但是若相机发生旋转运动，此时的模糊核就不是均一的，与空间位置有关。

这种运动模糊数学上以卷积表示，因为卷积实际表示的就是与固定的权重相乘累积求和。

$$Z = H \otimes X + V \quad (32)$$

其中 X 是清晰的图像， H 是模糊核， V 是加性噪声， Z 是模糊图像， \otimes 表示卷积运算。

此处目前仅仅处理均一模糊的情形，主要是相机发生平移造成的。文中(7)式实际是本问题的简化版本，去除了加性噪声。该问题的处理实际有两方面的任务：

1. 模糊核 H 估计；
2. 清晰图像 X 估计。

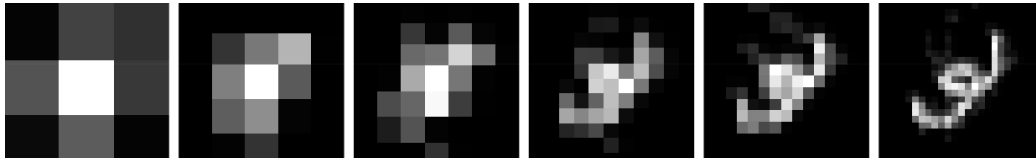


Figure 4: 多尺度模糊核估计过程[11]

将 H 用Toeplitz形式矩阵展开，在去模糊的文献中[11, 12, 13, 14, 15, 16]都是将模糊核表示成 k ，其Toeplitz形式记做 K ，为与文献统一，阅读方便，现将超分辨率理论中模糊核用 H 表示的传统换成 k ，上式就可以表示成

$$Z^\dagger = KX^\dagger + V^\dagger \quad (33)$$

$$Z^\dagger = Ak^\dagger + V^\dagger \quad (34)$$

其中， A 是 X 的Toeplitz形式矩阵，假设， X 的阶数为 $n \times n$ ， k 为 $m \times m$ ， k^\dagger 为 $m^2 \times 1$ ， X^\dagger 为 $n^2 \times 1$ ， K 的阶数为 $n^2 \times n^2$ ， A 的为 $n^2 \times m^2$ 。

(33)(34)两式在盲去模糊计算的过程中将分别发挥作用，其中(33)式用来估计 X ，(34)式则用来估计 k 。

去模糊算法在计算过程中采用多尺度计算的思想，基本过程是将初始图像先进行高倍下采样，这样处理的目的是进行高倍下采样之后的模糊图片，图片的模糊程度降低，模糊核运动形式变得简单。随着压缩尺度的减小，模糊核中也恢复了更多细节，运动信息更丰富，如图fig4所示。我在我写的MATLAB程序中依次使用的尺度是： $\frac{1}{10}, \frac{1}{7}, \frac{1}{5}, \frac{1}{3}, \frac{1}{2}, \frac{1}{1.5}, 1, \frac{1}{10}$ 表示图像压缩10倍进行计算。我们所采用的盲去模糊算法主要基于两篇文章的结果[16, 13]，其中Xu[16]的文章主要利用其两步模糊核估计的算法，Shan[13]的文章中主要利用其多阶导数去卷积的算法。

3.2 模糊核估计的模糊图像的预处理

此处主要说明如何从一张模糊的图像 Z 中通过预处理获取清晰图像 X 的某些信息。在单张模糊图像盲去模糊过程中，利用的信息主要是Step-Edge或者叫Sharp-Edge。fig5中就展示了Step-Edge被模糊核作用以后，虽然以前急速的梯度变化(黑色虚线)被平缓的梯度变化(蓝色实线)所代替，但是灰度级二阶导数为零的位置并没有因为模糊核的引入而发生变化。并且Step-Edge被模糊后灰度级的范围也没有发生变化。由此两点信息我们就可以构造一个模糊的Step-Edge边缘对应清晰图像中Step-Edge的近似。这个边缘的特点是以二阶导数为零的点，在梯度正反方向上各移动一个像素的位置，将该位置的灰度级设成灰度级的极大和极小值。

此处我们使用非线性扩散[17]和振荡滤波[18]实现模糊图像的预处理。在图像不含噪声时，仅使用振荡滤波可以达成模糊图像的预处理，但是

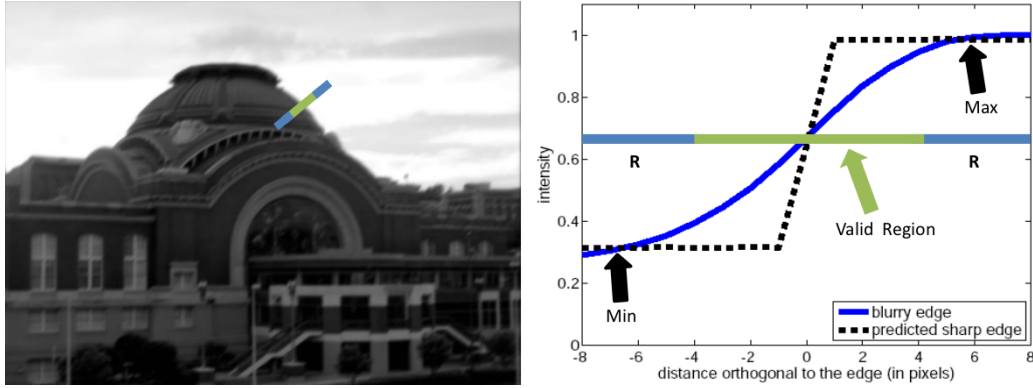


Figure 5: Step Edge被模糊核作用以后的变化[15]

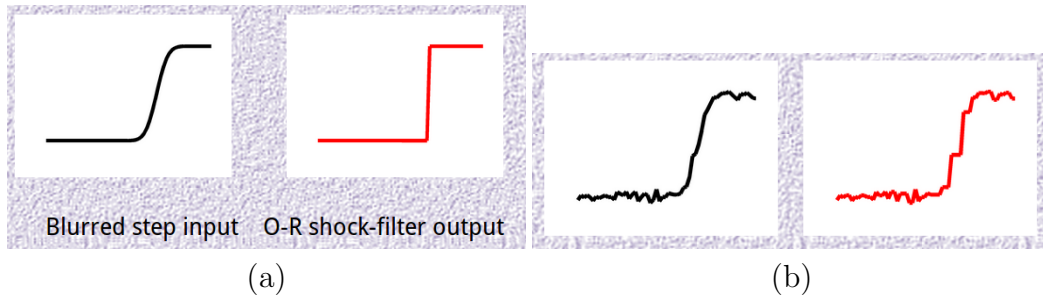


Figure 6: 振荡滤波器在有噪和无噪无情况对比，(a)是无噪情况信号，(b)是有噪情况

由于振荡滤波对于噪声敏感，有噪声时，就不能形成好的Step-Edge出来了。fig6所示。

为减弱噪声对于振荡滤波的影响，Xu[16]是使用高斯滤波，使用高斯滤波虽然能削弱噪声，但是也会减弱边缘，Cho[11]中使用的是双边滤波(Bilateral Filter)，此对于弱噪声效果较好，强噪声效果就不好，边缘影响也较小，但是其计算量比较大。此处我们使用Prona[17]提出的非线性扩散方法。其计算量大约是双边滤波的一半。fig7是线性和非线性扩散对比结果。

3.2.1 预处理过程数学描述

1. 非线性扩散数学描述[17]

$$I^{t+1} = I^t + \lambda(C_N^t * \nabla_N I^t + C_S^t * \nabla_S I^t + C_E^t * \nabla_E I^t + C_W^t * \nabla_W I^t) \quad (35)$$



Figure 7: 线性和非线性扩散对比，非线性扩散对于边缘影响较小。

其中， $0 \leq \lambda \leq 1/4$

$$\begin{aligned}
 \nabla_N I_{i,j}^t &= I_{i-1,j}^t - I_{i,j}^t \\
 \nabla_S I_{i,j}^t &= I_{i+1,j}^t - I_{i,j}^t \\
 \nabla_E I_{i,j}^t &= I_{i,j+1}^t - I_{i,j}^t \\
 \nabla_W I_{i,j}^t &= I_{i,j-1}^t - I_{i,j}^t
 \end{aligned} \tag{36}$$

$$\begin{aligned}
 C_N^t &= g(\nabla_N I^t) \\
 C_S^t &= g(\nabla_S I^t) \\
 C_E^t &= g(\nabla_E I^t) \\
 C_W^t &= g(\nabla_W I^t)
 \end{aligned} \tag{37}$$

$$g(\nabla I) = \frac{1}{1 + (\frac{\|\nabla I\|_2}{K})^2} \tag{38}$$

2. 振荡滤波[18]

$$I_{t+1} = I_t - \text{sign}(\Delta I^t) \|\nabla I^t\|_2 dt \tag{39}$$

其中， Δ 和 ∇ 分别是拉普拉斯算符和梯度算符。 dt 是步长。

从上可以看出，非线性扩散和振荡滤波都是迭代过程。

3.3 模糊核估计过程[16]

模糊核估计过程主要参考Xu[16]的文章，Xu先利用一般模糊核估计的方法估计出含噪的模糊核，再利用Iterative Support Dection(ISD[19])方法，将核的噪声去掉。但是目前算法实现中对于ISD的求解还有问题，ISD的数学求解方法还有待进一步讨论。其实对于模糊核的噪声去除除了ISD之外还有很多的方法，典型的就有Hysteresis thresholding(磁滞域值)的方法，该

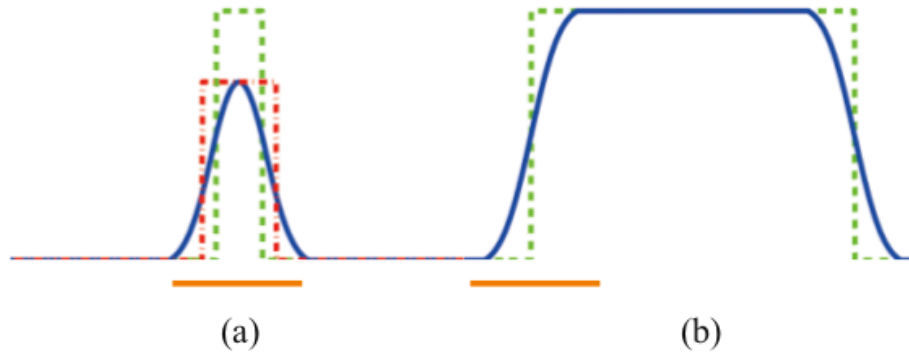


Figure 8: 相同模糊核对于不同尺寸大小的Step-Edge造成的影响。

方法最早是Canny[20]提出边缘检测Canny算子时提出的方法。Yuan[14]就使用了该方法来精细化模糊核。该方法主要的弱点就是域值不是自适应的，上下两个域值必须要人为设定。使用ISD的好处就是域值是自适应的，而且结果还相当好(至少Wang[19]和Xu[16]的文章中都表现出非常好的结果，虽然我还没有真正弄明白ISD该如何求解！)。

3.3.1 粗糙核估计过程[16]

模糊核粗糙估计使用的是多尺度估计迭代估计的过程。同时Xu在文章中还提到一类Step-Edge的情况，fig8所示，黄色实线表示的是模糊核的尺寸大小，绿色虚线表示的是清晰图像的Step-Edge。在(a)图中，经振荡滤波以后的结果是红色虚线。显然对于此类Step-Edge小于等于模糊核尺寸时并不能恢复出正确的Step-Edge出来。所以Xu就引入了R这样一个指标。R定义如下：

$$R_x(i, j) = \frac{\sum_{k, l \in N(i, j)} \nabla_x B(k, l)}{\sum_{k, l \in N(i, j)} |\nabla_x B(k, l)| + 0.5} \quad (40)$$

$$R_y(i, j) = \frac{\sum_{k, l \in N(i, j)} \nabla_y B(k, l)}{\sum_{k, l \in N(i, j)} |\nabla_y B(k, l)| + 0.5} \quad (41)$$

其中， $N(i, j)$ 表示的是以 (i, j) 为中心的窗口，其大小与模糊核大小相当。R指标的意义就是，当在 $N(i, j)$ 区域内若梯度是单调的R就会比较大，若梯度是双调或为零的R都比较小。0.5的作用就是是分母不为太小的值。0.5的值在灰度级 $\in [0, 1]$ 时是有效的。若是灰度级 $\in [0, 255]$ 时选择20来调节分母就可以了。不要太大，也不要太小。经过R的处理就会将fig8中(a)类情况给滤掉，而保留(b)类的情况。

再定义M

$$M = H(R - \tau_R) \quad (42)$$

$$R = \sqrt{R_x^2 + R_y^2} \quad (43)$$

τ_R 确定的规则如下: $\theta = \arctan(R_y/R_x)$, $\theta \in [-\pi/2, \pi/2]$, 依据 θ 将R分成四组, 分别是 $[-\pi/2, -\pi/4)$, $[-\pi/4, 0)$, $[0, \pi/4)$, $[\pi/4, \pi/2]$, 每一组中将依据R来从小到大排列, τ_R 的选择是保证每组中不少于 $0.5\sqrt{P_I P_K}$ 个元素。 P_I, P_K 分别是输入图像的像素个数与模糊核的像素个数。 H 是Heaviside函数, $H(x) = 0, x < 0; H(x) = 1, x \geq 0$

$$\nabla I^s = \nabla I^{sh} * H(M\|\nabla I^{sh}\|_2 - \tau_s) \quad (44)$$

其中, ∇I^{sh} 表示的是输入的模糊图像经过非线性扩散和振荡滤波以后的结果。 ∇I^s 表示经过筛选的一部分边缘。 τ_s 确定的规则与 τ_R 相似, 还是 $\theta = \arctan(\nabla_y I^{sh}/\nabla_x I^{sh})$, 同样依据角度分成四组, τ_s 的选择保证每组至少有 $2\sqrt{P_K}$ 个元素。至于为什么要分成四组, Cho[11]在其文章中有解释, 参见SEC.4, Fast Latent Image Estimation→ Prediction部分。不过目前我也只是大致了解这样做目的: 主要是Step-Edge有方向性, fig5中有展示, 所以尽量各个方向都有。 $M\|\nabla I^{sh}\|_2$ 中M的作用就是选择fig8中(b)类的边缘出来。

计算出 ∇I^s 就可以进行模糊核估计了。

快速的粗糙模糊核估计

$$E(k) = \|\nabla I^s \circledast k - \nabla B\|_2^2 + \lambda\|k\|_2^2 \quad (45)$$

求解过程如下, 同时将卷积用矩阵运算表示, ∇B 也用一维形式表示 ∇B^\dagger

$$E(k) = \|A_x k - \nabla_x B^\dagger\|_2^2 + \|A_y k - \nabla_y B^\dagger\|_2^2 + \lambda\|k\|_2^2 \quad (46)$$

$$\frac{\partial E(k)}{\partial k} = 2A_x^T(A_x k - \nabla_x B^\dagger) + 2A_y^T(A_y k - \nabla_y B^\dagger) + 2\lambda k \quad (47)$$

令(47)为零, 则有

$$2A_x^T(A_x k - \nabla_x B^\dagger) + 2A_y^T(A_y k - \nabla_y B^\dagger) + 2\lambda k \quad (48)$$

$$(A_x^T A_x + A_y^T A_y + \lambda)k = A_x^T \nabla_x B^\dagger + A_y^T \nabla_y B^\dagger \quad (49)$$

根据Plancherel 定理: 大概意思是空间域的二阶范数等于频率域的二阶范数, 即 $E(k) = E(\mathcal{F}(k))$ 。将(49)式转到频域求解, 则有

$$\begin{aligned} \{\widehat{\mathcal{F}(\nabla_x I^s)} \circ \mathcal{F}(\nabla_x I^s) + \widehat{\mathcal{F}(\nabla_y I^s)} \circ \mathcal{F}(\nabla_y I^s) + \lambda\} \mathcal{F}(k) = \\ \widehat{\mathcal{F}(\nabla_x I^s)} \circ \mathcal{F}(\nabla_x B) + \widehat{\mathcal{F}(\nabla_y I^s)} \circ \mathcal{F}(\nabla_y B) \end{aligned} \quad (50)$$

$$k = \mathcal{F}^{-1} \left(\frac{\widehat{\mathcal{F}(\nabla_x I^s)} \circ \mathcal{F}(\nabla_x B) + \widehat{\mathcal{F}(\nabla_y I^s)} \circ \mathcal{F}(\nabla_y B)}{\widehat{\mathcal{F}(\nabla_x I^s)} \circ \mathcal{F}(\nabla_x I^s) + \widehat{\mathcal{F}(\nabla_y I^s)} \circ \mathcal{F}(\nabla_y I^s) + \lambda} \right) \quad (51)$$

其中 $\mathcal{F}, \mathcal{F}^{-1}$ 分别是Fourier变换和其逆变换。 \circ 表示对应元素相乘，除也是对应元素相除。 $\widehat{\mathcal{F}(x)}$ 表示的是取其复共扼。同时注意此处计算都是使用二维形式，而非一维形式！

快速的去卷积过程

根据Xu[16]中此步，作者使用的是

$$E(I) = \|I \otimes k - B\|_2^2 + \gamma(\|\nabla_x I - \nabla_x I^s\|_2^2 + \|\nabla_y I - \nabla_y I^s\|_2^2) \quad (52)$$

同样利用上面使用的Plancherel 定理，则有

$$I = \mathcal{F}^{-1} \left(\frac{\widehat{\mathcal{F}(k)} \circ \mathcal{F}(B) + \gamma \widehat{\mathcal{F}(\nabla_x)} \circ \mathcal{F}(\nabla_x I^s) + \gamma \widehat{\mathcal{F}(\nabla_y)} \circ \mathcal{F}(\nabla_y I^s)}{\widehat{\mathcal{F}(k)} \circ \mathcal{F}(k) + \gamma \widehat{\mathcal{F}(\nabla_x)} \circ \mathcal{F}(\nabla_x) + \gamma \widehat{\mathcal{F}(\nabla_y)} \circ \mathcal{F}(\nabla_y)} \right) \quad (53)$$

但是此步，我想可以使用一些比较优秀的去卷积算法，如利用Shan[13]的结果。Shan的公式如下：

$$E(I) = \sum_{* \in \Theta} \omega^* \|\partial^* I \otimes k - \partial^* B\|_2^2 + \gamma(\|\nabla_x I - \nabla_x I^s\|_2^2 + \|\nabla_y I - \nabla_y I^s\|_2^2) \quad (54)$$

其中 $\Theta = \{0, 1, 2\}$ ， ∂^* 分别表示0,1,2阶导数。使用多阶导数的好处就是可以抑制去卷积的噪声。 $\omega^0 = 1, \omega^1 = 0.5, \omega^2 = 0.25$ 。可以尝试这样的去卷积公式。以后还可以尝试在惩罚项中引入局部先验，看看能不能抑制去卷积中的振铃效应。

Algorithm 1，说明了在粗糙模糊核估计过程中如何使用以上(51)(53)式的结果。

Data: Blurry Image B , Multi-Scale Coefficients C
Result: Raw Kernel k^o and Raw Latent Image I^o
 Build the Blurry Image pyramid with the Multi-Scale Coefficients C ;
for $l = C_1$ **to** C_n **do**
 if $l == C_1$ **then**
 $I = B$;
 edgetaper I, B with Gaussian Kernel with size equal to the size
 of k_l ;
 compute ∇I^{sh} ;
 compute R, M, τ_R, τ_s ;
 for $i = 1$ **to** m (m is the number of iterations) **do**
 (a) compute Selected Step-Edge ∇I^s .(eq.44);
 (b) Estimate Kernel k^i with Gaussian prior(eq.51);
 (c) Estimate The Latent Image I_i with Spatial prior and
 k^i (eq.53);
 (d) update $\tau_R \leftarrow \tau_R/1.1, \tau_s \leftarrow \tau_s/1.1$
 end
 $I^l = I_m$;
 $I^{l+1} = I^l \uparrow$;
 else
 edgetaper I, B with Gaussian Kernel with size equal to the size
 of k_l ;
 compute ∇I^{sh} ;
 compute R, M, τ_R, τ_s ;
 for $i = 1$ **to** m (m is the number of iterations) **do**
 (a) compute Selected Step-Edge ∇I^s .(eq.44);
 (b) Estimate Kernel k^i with Gaussian prior(eq.51);
 (c) Estimate The Latent Image I^l with Spatial prior and
 k^i (eq.53);
 (d) update $\tau_R \leftarrow \tau_R/1.1, \tau_s \leftarrow \tau_s/1.1$
 end
 $I^l = I_m$;
 if $l < C_n$ **then**
 $I^{l+1} = I^l \uparrow$;
 else
 $I^o = I^{C_n}$;
 $k^o = k^m$;
 end
 end
end

Algorithm 1: 多尺度粗糙模糊核 k^o 估计

3.3.2 精细核估计过程

在模糊核精细化过程中，Xu[16]借用的ISD思想来自Wang[19]的工作。Wang提出该方法是为压缩传感理论而来。在文章中Wang指出ISD方法适合非零元素呈现快速衰减分布的情况。而模糊核本身正满足这一要求。

此处需要说明的是，对于ISD的求解我没能弄懂！需要解决一个关键问题！ISD首先选择一个域值将 k 分成两组。

$$S^{i+1} = \{j | k_j^i > \epsilon\} \quad (55)$$

$$\widehat{S}^{i+1} = I - S^{i+1} \quad (56)$$

因为ISD是迭代求解， i 表示第 i 次迭代。

域值 ϵ 确定的规则如下：将模糊核 k 的元素从小到大排列， $d_j = k_{j+1}^i - k_j^i$ ，当第一次出现 $d_j > \max(k^i)/(2h * i)$ 时，定义 $\epsilon = k_j^i$ 。

$$E(k) = \frac{1}{2} \|\nabla I^s \otimes k - \nabla B\|_2^2 + \beta \sum_{j \in \widehat{S}^{i+1}} \|k_j\|_1 \quad (57)$$

对于(57)式的求解，Xu文章中的方法，我没有看懂，所以我就按照我看人家文章的经验，提出我的求解方法。我定义 $Fk^\dagger = \text{Vec}(k_j)$ ， $\text{Vec}(k_j)$ 表示由 $k_j(j \in \widehat{S}^{i+1})$ 构成的 $m^2 \times 1$ 的列矢量。(57)式就可以表示成

$$\begin{aligned} E(k) &= \frac{1}{2} \|\nabla I^s \otimes k - \nabla B\|_2^2 + \beta \|\text{Vec}(k_j)\|_1 \\ E(k^\dagger) &= \frac{1}{2} \{\|A_x k^\dagger - B_x^\dagger\|_2^2 + \|A_y k^\dagger - B_y^\dagger\|_2^2\} + \beta \|Fk^\dagger\|_1 \end{aligned} \quad (58)$$

因为上面形式是 $l1, l2$ 范数混合的形式。所以不能使用Plancherel定理来直接求解。所以此处使用共扼梯度下降来求解。

$$\frac{\partial E(k^\dagger)}{\partial k^\dagger} = A_x^T (A_x k^\dagger - B_x^\dagger) + A_y^T (A_y k^\dagger - B_y^\dagger) + \beta F \text{sign}(k^\dagger) \quad (59)$$

$$k_{m+1}^i = k_m^i - \alpha \frac{\partial E(k^\dagger)}{\partial k^\dagger} \quad (60)$$

其中 m 表示的是在某个 ϵ 下的迭代的次数。此种求解ISD的方法会有内外两层迭代。对于(59)式中各项的计算，根据Cho[11]文章介绍的方法。

$$A_x^T A_x k^\dagger = \mathcal{F}^{-1} \left(\widehat{\mathcal{F}(\nabla_x I^s)} \circ \mathcal{F}(\nabla_x I^s) \circ \mathcal{F}(k) \right) \quad (61)$$

$$A_x^T B_x^\dagger = \mathcal{F}^{-1} \left(\widehat{\mathcal{F}(\nabla_x I^s)} \circ \mathcal{F}(\nabla_x B) \right) \quad (62)$$

但是求解结果不能达到Xu[16]计算的结果。我不知错误在哪？还是程序实现过程出错？算法见Algorithm 2。

Data: $\nabla B, \nabla I^s, k^o$
Result: Refine Kernel k^r
for $i = 0$ **to** $otIter$ **do**
 if $i == 0$ **then**
 compute $\widehat{S^1}$ with k^o (eq.56);
 compute F ;
 for $m = 0$ **to** $inIter$ **do**
 compute k_{m+1}^i (eq.60);
 end
 $k^i = k_{inIter}^i$;
 else
 compute $\widehat{S^i}$ with k^{i-1} (eq.56);
 compute F ;
 for $m = 0$ **to** $inIter$ **do**
 compute k_{m+1}^i (eq.60);
 end
 $k^i = k_{inIter}^i$;
 end
end
 $k^r = k^{otIter}$;

Algorithm 2: k^o 的精细化算法描述

3.3.3 利用精细化模糊核 k^r 去卷积

该步利用(54)式完成。

$$E(I) = \sum_{* \in \Theta} \omega^* \|\partial^* I \circledast k^r - \partial^* B\|_2^2 + \gamma (\|\nabla_x I - \nabla_x I^s\|_2^2 + \|\nabla_y I - \nabla_y I^s\|_2^2)$$

在频域求解结果是:

$$I = \mathcal{F}^{-1} \left(\frac{\sum_{* \in \Theta} \omega^* \widehat{\mathcal{F}(\partial^*)} \circ \widehat{\mathcal{F}(k^r)} \circ \mathcal{F}(\partial^* B) + \gamma [\widehat{\mathcal{F}(\nabla_x)} \circ \mathcal{F}(\nabla_x I^s) + \widehat{\mathcal{F}(\nabla_y)} \circ \mathcal{F}(\nabla_y I^s)]}{\sum_{* \in \Theta} \omega^* \widehat{\mathcal{F}(\partial^*)} \circ \widehat{\mathcal{F}(k^r)} \circ \mathcal{F}(k^r) \circ \mathcal{F}(\partial^*) + \gamma [\widehat{\mathcal{F}(\nabla_x)} \circ \mathcal{F}(\nabla_x) + \widehat{\mathcal{F}(\nabla_y)} \circ \mathcal{F}(\nabla_y)]} \right) \quad (63)$$

其中 ∂^x, ∂^y 与 ∇_x, ∇_y 表示意思相同。 $\partial^* = \{\partial^0, \partial^x, \partial^y, \partial^{xx}, \partial^{xy}, \partial^{yx}, \partial^{yy}\}$, $\omega^* = \{\omega^0 = 1, \omega^1 = 0.5, \omega^2 = 0.25\}$ 。

3.4 盲去模糊需要改进的地方

1. ISD求解;

2. 目前去模糊是均一模糊的情况。但是在相机曝光时间内物体有局部运动时，即局部模糊的情况，该如何处理呢？即一张图像中，有的地方模糊，有的地方不模糊。**重点未来需要突破的地方，也是视觉体验提升的主要地方。**
3. 该算法目前的计算量还不大，需要使用并行计算的机会较小，但是局部去模糊的情况时，是否要并行化还不得而知。

4 未来工作的主要内容

以算法研究的工作来说，主要还是局部去模糊的问题。因为在相机曝光的时间内场景相对相机运动是非常普遍的情况。对于普通用户来说，还是非常实用的工作。关于数据融合的问题，我想主要还是提升算法的计算速度。

References

- [1] S. Farsiu, M. D. Robinson, M. Elad, and P. Milanfar, “Fast and Robust Multiframe Super Resolution” , *IEEE Trans. Image Processing*, Volume 13, pp.1327-1343,Oct.2004.
- [2] A. Zomet, A. Rav-Acha, and S. Peleg, “Robust super resolution,” in *Proc. Int. Conf. Computer Vision and Patern Recognition*, vol. 1, Dec.2001, pp. 645-650.
- [3] N. Nguyen, P. Milanfar, and G. H. Golub, “A computationally efficient image superresolution algorithm,” *IEEE Trans. Image Processing*, vol.10, pp. 573-583, Apr. 2001.
- [4] T. S. Huang and R. Y. Tsai, “Multi-frame image restoration and registration,” *Adv. Comput. Vis. Image Processing*, vol. 1, pp. 317-339, 1984.
- [5] M. Protter, M. Elad, H. Takeda, and P. Milanfar, “Generalizing the non- local-means to super-resolution reconstruction,” *IEEE Trans. Image Processing*, vol. 18, no. 1, pp. 3651, Jan. 2009.
- [6] H. Takeda, P. Milanfar, M. Protter, and M. Elad, “Super-resolution without explicit subpixel motion estimation,” *IEEE Trans. Image Processing*,
- [7] M.D.Robinson, PhD Thesis, “Estimation Theoretic Analysis of Motion in Image Sequences” Univ. Cali.Santa Cruz, Dec.2004.
- [8] M. Protter and M. Elad, ” Super-Resolution With Probabilistic Motion Estimation”, *IEEE Trans. Image Processing*, Vol. 18, No. 8, Pages 1899-1904, Aug. 2009.
- [9] S. Liu, M. S. Brown, S. J. Kim, Y. Wing Tai, “Colorization for Single Image Super Resoluton”
- [10] D. Glasner, S. Bagon, M. Irani, “Super-Resolution from a Single Image”
- [11] S. Cho, S. Lee, “Fast Motion Deblurring” *Proceedings of SIGGRAPH ASIA 2009*, 28, 5, Dec. 2009.
- [12] D. Krishnan, R. Fergus, “Fast Image Deconvolution using Hyper-Laplacian Priors”

- [13] Q. Shan, J. Jia, A. Agarwala, “High-quality Motion Deblurring from a Single Image”
- [14] L. Yuan, J. Sun, L. Quan, H. Shum, “Image Deblurring with Blurred Noisy Image Pairs”
- [15] N. Joshi, R. Szeliski, D. Kriegman, “PSF Estimation using Sharp Edge Prediction”
- [16] L. Xu, J. Jia, “Two-Phase Kernel Estimation for Robust Motion Deblurring”
- [17] P. Prona, J. Malik, “Scale-Space and Edge Detection Using Anisotropic Diffusion” , *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 12, No. 7, July 1990.
- [18] S. Osher, L. I. Rudin, “Feature-oriented image enhancement using shock filters” . *SIAM Journal on Numerical Analysis* 27, 4, 919-940.1990.
- [19] Y. Wang, W. Yin, “Sparse Signal Reconstruction via Iterative Support Detecion”
- [20] Y. Canny, “A computational approach to edge detection” . *IEEE Trans. on PAMI*.8, 6, 679-698.1986.