CLEAN Algorithm Roadmap

Renkun Kuang

October 9, 2019

Contents

1 Introduction **2**
1.1 Related Links 2

2 Original Paper:
Aperture Synthesis with a Non-Regular Distribution of Interferometer Baselines [link] **2**
2.1 Abstract 2
2.2 Introduction 3

3 NARO slide, Radio Imaging and CLEAN **4**

4 CLEAN Algorithm implementation **4**
4.1 Outline 4
4.2 Codes and Results 6

clean algorithm

1 Introduction

1.1 Related Links

Original Paper: Aperture Synthesis with a Non-Regular Distribution of Interferometer Baselines
NRAO - Radio Imaging and CLEAN
CLEAN Algorithm
NRAO - The 'CLEAN' algorithm
Build a Homebrew Radio Telescope
Fourier Synthesis Imaging
Radio2Space
Deconvolution with CLEAN
Hogbom's CLEAN algorithm. Impact on astronomy and beyond
An analysis of the properties of CLEAN and smoothness Stabilized CLEAN - some warnings
SparseRI: A Compressed Sensing Framework for Aperture Synthesis Imaging in Radio Astronomy
PPT - Deconvolution
Antennas & Receivers in Radio Astronomy
ADAPTIVE REAL TIME IMAGING SYNTHESIS TELESCOPES
Annular synthesis telescopes - A novel class of multiple beam telescopes operating at mm-wavelengths
aperture synthesis
Antennas in Radio Astronomy
Synthesis of Wide Beam Array Patterns Using Quadratic-Phase Excitations

2 Original Paper: Aperture Synthesis with a Non-Regular Distribution of Interferometer Baselines [link]

打算翻译本文，翻译的同时理解思考。

2.1 Abstract

In high-resolution radio interferometry it is often impossible for practical reasons to arrange for the measured baselines to be regularly distributed. The standard Fourier inversion methods may then produce maps which are seriously confused by the effects of the prominent and extended sidelobe patterns of the corresponding synthesized beam. Some methods which have been proposed for avoiding these difficulties are discussed. In particular, the procedure CLEAN s described in some detail. This has been successfully applied to measurements taken

with several different radio telescopes and appears to be the best method available at the time of writing.
在高分辨率射电干涉测量的实际过程中，不可能将测量基线布置的均匀。所以标准傅立叶反变换方法生成的 map 可能会被主旁瓣和扩展旁瓣(confused by the effects of the prominent and extended sidelobe patterns of the corresponded synthesized beam)污染（混淆，不知道旁瓣中的结构是真实源的结构还是只是（对应的脏束synthesis beam的）旁瓣）。下文讨论了一些消除这种影响的方法，特别地，详细介绍了在写作本文时效果最好的 CLEAN 算法。

2.2 Introduction

Aperture synthesis measurements are usually made at a set of interferometer spacings and orientations that form a regular pattern in the baseline (,v) diagram. Such a regular coverage has many practical advantages both in connection with the formal synthesis calculations and the later astronomical interpretation of the synthesis map. However, there are occasions when irregularities in the baseline coverage cannot be avoided. Interference or malfunctioning of some part of the equipment can make it necessary to reject certain portions of the measurements and this will leave gaps in an otherwise regularly covered u,v plane. The gaps give rise to undesirable sidelobes in the synthesized beam, making the synthesis map difficult or impossible to interpret. Similar problems arise when the u,v plane has been covered by a coarse grid of measurements as this will give rise to prominent grating responses in the synthesized beam pattern.
综合孔径测量通常通过一系列在基线(u,v)平面中，位置和方位分布均匀的干涉仪来实现。这种归整(regular)的覆盖在连接以下两方面有不少优势：
1. 早期的合成计算
2. 后期的对脏图(synthesis map)在天文上的解释

然而，基线覆盖不规则无法避免的情况可能会发生。
干涉或设备某些部分无法正常工作使得有必要删除一些测量数据，而这就会使理想的归整的 u，v覆盖存在缺口 gaps。
这些 gaps 会在合成束(synthesized beam)中产生不想要的旁瓣, 使得很难去解读合成图（synthesized map）。
类似的问题会在 u，v 平面被粗糙的测量网格 (coarse grid of measurements) 时发生, 因为这样会在合成束中产生主响应。(give rise to prominent grating responses in the synthesized beam pattern)
Need to know what are synthesized beam and synthesized map:
synthesized beam 就是进行 UV 覆盖采样 Visibility 的 可见度采样函数的 傅立叶逆变换：即脏束，其实就是 PSF?
synthesized map 就是可见度采样函数与实际天空理论可见度函数相乘得到的实测可见度函数的傅立叶逆变换：即脏图，就是实际观测到的图。

It may, in certain cases, be impossible (or impractical) to arrange for the interferometer measurementtho fall on a regular grid in the baseline (u,v) diagram. This is the case for

measurements taken with instrumernts such as the Caltech and Green Bank interferometers and, in general, for measurements that involve large inter- ferometer spacings. Interferometers can be operated over spacings up to the full diameter of the Earth and fram space vehicles and one shall ultimately want to use such measurements in a systematic way to synthesize very high-resolution maps of small diameter sources. Occultations of radio sources by the Moon give rise to similar problems. A few occultations of the same source (or one occultation measured at several observatories) will deliver a number of strip scans at a non-regular set of position angles. These strip scans are equivalent to a u,v coverage along radii at these same position angles.
在特定情况下，不可能（或不切实际）去将干涉阵的测量点落在基线 u，v平面里的一个归整的网格里。用类似Caltech和Green Bank干涉仪做的测量就是这样，大体上，对于涉及基线很长的干涉阵都如此。干涉仪可以在基线长度跨越整个地球直径甚至空间望远镜下运行，人们总是希望（系统地进行）这样的测量，以合成小直径圆的超高分辨率的图。月球遮挡射电源会导致类似的问题。同一个源的几个不同遮挡（或一次遮挡被几次观测到）将在一系列的方位角中带来一些条带的扫描。这些条带扫描等价于沿着相同方位角下的半径的 u，v覆盖。
其实标题就体现了这里讨论的问题

3 NARO slide, Radio Imaging and CLEAN

4 CLEAN Algorithm implementation

4.1 Outline

1. UV覆盖
   给定台站位置(经纬度)[(Longitude1, Latitude1),(东经，北纬)]，台站指向[(u1,v1,s1),(u2,v2,s2)] 采样间隔（秒，结合地球自转角速度），观测初始时间（年月日时分秒），观测结束时间（年月日时分秒），观测频率，根据地球真实位置，给出UV覆盖数据点坐标（以波长为单位，kilolambda（参考Radio Imaging and CLEAN））,也有用米为单位, 并能实现UV覆盖的绘制 →
   Telescope类
   telescope.py:
   class Telescope:
   def __init__(self,poslist,directionlist):
   self.poslist = poslist
   self.directionlist = directionlist
   def uvcoverage(self,dt,t0,t1,freq):
   pass
   return uvcover
   ...

   def fake_uvcoverage():
   return uvcover

调用：
tele1 = Telescope(poslist,directionlist)
uvcover = tele1.uvcoverage(dt,t0,t1,f)
tele1.uvplot()

2. Clean 算法：

→
Cleaner类：

cl = Cleaner()

属性：
cl.dirty_map
cl.residual
cl.model_beam
cl.cleaned_map
cl.brightness（模拟时传入的原始高清图）
cl.uvcover(uv覆盖)

方法：
cl.dirty_map_plot(scale="log")
cl.dirty_beam_plot(scale="log")（脏束，为uv采样的傅立叶逆变换）
cl.residual_plot(scale="log")
cl.model_beam_plot(scale="log")
cl.cleaned_map_plot(scale="log")
→
把这几个画图的合并到一起：
cl.plot(whichimg, scale = "log",isuv=False（默认不是画频率的图）,save = False（默认不保存图像））

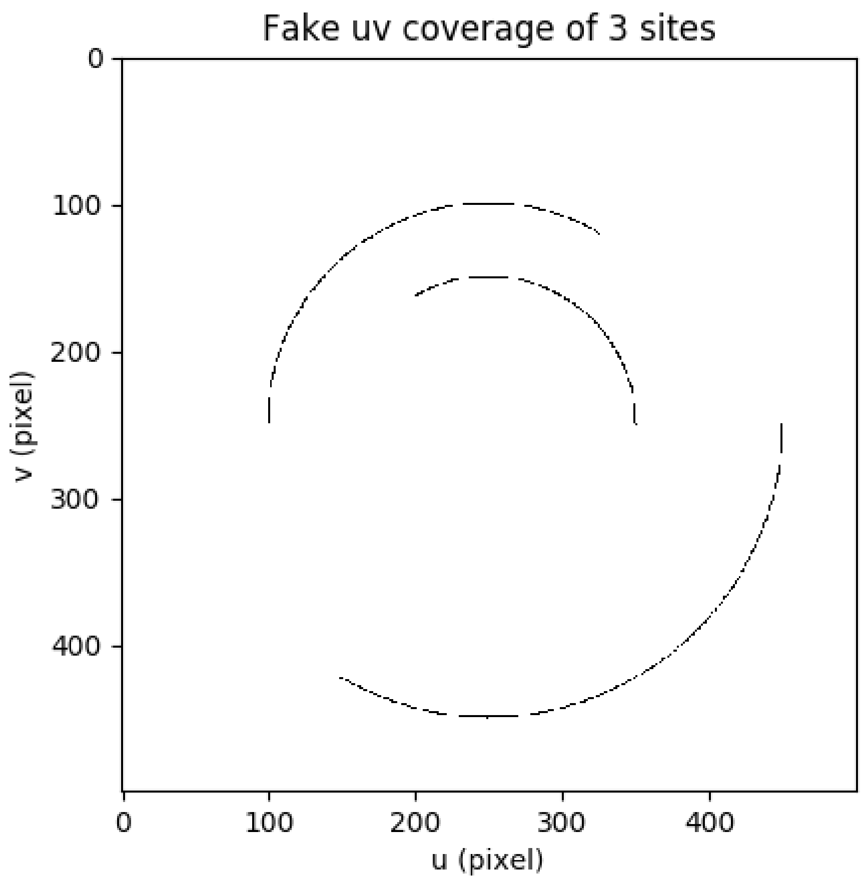
还包含的方法：
cl.clean(dirty_map,itertime=100,gamma=0.1)
visibility = cl.gen_visibility(self,self.brightness)（模拟时传入高清大图得到理论可见度图像）
sampled_visibility = cl.samp_visibility(self,visibility,uvcovrage)（模拟时，用uv覆盖采样理论可见度图像得到实测可见度图像）
dirty_map = cl.gen_dirty_map(self,samp_visibility)（从模拟或实际测量得到的实测可见度函数生成脏图），然后可与 cl.clean() 函数对接上了
cl.save_visibility()将实测可见度数据保存，要能与当前通用文件格式对的上，再写个读取实测可见度的函数，就能与实测数据对接上了。

3. 其他
   var_config.py → 配置台站，采样间隔，起始时间等参数，直接import 进测试代码主程序

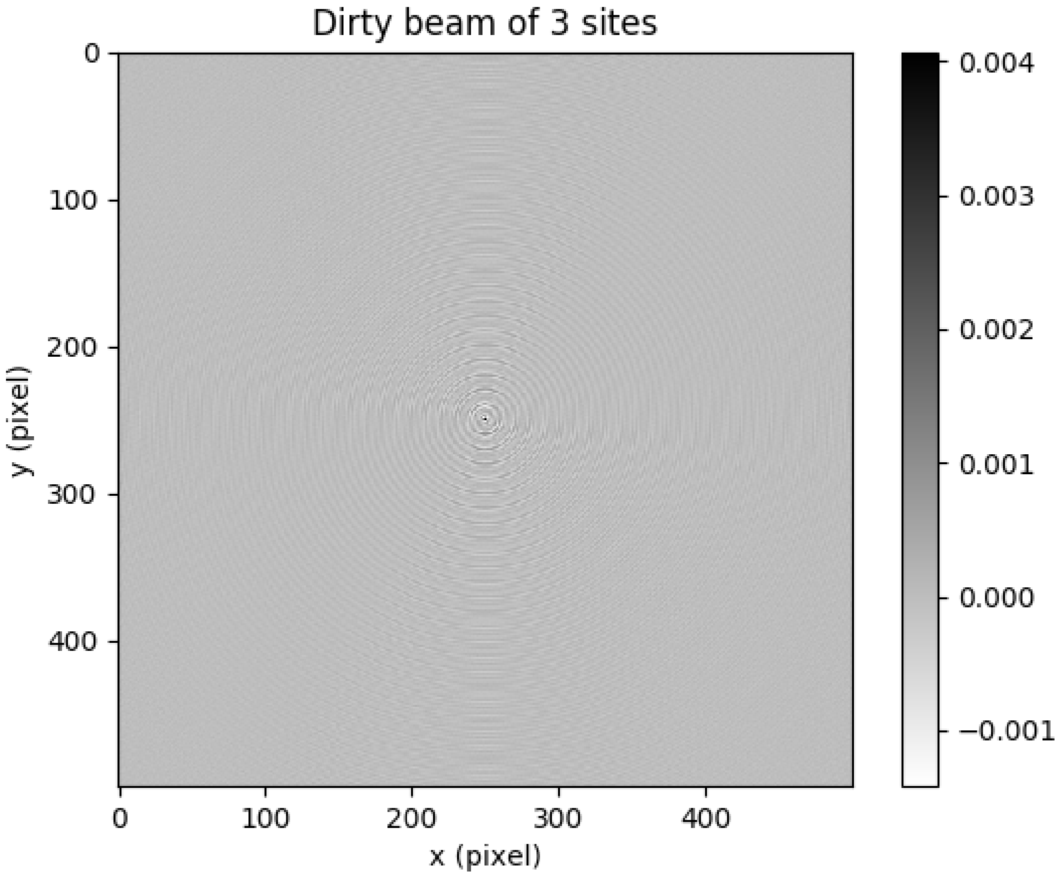
4. 模拟测试
   simu_test.py 测试代码主程序
   from telescope import Telescope
   from cleaner import Cleaner
   import var_config
   tele1 = Telescope(poslist)
   uvcover = tele1.uvcoverage(dt,t0,t1,f)
   uvcover.plot()

4.2 Codes and Results

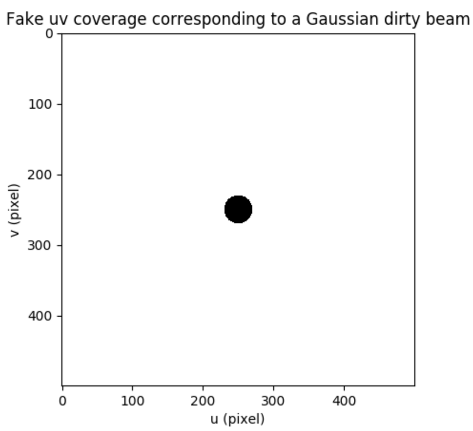
1. telescope.py
   先调用 fake_uvcover()函数产生一个虚假的UV覆盖, 如下图所示：

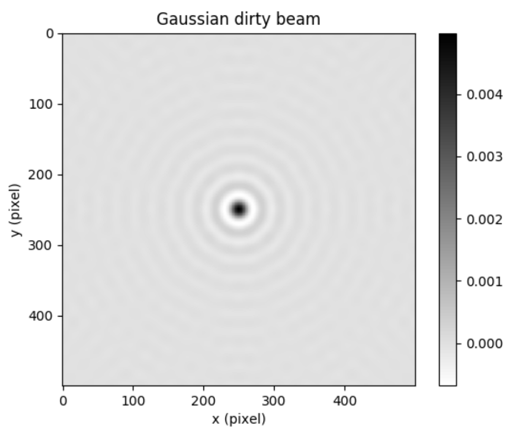


之后，又添加了画dirty beam的功能



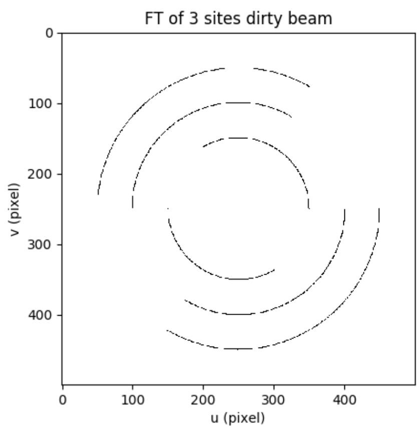
其它UV覆盖的测试：

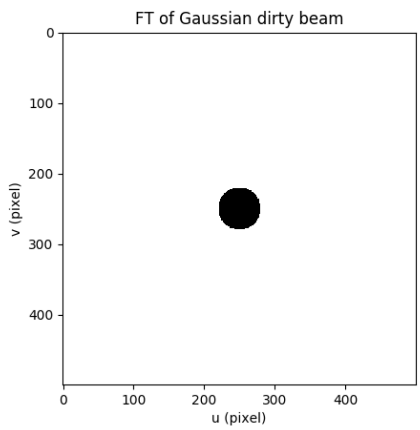




2. utils.py
由于画图函数 plot 重用需求较多，所以将其放入一个单独的文件 utils.py
另外读取原始测试图像（如lenna）的函数也放到 utils.py 里面

另外还将计算图像傅立叶变换的程序放到utils.py 里面：
可以测试前面的 dirty beam的FT是否和初始设置的 UV coverage 一样：





plt.imshow() → cmap = 'gray'
现在可以读取lenna图，做傅立叶变换得到理论可见度函数，再用UV覆盖采样得到实测可见度函数，之后进行傅立叶逆变换得到脏图 dirty map：
RC = (512,512)
tele1.fake_uvcover2(RC, 30):

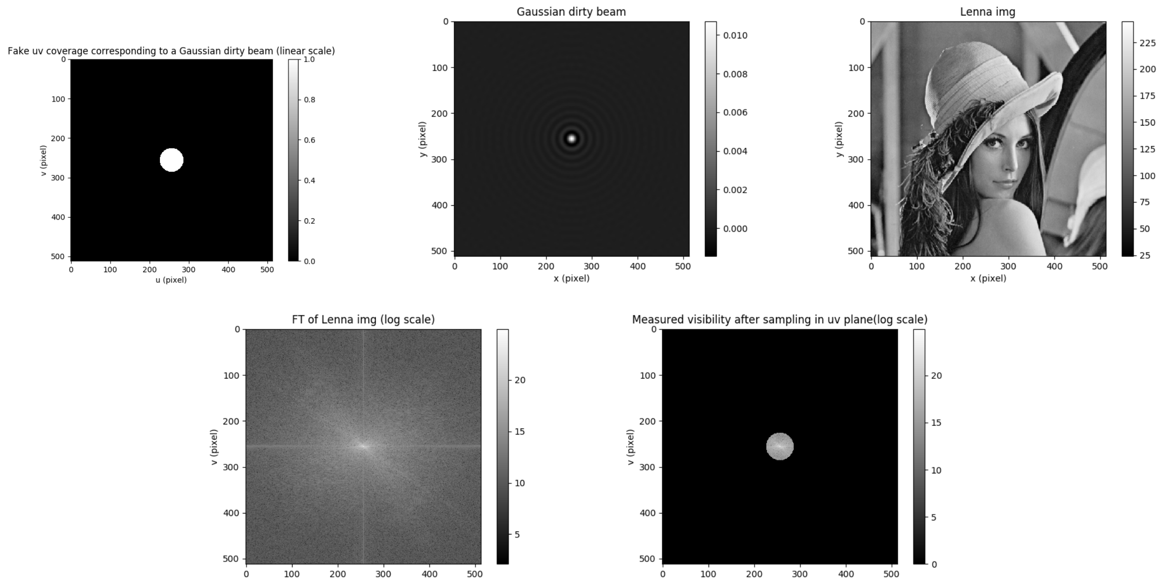
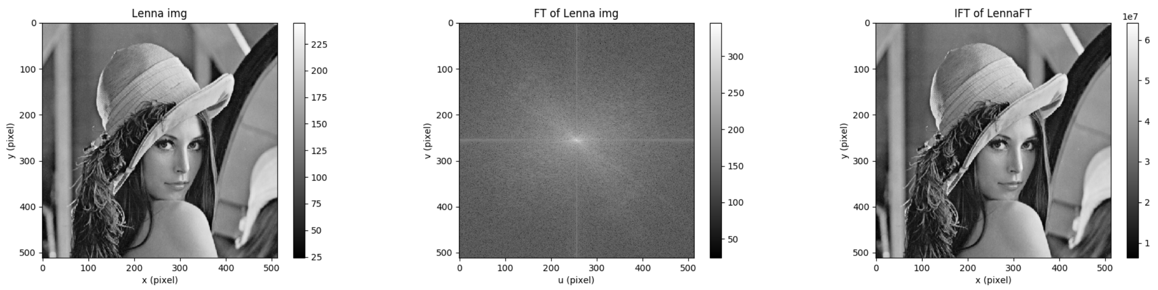


Figure 1: 当UV覆盖中心圆的半径为30个像素时的：UV采样函数、脏束、Lenna原图、理论可见度、实测可见度

前面的代码有问题，从实测可见度逆傅立叶变换得到的脏图不对。
重新验证：



现在可以了：

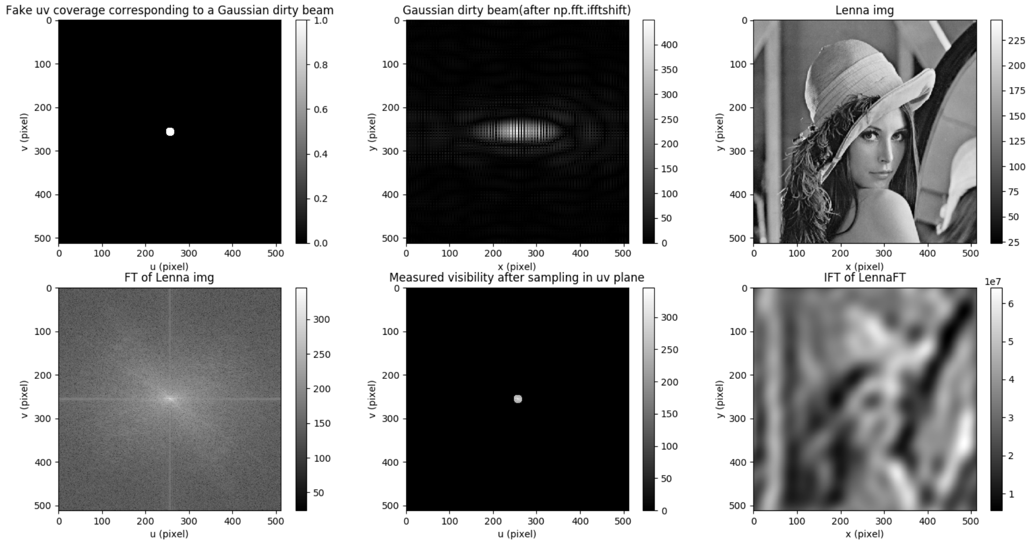


Figure 2: 当UV覆盖中心圆的半径为10个像素时的：UV采样函数、脏束、Lenna原图、理论可见度、实测可见度, 脏图

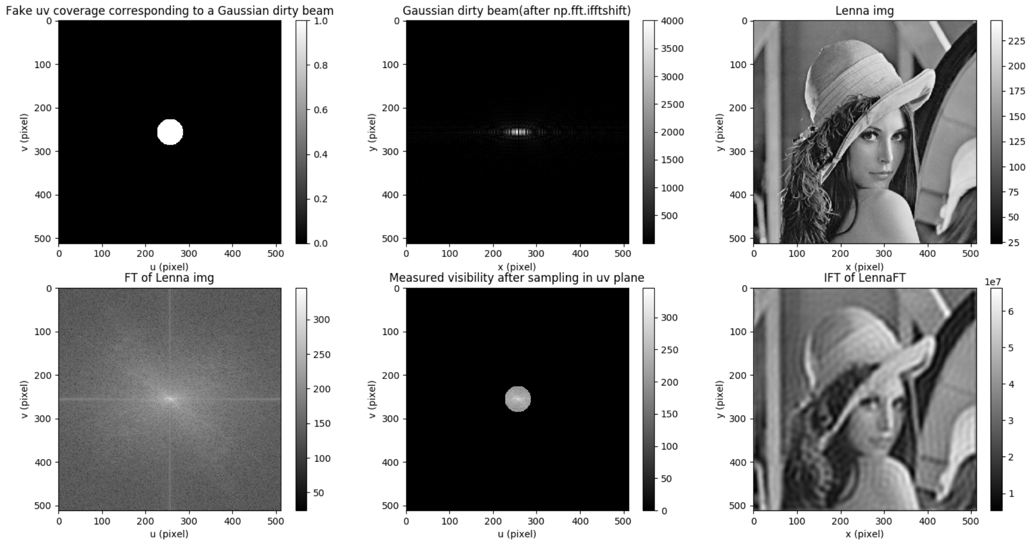


Figure 3: 当UV覆盖中心圆的半径为30个像素时的：UV采样函数、脏束、Lenna原图、理论可见度、实测可见度, 脏图

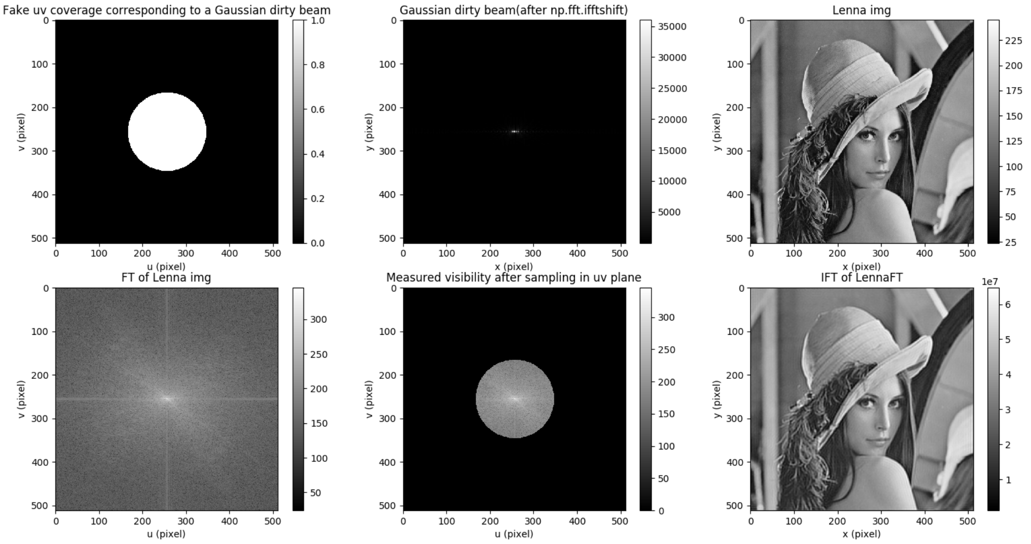


Figure 4: 当UV覆盖中心圆的半径为90个像素时的：UV采样函数、脏束、Lenna原图、理论可见度、实测可见度, 脏图

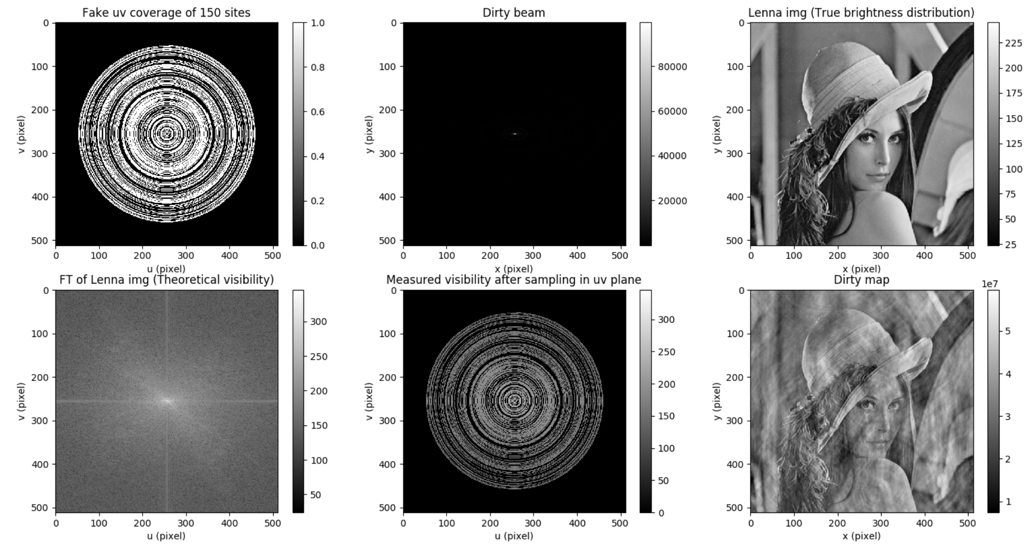


Figure 5: 当UV覆盖来自 150 个台站时的：UV采样函数、脏束、Lenna原图、理论可见度、实测可见度, 脏图