

理论学习笔记

Sunday, December 17, 2017 12:00 AM

<https://cn.mathworks.com/help/rtw/examples/adding-a-custom-toolchain.html>

<https://cn.mathworks.com/company/newsletters/articles/the-joy-of-generating-c-code-from-matlab.html>

关于代码生成的学习

<http://www.cnblogs.com/daocaoren/archive/2012/05/30/2526495.html>

C++ DLL 编写入门

<http://www.cnblogs.com/Imageshop/p/3281703.html>

目前的多语言实现，已经下载了dll
速度提升上guided image filtering比较重要

<http://blog.huajh7.com/2017/04/07/Image-dehazing/>

去雾算法的进展总结；
包含了很多很有用的思路

关于视觉

辐射，物理学上的辐射指的是能量以波或是次原子粒子移动的型态，在真空或介质中传送。

物理学上的辐射指的是能量以波或是次原子粒子移动的型态，在真空或介质中传送。包含：

- 电磁波：微波、可见光、X射线、 γ 射线(γ)
- 粒子辐射： α 射线(α)、 β 射线(β)、中子辐射
- 声辐射：超声波、声波、地震波
- 引力波

<https://caibolun.github.io/DehazeNet/>

关注这个使用深度学习来完成算法的例子
是一个很不错的实现

<http://1994epiphqny.com/2017/02/10/20170209/>

单通道去雾

基于暗通道先验的去雾算法

原始论文的中文版本

实现使用了python语言

<https://zh.wikipedia.org/wiki/WKB%E8%BF%91%E4%BC%BC>

WKB近似的介绍，可能没有用

<http://users.rowan.edu/~polikar/WAVELETS/WTtutorial.html>

小波分析入门

<http://www.jianshu.com/p/9e786be6dccb>

图像傅里叶变换

<https://docs.microsoft.com/zh-cn/visualstudio/cross-platform/>

<https://developer.android.com/guide/topics/media-apps/video-app/building-a-video-app.html>

<https://blogs.msdn.microsoft.com/c/2016/02/29/visual-studioc-android/>

<http://blog.csdn.net/column/details/androidrealfilter.html>

<http://anddymao.com/2017/03/27/2017-3-27-Android%E8%A7%86%E9%A2%91%E6%BB%A4%E9%95%9C/>

<https://developer.arm.com/graphics/tutorials/android-opengl-es-3-0-and-3-1-tutorials>

- 声辐射: 超声波、声波、地震波
- 引力波

辐射之能量会从辐射源往外向所有方向直线放射。一般依其能量的高低及电离物质的能力分类为电离辐射和非电离辐射。电离辐射所携带的能量大于10电子伏特(eV), 可以将原子或分子电离、打断化学键, 非电离辐射则否。主要电离辐射来源为放射性物质, 放射出 α 、 β 或 γ 射线, 分别带有氦核、电子、正电子、光子。其他电离辐射来源有医学影像造影使用的X射线、渺子、介子、正电子、中子, 以及宇宙射线与地球大气作用所产生的其他粒子。

散射, 当传播中的辐射, 像光波、音波、电磁波、或粒子, 在通过局部性的位势时, 由于受到位势的作用, 必须改变其直线轨迹, 这物理过程, 称为散射。这局部性位势称为散射体, 或散射中心。局部性位势各式各样的种类, 无法尽列; 例如, 粒子、气泡、液珠、液体密度涨落、晶体缺陷、粗糙表面等等。在传播的波动或移动的粒子的路径中, 这些特别的局部性位势所造成的效应, 都可以放在散射理论 (scattering theory) 的框架里来描述。

电磁波是一种最为人熟知, 最常碰到的辐射形式。其中, 光波散射是不可避免的日常生活现象。无线电波散射则乃雷达科技的核心物理机制。因为某些方面的不同, 电磁波散射可以清楚地分支为不同的领域, 各自有各自的取名。弹性散射 (涉及极微小的能量转移) 主要有瑞利散射和米氏散射。非弹性散射包括布里渊散射 (Brillouin scattering)、拉曼散射、非弹性X-光散射、康普顿散射等等。

大多数物体都可以被看见, 主要是因为两个物理过程: 光波散射和光波吸收。有些物体几乎散射了所有入射光波, 这造成了物体的白色外表。光波散射也可以给予物体颜色。例如, 不同色调的蓝色, 像天空的天蓝、眼睛的虹膜、鸟的羽毛[2]等等。奈米粒子的共振光波散射会产生不同的高度饱和的, 生动的色相, 特别是当涉及表面等离子体共振 (surface plasmon resonance) [3]。

在瑞利散射里, 电磁辐射 (包括光波) 被一个小圆球散射。圆球可能是一个粒子、泡沫、水珠、或甚至于密度涨落。物理学家瑞利勋爵最先发现这散射效应的正确模型, 因此称为瑞利散射。为了要符合瑞利模型的要求, 圆球的直径必须小于入射波的波长。通常上界大约是

[A7%86%E9%A2%91%E6%BB%A4%E9%95%9C/
https://developer.arm.com/graphics/tutorials/android-opengl-es-3-0-and-3-1-tutorials](https://developer.arm.com/graphics/tutorials/android-opengl-es-3-0-and-3-1-tutorials)

安卓开发

<http://www.opencv.org.cn/opencvdoc/2.3.2/html/doc/tutorials/tutorials.html>
OpenCV

<https://cn.mathworks.com/videos/video-processing-in-matlab-68745.html>
Matlab视频处理

<http://kaiminghe.com/cvpr09/>
何凯明主页

<http://blog.wolfram.com/2014/11/21/removing-haze-from-a-color-image-using-the-near-infrared-with-the-wolfram-language/>
Mathematica实现

<https://stackoverflow.com/questions/27008469/how-to-find-haze-of-an-image-on-matlab>
Matlab问答

当微粒半径的大小接近于或者大于入射光线的波长 λ 的时候, 大部分入射光线会沿着前进的方向进行散射, 这种现象被称为米氏散射。这种大微粒包括灰尘, 水滴, 来自污染物的颗粒物, 如烟雾等。即是形成所谓的丁达尔效应。

瑞利散射可以解释天空为什么是蓝色的。白天, 太阳在头顶, 当太阳光经过大气层时, 与空气分子 (其半径远小于可见光的波长) 发生瑞利散射, 因为蓝光比红光波长短, 瑞利散射发生得比较激烈, 被散射的蓝光布满了整个天空, 从而使天空呈现蓝色, 但是太阳本身及其周围只显白色或黄色。因为此时受到更多的蓝光散射而蓝光散射

是一个粒子、泡沫、水珠、或甚至于密度涨落。物理学家瑞利勋爵最先发现这散射效应的正确模型，因此称为瑞利散射。为了要符合瑞利模型的要求，圆球的直径必须超小于入射波的波长，通常上界大约是波长的 $1/10$ 。在这个尺寸范围内，散射体的形状细节并不重要，通常可以视为一个同体积的圆球。当阳光入射于大气层时，气体分子对于阳光的瑞利散射，使得天空呈现蓝色。这是根据瑞利著名的方程。阳光的蓝色光波部分波长比较短，散射强度比较大；而红色光波部分波长比较长，散射强度比较小。外太空的辐射通过地球大气层时，衰减的主要原因是辐射吸收和瑞利散射。散射的程度变化是粒子直径与波长比例的函数，连同许多其它因子，像极化、角度、以及相干性等。

瑞利散射不适用于直径较大的散射体。德国物理学家古斯塔夫·米最先找到这问题的解答。因此，大于瑞利尺寸的圆球的散射被称为米氏散射。在米氏区域内，散射体的形状变的很重要。这理论只能用在类球体。

瑞利散射和米氏散射都可以被视为弹性散射，光波的能量并没有大幅度地改变。可是，移动的散射体所散射的电磁波会产生多普勒效应，能量会稍微改变。这效应可以被用来侦测和测量散射体的速度，可以应用于光达（LIDAR）和雷达这一类科技仪器。

当粒子直径与波长比例大于10的时候，几何光学的定律可以用来描述光波与粒子的相互作用。在这里，通常不称这相互作用为散射。

对于一些瑞利模型和米氏模型不适用的案例，像不规则形状粒子，有很多种不同的数值计算方法可以让我们选择使用，求算散射的解答。最常见的方法是有限元方法。此法解析麦克斯韦方程组，寻求散射的电磁场的分布。程式工程师特别设计出复杂的软件，专门计算这类问题。只需要使用者给出散射体的折射率或折射率函数，电脑就可以计算出电磁场结构的二维或三维模型。假若结构比较庞大复杂，则需要高性能电脑大量的运算时间，才能得到结果。

另外一种特别的电磁散射是相干回散射（backscatter）。这是一个相当

利散射，因为蓝光比红光波长短，瑞利散射发生得比较激烈，被散射的蓝光布满了整个天空，从而使天空呈现蓝色，但是太阳本身及其周围呈现白色或黄色，是因为此时看到更多的是直射光而不是散射光，所以日光的颜色（白色）基本未改变——波长较长的红黄色光与蓝绿色光（少量被散射了）的混合。但因为人眼对不同颜色的敏感度不同，以黄绿色敏感度最高，往两边呈钟形分布，因此人眼对蓝色的敏感度远大于紫色，所以即使散射的可见光波长中紫光能量最高，人眼看起来仍是蓝色。当日落或日出时，太阳几乎在我们视线的正前方，此时太阳光在大气中要走相对很长的路程，所看到的直射光中的蓝光大量都被散射了，只剩下红橙色的光，这就是为什么日落时太阳附近呈现红色，而云也因为反射太阳光而呈现红色，但天空仍然是蓝色的，只能说是非常昏暗的蓝黑色。如果是在月球上，因为没有大气层，天空即使在白天也是黑的。

百度网盘：

用户名：Dehaze_Team

密码： pi3.1415926

C. Feng, S. Zhuo, X. Zhang, L. Shen, and S. Ssstrunk. "Near-Infrared Guided Color Image Dehazing," IEEE International Conference on Image Processing, Melbourne, Australia, September 2013 (ICIP 2013).

Mathematica实现的方法

<https://www.jianshu.com/p/0261e6cceb3e>

安卓开发

After nine attempts, there was a problem downloading the following file:

<https://go.microsoft.com/fwlink/?linkid=849033>

另外一种特别的电磁散射是相干回散射 (backscatter)。这是一个相当不为人知的现象。当相干辐射 (像激光光束) 传播通过一个拥有很多散射体的介质时, 电磁波会被散射很多次。一个代表性的多重散射介质例子是浓厚云块。朝着原本入射方向的反方向, 相干回散射效应会产生一个非常大的峰值强度。实际上, 一般的电磁波很大部分都会散射回去。对于非相干辐射, 散射通常会在反方向产生一个局部最大值。可是, 相干辐射的峰值强度是非相干辐射的两倍。测量这些数值是很困难的。原因有两个。第一个原因是, 直接地测量回散射同时也会阻挡入射电磁波。但是, 科学家已经想出精巧的方法来克服这问题。第二个原因是, 强度峰通常会是非常的尖锐。侦测器必须拥有非常高的角分辨率, 才能够看到峰值, 不会将强度峰值与邻近的低强度值平均起来。

安卓开发可以使用 kotlin 语言

安装java文档库
P29

<https://go.microsoft.com/fwlink/?linkid=849033>

Select Continue to install Visual Studio without downloading this file.
This might cause problems with other parts of the installation.

Select Retry to try downloading the file again.

Select Cancel to cancel the Visual Studio install.

<http://hukai.me/android-training-course-in-chinese/basics/firstapp/index.html>

安卓教程中文文档

<http://wiki.jikexueyuan.com/project/android-ndk-development-tutorial/>

NDK开发

http://blog.csdn.net/carson_ho/article/details/73250163
NDK与JNI

最开始写代码, 都是经过编译后生成汇编码, 直接在cpu上执行。因为不同的硬件架构和操作系统, 会导致不同的cpu支持的指令可能不同, 也就是说不通类型的cpu所能执行的代码就是我们说所的本地代码, 也叫本机代码或者native代码。这给程序的移植性带来很大的麻烦, 我们在一台机器上运行的好好的代码, 如果拿到另一个硬件架构不同的机器上去的话, 有可能运行不起来。C, C++写的代码等都是本地代码。但是, 本地代码的好处就是效率比较快, 因为它少了一个转换的过程。

后来, 形成了另一种编程模式, 例如: java。就是采用一种虚拟机的机制, 屏蔽掉底层硬件和操作系统的不同, 写的代码经过编译后形成一种中间语言代码, 例如

<https://www.cnblogs.com/Imageshop/archive/2013/04/07/3006334.html>

限制对比度自适应直方图均衡化算法原理、实现及效果 - Imageshop - 博客园

<https://github.com/Sar-Kerson/dehazeProcessor>

java的class字节码。这种中间码直接在虚拟机里面执行，虚拟机负责将中间码转换成其所在平台上能执行的本地机器码。在微软的.net平台下面的编程模型也是这样的，不管是什么语言，写的代码经过相应的编译器编译以后，会生成中间码，然后.net framework里面的虚拟机读入这种中间码，产生本地机器上的本地码。微软把这种产生中间码的语言所写的代码称为托管代码，意思就是说，这种代码是受.net framework管理的，不能直接访问内存等硬件，需要通过.net framework来访问。

<https://www.jianshu.com/p/9bf3d308ff7f>

<https://juejin.im/entry/5759783c207703006fe8464c>

Android Studio NDK 开发最佳入门实践