

DICOM 医学图像显示算法改进与实现^①

林红利^{1,2} 陈真诚¹ 王伟胜² (中南大学 信息物理工程学院 湖南 长沙 410083;

2. 湖南大学 计算机与通信学院 湖南 长沙 410082)

摘要: 为了解决医学图像处理过程中在需要频繁改变窗宽、窗位的情况下图像不能实时显示, 显示过程存在明显迟滞的问题, 分析了 DICOM 医学图像的显示过程以及传统显示过程中可以改进的地方, 提出了使用统一 LUT(Look Up Table)的方法来提高显示效率的解决方法, 并使用 Bresenham 算法改进了传统的 LUT 的生成算法。实验结果表明该改进后的算法能提高医学影像的显示效率。

关键词: DICOM; 医学图像显示; Bresenham 算法; 查找表

Implementation of the Improved Algorithm for DICOM Medical Image Display

LIN Hong-Li^{1,2}, CHEN Zhen-Cheng¹, WANG Wei-Sheng²

(1. School of Info-physics and Geomatics Engineering, Central South University, Changsha 410083, China;

2. School of Computer and Communication, Hunan University, Changsha 410082, China)

Abstract: To improve display speed of DICOM medical image with continuously adjusting window values, this paper describes the transformation of manufacturer dependent pixel values of DICOM medical image into pixel values that are meaningful for display, and proposes a solution using LUT. Finally, the Bresenham algorithm is proposed to improve the LUT generation algorithm. A good result is got and an affirmation from doctors is made in practice.

Keywords: DICOM; medical image display; bresenham algorithm; LUT(Look Up Table)

1 引言

随着 UI(超声成像)、CT(计算机断层成像)、MRI(核磁共振成像)、CR(计算机 X 线成像)、电子内窥镜、PET(正电子发射断层成像)和 MI(分子影像)等医学影像设备不断涌现, 利用计算机对医学影像设备采集到的图像进行后处理的医学图像处理与分析技术也越来越成为放射科医生和临床医生进行疾病诊断的重要辅助手段。在医学图像的处理过程中医生经常需要频繁地变换窗宽、窗位来对感兴趣的病灶部位进行细致的研究, 但传统的医学图像的显示方法不能达到实时显示图像的目的, 图像的显示具有明显的迟滞现象, 致使医生很多的时间浪费在等待影像的显示中, 工作效率比较低, 为此开发新的医学图像的显示算法来满足

实际医生操作的需要有着重要的实际意义。本文从讨论 DICOM 格式的图像显示过程入手, 分析了显示过程中的关键问题, 提出了统一 LUT 的解决方法, 并对 LUT 的生成算法进行了改进。

2 DICOM医学影像显示过程分析

DICOM 是美国放射学会和美国电器制造商协会组织指定的用于医学图像存储和传输的标准^[1,2]。目前几乎大部分的医学设备产生的医学图像都是采用这种格式。作为一种特殊的图像编码格式, 通用图像浏览软件并不能直接显示, 要显示 DICOM 格式的医学图像必须将原始图像数据经过一系列的转换才能得到可直接在显示设备上显示的数据(称之为 P-Values)。

^① 基金项目:湖南省自然科学基金(39420104,07J6133)

收稿时间:2009-09-18;收到修改稿时间:2009-11-21

DICOM 医学图像显示转换过程如图 1 所示, 需要经过 Modality LUT、VOI LUT、Presentation LUT 三个转换过程, 最终输出的 P Values 才是可以直接显示的图像数据。

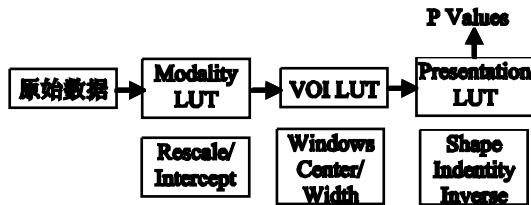


图 1 DICOM 图像像素转换过程

2.1 Modality LUT 转换(数据规范化转换)

通常不同生产厂商的设备很难保证在一种设备上生成的图像和其他生产厂商的同类型设备上生成的图像在度量上是一致的, 为此就需要将不同设备厂家产生的图像的原始数据转换到一个标准的度量空间, Modality LUT 转换就是完成这个功能的。医疗设备的生产厂商都会在自己的图像中采用 DICOM 标准规定的格式说明如何将数据转换为标准图像数据, DICOM 中规定可以使用通过查找表(Look Up Table, 简称 LUT)查找和通过斜率/截距(Rescale/Intercept)转换两种方法中的一种。

查找表方法(Modality LUT)是一种非线性变换算法。一个查找表由很多数据项组成, 每一个数据项为相应原始数值转换后的数据值, 同时还会提供一个被转换为查找表第一个数据项的原始数据的数据值。在进行数据转换时只要知道被转换为查找表第一项的原始数据, 比原始数据大 1 的数据则转换为查找表第二项的值, 比原始数据大 2 的数据转换后的数据则为查找表的第三项的值, 依次类推直到查找表的最后一项。大于查找表能转换的最大的原始数据则转换为查找表的最后一项的值, 小于查找表能转换的最小的原始数据则转换为查找表的第一项的值。用公式表示为:

$$M(Fp) = \begin{cases} M_0 & Fp \leq Fp_{\min} \\ M_{Fp-Fp_{\min}} & Fp_{\min} < Fp < Fp_{\min} + Ecount \\ M_{Ecount-1} & Fp \geq Fp_{\min} + Ecount \end{cases}$$

其中: M : 规范化后的标准图像像素值, Fp : 原始像素值, Fp_{\min} : 被转换为查找表第一项的原始数据, M_0 : LUT 中的第一个数据项的值, $M_{p-p_{\min}}$: LUT 中第 $Fp-Fp_{\min}$ 个数据项的值, $Ecount$: 查找表的数据项个数。

斜率/截距(Rescale/Intercept)变换是一种线性变换, 使用的公式是: 标准图像像素值 = 原始图像像素值 \times 斜率 + 截距, 斜率和截距在 DICOM 文件中可以读取。通过这一步转换后图像像素就从设备有关变成设备无关了。

2.2 VOI(Value Of Interest) LUT 转换(感兴趣区转换)

由于医学图像数据动态范围大(像素深度通常不低于 4096 个灰度级), 因此一般显示器很难提供如此高的动态范围一次显示整幅图像的全部信息细节, 在图像的处理中一般都是先选择一个操作者感兴趣的区域, 然后将该区域的图像信息映射到显示器能显示的整个数据范围, 这样就增加了该区域的图像信息的对比度。这个过程 DICOM 标准中称之为感兴趣区(value of interest 简称 VOI)LUT(Look Up Table)转换。临床医生感兴趣的窗宽、窗位调节功能就是 VOI LUT 转换的一种算法实现。VOI LUT 转换可以使用设置窗宽、窗位的线性转换算法和通过查找表(LUT)转换的非线性算法两种算法中的一种且只能使用其中的一种, 具体使用哪种算法在 DICOM 文件中有专门的标记来设置。

窗宽、窗位线性转换算法是根据预知的窗宽和窗位值获得需要显示的窗口大小(窗宽)和中心位置(窗位), 从而将窗口区域的图像数据线性地转换到显示器的最大范围内, 高于或低于窗口上、下限的图像数据则分别设置为最高或最暗的显示值。用公式表示为:

$$G(V) = \begin{cases} 0 & V < C - W/2 \\ \frac{gm}{W}(V + W/2 - C) & C - W/2 \leq V \leq C + W/2 \\ gm & V > C + W/2 \end{cases}$$

其中 V : 图像数据, G : 显示器显示值, gm : 显示器的最大显示值, W : 窗宽, C : 窗位。在这里窗宽是指需要显示图像的范围, 调节窗宽主要影响对比度, 窗宽越大图像灰度层次多, 组织对比度减少, 细节显示差, 而窗位也称窗中心, 表示显示区域的中心位置, 例如骨骼的窗位(C)、窗宽(W)分别为 $C=400$, $W=2000$, 那么就可以利用调窗处理将窗宽调节到骨骼窗宽, 窗位调节到骨骼的窗位, 然后利用上面的公式将图像 V 换算成显示器显示值, 最终得到的结果是只显示窗口范围内的图像, 也就是骨骼。

VOI LUT 是一种查找表算法, 算法处理过程同 Modality LUT。

2.3 Presentation LUT 转换

Presentation LUT 是对图像像素要做的最后一

个变换,它用于特定图像的显示。这一模块转换完成后的输出值为 P-Values, P-Values 是独立于任何显示设备的特性曲线,与人的视觉反应近似相关的值,可直接作为已经校正的软拷贝设备或硬拷贝的输入。Presentation LUT 转换也有两种转换方法,一种是通过 Presentation LUT 进行转换的非线性转换方法,一种是通过 Presentation Shape 的转换方法,这两种转换方法只能使用一种。Presentation LUT 转换的过程基本同上面介绍的两种 LUT 算法。Presentation Shape 算法使用 INDENTITY 和 LIN OD 两者中的一个。INDENTITY 表示经过 VOI LUT 转换得来的数据值即 P-Values 值,不需要转换,一般图像的软拷贝使用这个值;LIN OD 是实现图像的硬拷贝时使用的,在软件的开发中没有使用到就不讨论了。

3 算法改进

在图像后处理软件的开发过程中我们发现一幅 DICOM 图像从调入,到显示需要的时间比较长,而且医生在利用医学图像进行疾病诊断的过程中需要频繁地变换图像的窗宽、窗位来对感兴趣的区域进行仔细的研究,每次调窗处理后图像的显示具有明显的迟滞现象,达不到医生的临床诊断要求,为了能动态、实时地对图像进行调窗必须改进图像处理算法,缩短每次图像调窗处理时间。

3.1 使用统一的 LUT 算法

通过分析图像显示过程,我们发现医学图像从原始数据到 P-Values 要经过 Modality LUT、VOI LUT、Presentation LUT 三个处理步骤,而 Modality LUT、VOI LUT 又有线性变换和非线性变换两种算法,其中非线性变换算法是采用查找表的方法,这种方法相比线性变换算法来说的优点是变换效率高,而内存的占用会大些;而线性变换算法的优点是内存占用少,但算法效率不高,例如 512×512 的一幅医学图像来说,每一次 VOI LUT 变换需要进行 512×512 次的浮点数乘除法运算,对实时性要求比较高的医学图像处理来说性能上是难以接受的,为此借鉴 LUT 的非线性处理方法考虑将线性处理算法改造成 LUT 算法,同时找出医生实际操作时频繁需要的调窗操作对应的窗宽、窗位提取出来,在图像装入的时候创建其相应的 LUT,这样在医生进行调窗处理的时候不需要重新生成 LUT,直接从对应的 LUT 表中查表就可以了,这样可

以大大提高图像的调窗处理时间,达到实时调窗的性能要求。在和医生沟通后我们发现这个想法是可行的,因为在进行疾病诊断中医生一般会根据医学图像的类型和检查的身体部位不同去查看几种典型的窗宽、窗位。例如在 CT 图像中医生频繁使用的是骨骼 ($C=400, W=2000$), 胸腔 ($C=50, W=350$), 肺部 ($C=-600, W=1500$), 腹部 ($C=45, W=250$) 等。而 Modality LUT 考虑到图像操作时只使用一次,可以将原始像素规范化后的数据值保存在一个矩阵中,每次 VOI LUT 时只要直接使用这个矩阵中的值就可以了,不需要进行算法改造了; Presentation LUT 的线性处理算法在软拷贝中只使用 Presentation Shape 中的 INDENTITY 值也不需要算法改造;所以只需要改造 VOI LUT 的线性算法就可以了。

分析 Modality LUT、VOI LUT 和 Presentation LUT 发现 DICOM 标准的 LUT 是由多个数据项组成,第一个数据项保存的是第一个被转换的原始数据在查找表的对应值,第二个数据项为第一个被转换的数据项 + 1 的原始数据项在查找表中的对应值,依次类推,其数学表示方法为: $Y[0]=F(X_0)$, $Y[1]=F(X_0+1)$, $Y[i]=F(X_0+i)$, 其中 i 的取值范围为 $[0-LUT \text{ 数据项个数 } - 1]$, X_0 表示第一个被转换的原始数据。结合我们要实现的算法,要将线性算法改造成 LUT 算法首先必须计算出第一个被映射的原始数据的值,被映射的原始数据的个数(决定了要构造的 LUT 中数据项的个数),然后是构造 LUT。考虑到经过 VOI LUT 转换后的灰度值最小为 0,最大为 255,灰度间隔为 1,因此第一个被映射的原始数据的值为 $X_0=C-0.5-(W-1)/2$, LUT 数据项的个数为 W ,其中 C 为窗位, W 为窗宽。

3.2 LUT 生成算法优化

最简单的 LUT 生成算法是将取值范围在 $C-W/2 - C+W/2$ 之间的的整数(DICOM 中图像图像为整数)使用 VOI LUT 转换中的线性算法公式进行计算后保存到 LUT 中。分析发现这个线性算法需要进行大量的浮点数的乘法和除法运算,是比较耗时间的,如果能对这个线性算法进行改进将能提高图像的处理效率。再看线性转换的公式:

$$G(V) = \begin{cases} 0 & V < C - W/2 \\ \frac{gm}{W}(V + W/2 - C) & C - W/2 \leq V \leq C + W/2 \\ gm & V > C + W/2 \end{cases}$$

可表示为 $G=av+b$ 的形式,生成查找表的过程和

画直线的过程是一样的,因此可以使用计算机图形学中的直线扫描转换算法来对线性算法进行优化。直线扫描转换算法可以将大量的浮点数乘法、除法运算转换成加法和减法运算,提高运算效率。常用的直线扫描转换算法有数值微分(Digital differential analyzer, DDA)算法^[3,4]、中点画线算法^[5]、Bresenham算法^[6,7],其中以 Bresenham 算法效率最高,因此考虑使用 Bresenham 算法来生成 LUT。

3.3 改造后算法

在系统设计时我们使用一个全局的链表来保存当前系统中设置了哪些窗宽、窗位的信息,链表中的每个结点保存的数据类型是一个结构体,结构体的定义如下:

```
typedef struct WindowPar //定义结构体
{int i_count; //查找表数据项数
int i_FirstValue; //映射到查找表第一项的原始数据的值
int *LUTP; //指向查找表的指针
}
```

将 Bresenham 算法应用于 LUT 的生成的最终算法为:

```
Void CreateLut(int C,int W) //创建 LUT
{//C:窗位, W: 窗宽
int i_Ecount,i_FValue; //i_Ecount:查找表数据项数
```

目,i_FValue:映射到查找表第一项的原始数据

```
struct WindowPar WinP; //WinP: 查找表结构变量
//计算 LUT 数据项的个数
i_Ecount=W;
//计算第一个被映射的原始数据的值
i_FValue=(int)(C-0.5-(W-1)/2?);
//创建 LUT 数据结构
```

```
int *p=new int[i_Ecount];
```

//设置保存窗宽、窗位以及该窗宽、窗位下的 LUT 的首地址的信息到一个全局链表中,链表中的数组元素的类型是一个结构体

```
WinP.i_count= i_Ecount;
```

```
WinP. i_FirstValue= i_FValue;
```

```
WinP. LUTP= p;
```

//生成一个新的结点并加入到全局链表中,此处代码省略;

```
//循环执行 Bresenham 算法
```

```
//在这里因为显示器能表示的灰度范围为 0-255,
```

因此查找表的数据项的最小值为 0,作为查找表的第一个数据项,数据项的最大值为 255 为查找表的最后一个数据项。

```
float k,e;
```

```
int x,y; //x 表示被转换的原始数据值, y 表示转换后
```

的值

```
k=256/i_Ecount; //256 为显示器的能表示的灰度
值个数,即 LUT 的最大值
```

```
e=-0.5;
```

```
x= i_FValue; //第一个被转换的原始数据值
```

```
y=0; //转换后的第一个数据值,LUT 的第一项数据值
```

```
for(int i=0; i<=i_Ecount; i++)
```

```
{*P=y; //将数据项保存到向 LUT
```

```
P++ //指针前指
```

```
x=x+1; //被转换的原始数据值加 1
```

```
e=e+k; //e 在这里表示对应的查找表数据项的值是
否需要加 1,当 e>0 时表示查找表的数据项的值要加 1
```

```
if (e>0) //如果误差 e>0 则查找表数据项值加 1
```

```
{y++;e=e-1;}
```

```
} //循环结束
```

```
}
```

3.4 优化效果

通过优化后,效果比较明显,达到了医学图像实时调窗的要求。另外算法还可以继续完善,下一步我们计划借鉴计算机操作系统对内存的调度方法来进一步优化 LUT 的生成管理。

4 结语

本文在分析 DICOM 医学图像显示过程的基础上,使用 LUT 来统一 DICOM 医学图像的处理过程,并对 LUT 的生成算法进行了改进,并给出了具体的代码实现。实验证明改进效果明显,具有很好的实用价值。

参考文献

- 1 田捷,包尚联,周明全.医学影像处理与分析.北京:电子工业出版社,2003.20-40.
- 2 ACR/ NEMA. Digital Imaging and Communication in Medicine, draft Standard 3.0 2003, from ftp://ftp.nema.org/medi2cal/dicom, 2008.7.
- 3 Hearn D, Baker MP. Computer Graphics.2nd ed., Beijing: Tsinghua University Press, 1998. 120-127.
- 4 Bresenham JE. Algorithm for computer control of a digital plotter. IBM System Journal, 1965,4(1):25-30.
- 5 刘晶,李俊,孙涵.改进的 Bresenham 直线生成算法.计算机应用与软件,2008,25(10):133-137.
- 6 贾银亮,张焕春,经亚枝. Bresenham 直线生成算法改进.中国图形图像学报,2008,13(1):20-24.
- 7 孙岩,唐棣.并行的 Bresenham 直线生成算法.计算机工程与应用,2001,37(21):136-140.