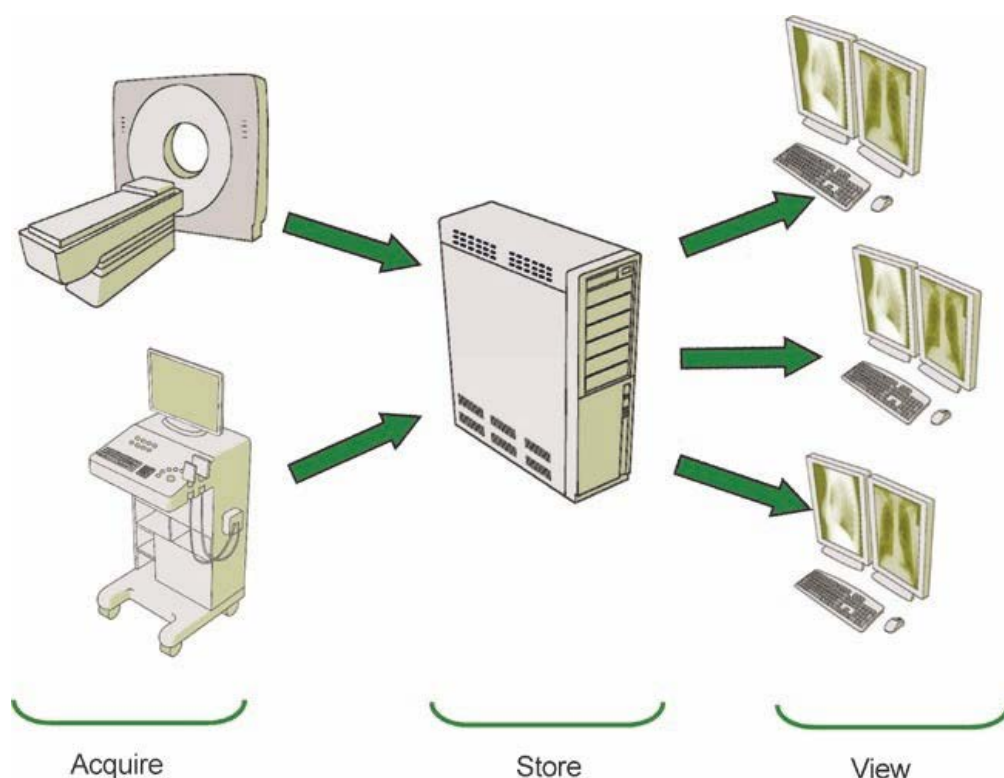


第一章， 啥是 DICOM

我们可以带着这个问题走进最现代化的、数字化的、时髦的医院，花上数小时来找个人，希望他能准确解答什么是 DICOM。但是我们通常只能扯一堆名词和缩略词，很少去考虑这些词的真正含义。哎，不知道你在干啥，就会离成功越来越远。

DICOM 就是医学数字成像和传输协议(Digital Imaging and COmmunications in Medicine)的简称，多年来致力于创造最广泛和根本的数字医学成像标准。比如，DICOM 提供使医学成像数据能够在诊断中被精确地表达和处理的所有工具。此外，与普遍看法相反，DICOM 不只是图像或者文件，他是数据传输、存储和显示协议的总和，用来覆盖数字医学成像的所有方面（这就是为啥说 DICOM 是标准集，而不是单个标准）。无疑的，DICOM 支配着日常的数字医学。

另一个重要的缩略词 PACS 图像传输和归档系统 (Picture Archiving and Communication Systems)。PACS 是一种医疗系统 (由必要的软硬件组成)，是被用来实现数字医学成像的。PACS 是由数字影像获取设备 (modality, 比如 CT、核磁、DR、CR、超声等)、数字影像归档 (archive, 他来存档影像设备产生的图像)，以及工作站 (workstation, 用来让影像科医师浏览影像)。当你用数码相机 (modality) 照相，在电脑 (archive) 上存储照片，并且将照片发送给你的好友 (reviewers 评论者，相当于在 workstation 上看片的大夫) 时，实际上就是 DICOM 模型的实例，当然实际情况会更复杂些。



PACS 直接与 DICOM 相关。PACS 的功能是由 DICOM 驱动的，DICOM 保证 PACS 的交互。因此 PACS 设备或软件具有他们自己的 DICOM 一致性声明 (DICOM Conformance Statement)，他是解释设备支持 DICOM 程度的重要文档。可以说，PACS 给 DICOM 带来了生命。

完全不能想象现代数字医疗没有 DICOM 和 PACS。DICOM 标准已经诞生 20 年，在数字医疗演化中扮演了重要的支撑作用，DICOM 一直以最高的诊断标准和最好的成效要求自己。它已经规划出了当代医疗的面貌：

1. 数字医疗的普遍标准。所有时髦的数字成像设备都会产生 DICOM 图像并通过 DICOM 网络传输。当代医疗工作流程也被 DICOM 规则的许多方面所左右。这些在本书中将给予介绍。
2. 绝佳的图像品质。比如，对于显示黑白图像，DICOM 支持最高 65536（16 比特）个灰阶，保证能够抓住医学成像中最细小的微妙所在。相较之下，将 DICOM 图像转换为 JPEG 或 BMP 图像，则只能保证 256 个灰阶，无法保证医生诊断读片的需要。DICOM 利用最新的和最先进的数字影像显示技术来提供绝佳的诊断图像品质。
3. 对于许多成像参数和不同数据类型的支持。DICOM 不仅能用来存图像，还能保存众多其他影像相关参数，比如病人的三维位置，在图像中被检对象的实际大小，层厚，图像曝光参数等。这些数据极大地充实了 DICOM 图像的信息内涵，并且方便在各个方面处理和表示图像数据（比如，用二维 CT 序列重建出三维影像）。
4. 医疗数据的完整编码。从病人姓名到图像颜色深度，再到当前病人的诊断信息，DICOM 文件和消息使用超过 2000 个标准的属性来传送各类医疗数据。这些数据是准确诊断所必不可少的，并且覆盖当代医学影像领域的所有方面。
5. 明确地描述了数字成像设备及其功能，是任何成像方案的支柱。DICOM 极准确地，基于独立的设备定义医疗设备的功能。通过医疗设备的 DICOM 接口工作，过程非常顺畅，产生错误的余地极小。

成书之时，DICOM 标准由 16 卷组成（9 和 13 已经弃用），传统上是从 PS3.1 到 PS3.18 进行的编号。2007 年发布的最新版本也在书中使用。

第2章， DICOM 是如何运作的

为了阐述复杂的医疗环境，DICOM 使用独特的，基于现实世界模型的术语（DICOM 信息模型）。

所有现实世界的对象——病人、检查、医疗设备等，DICOM将他们看作具有各自属性（properties or attributes¹）的对象。对象和属性是以DICOM信息对象定义（DICOM Information Object Definitions, IODs）为标准制定的。IODs是属性的集合，描述着每个特定的数据对象。一个病人的IOD，可以用姓名、电子病历编号（ID）、性别、年龄、体重、吸烟史等各类数据描述，囊括与病人有关的所有临床病人信息，能获得多少信息，IOD就囊括多少。病人（就像其他DICOM对象一样）其实就是一组属性，可参看图 2。DICOM维护一个所有标准属性（超过 2000 个）的列表，即所谓的DICOM数据字典（DICOM Data Dictionary）。DICOM依靠数据字典来保证属性命名和处理的一致性。比如，病人属性：姓名、生日、性别等在DICOM数据字典中都有。所有DICOM属性都要根据 27 个值表现（Value Representation, VR）类型进行格式化，比如日期、时间、姓名等。

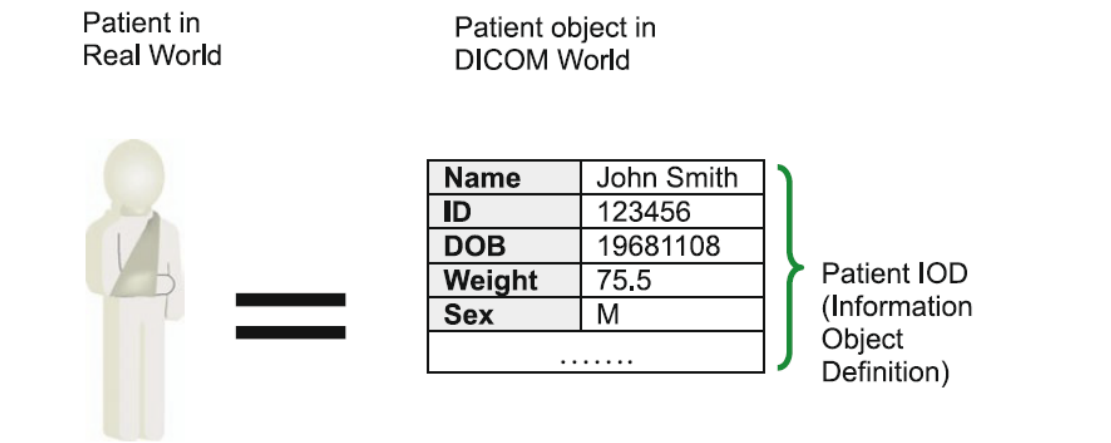


图 2.从现实世界到 DICOM 信息对象定义(IOD)，DICOM IOD 就是属性的集合

当数据以 DICOM 数据属性的形式被采集后，这些数据就能够在各类 DICOM 设备和软件（应用实体 Application Entities, AEs）间进行传送和处理了。DICOM 是依靠服务提供模型来描绘事物处理过程的。也就是说，一个 DICOM 应用会为参与处理过程的双方提供服务。每个服务通常都要涉及一些数据交换（特别是通过电脑网络进行交换）。数据交换过程中，服务会带着需要处理的数据（IODs）与特定的服务类型建立联系。DICOM 称这种联系叫做服务对象配对（Service-Object Pairs, SOPs），并且将 SOP 组合起来，称为 SOP 类（SOP Class）。比如，从数字 CT 机要将 CT 图像存储到数字 PACS 中，就可以称作：CT 存储 SOP，如图 3。在此例中，CT 图像就是服务所要处理的数据，所以 CT 图像即代表 DICOM IOD（DICOM 数据对象）。

¹ 详见 OO 等资料对于 property 和 attribute 这两个术语的讲解，本书如无特殊标记则均为 attribute。

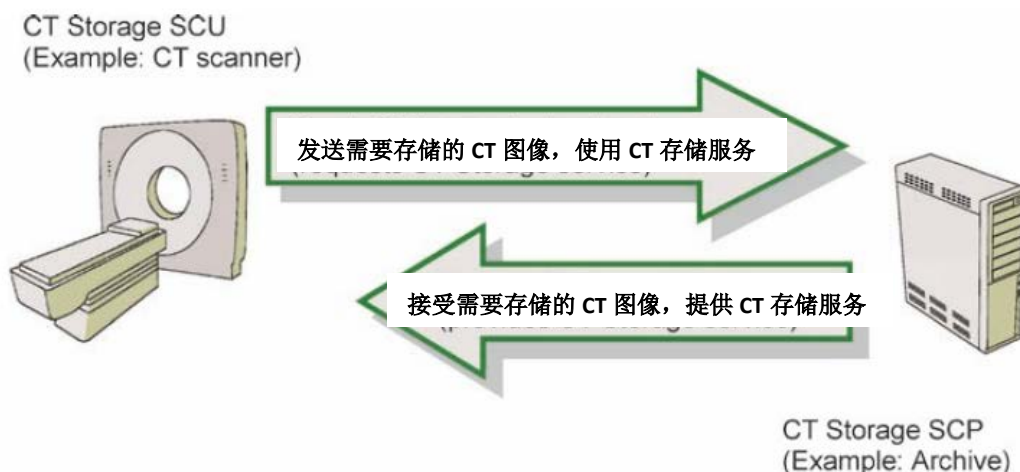


图 3. DICOM 服务

CT 机向图像归档请求图像存储服务，而图像归档向 CT 机提供存储服务。为了区分服务请求者和服务提供者，DICOM 称前者为服务类用户（Service Class Users，SCU），称后者为服务类提供者（Service Class Providers，SCP）。在之前 CT 的那个例子中，CT 机扮演 CT 存储服务类用户（CT Storage SCU），数字图像归档扮演 CT 存储服务类提供者（CT Storage SCP）。

每个 SCU 和 SCP 哥俩之间的 DICOM 网络数据交换都称作联系（association²）。因此每个网络传输都是从联系建立（DICOM 握手）开始的，即两个接通的应用开始交换彼此信息。

由于五花八门的 DICOM 设备和应用程序是由五花八门的 DICOM 厂商制造的，每个 DICOM 装置将从厂商那里继承它自己的 DICOM 一致性声明（DICOM Conformance Statement）。这个声明解释了装置所支持的 SOP（即服务），并且说明支持的范围（是 SCU，还是 SCP，还是两者都可以）。DICOM 一致性声明是我们完成 DICOM 相关项目中最必不可少的路引，因此一定要提前从厂商那里获得 DICOM 一致性声明并读懂它。比如，如果你买了个只支持 CT 存储 SCU（但是不支持 CT 存储 SCP）的数字图像归档，那么你是不能在里面存储 CT 图像的。这个归档实际上并不能提供 CT 存储服务。

本章这个简短的概要反映了 DICOM 功能性的核心，如您所见，通俗易懂。事实上，理解 DICOM 的理论容易，实际应用 DICOM 才是挑战。本书最大的目标就是帮助您接受挑战。

² 中文听起来有点怪，为了和 connection 区分。

第3章 从哪里获得 DICOM ?

DICOM 标准是免费的，能够在 DICOM 官网主页 (<http://medical.nema.org>) 上找到，该官网是有美国电气工程师协会 (National Electrical Manufacturers Association (NEMA)) 维护的。然而，从实际应用的角度里看，你通常还是在软件和设备的实施中才用到 DICOM。当前，市场上有成百上千的相关产品，不断地竞争，改版，争相搏人关注。这是物竞天择，然而这都会在原来 DICOM 标准的复杂性上添加一些常见的误会和混乱的观点。

此外，“你是否完全兼容过 DICOM”，这个问题永远不能那个用简单的“是”或“不是”来回答。事实上，这个问题问的就不准确。所有 DICOM 设备和软件只根据所需功能实现 DICOM 标准的特定部分而已。由于多样化的问题，在组织任何医学成像实践的过程中，实现 DICOM 工作流程已经变成最关键、最凶险的问题了。如何将 DICOM 引入实践，要避免什么？我们从最基础开始。

3.1 DICOM和数字化

像人类一样，我们通过模拟方式认知周围的世界。另一方面电脑在数字模式下运行，他们一个接一个点、一个数字接一个数字地存储和处理图像。DICOM 如它名字所说只能处理数字图像。因此将图像获取和转化为数字格式是实现 DICOM 的第一步。

所有当代的图像获取影像设备都提供数字图像输出。这些影像设备包括 CT、磁共振、超声、核医学等等。事实上，一部分这样的设备已经是数字化的，主要是由于他们自身的图像获取方法决定的，另一部分则因为考虑技术进步，而补充了数字线路。

问题是 DICOM 必然是数字化的，而数字化的不一定能保证 DICOM。你可能添置一台数字医学成像设备，但根本不支持 DICOM。

怎么知道如何在医学成像设备上使用 DICOM 呢？这很简单。如前面提到的，任何 DICOM 设备都应有 DICOM 一致性声明。这个一致性声明，在许多情况下，比设备用户手册还重要，并且应该由设备厂商提供作为设备所支持的 DICOM 功能的概要。也因此，在设备功能性方面，声明必须做到对于各种错误或者不正确的臆断不留任何的余地。

任何时候，当你计划或实施与 DICOM 有关的项目时，你应抱有监护自家设备 DICOM 一致性声明的精神。总是要求厂商提供它，并且要确保声明的版本与你的型号相符。相同的设备或软件都可能具有不同的型号和修订，因此你手中的一致性声明一定要与使用的型号或者版本一致。

3.2 DICOM , 兼容DICOM , 具有DICOM能力

如果说你已经清楚了自己的成像需求，并且准备好了购置 DICOM 装置。在完成核对之前你需要了解什么吗？显然需要。每一个兼容 DICOM 的设备或软件应用程序就像在昂贵的餐厅吃的晚饭。你的设备厂商会向你展示长长的主菜列表，他们中的大多数都是只有花俏的名字和昂贵的价格。每个主菜需要由各类可选的配菜来补充。慎重地选啊！这些可选的配菜有可能正是设备实现 DICOM 标准能力的关键。许多临床从业者都会犯这个错误：漏掉了选件，没有订购他们最需要的部分，最后只能哑巴吃黄连。

你也许会问：“为什么我们要购买附加 DICOM 设备选件来实现 DICOM 的功能呢，有个

无法实现 DICOM 的 DICOM 设备是毫无意义的呀？”是的，对你来说确实毫无疑义，但是对于厂商来说就不是这么回事了。为了更好地理解这对矛盾，你回顾一下之前所学。

所有当前的医学影像获取设备都是数字化的，但这不代表他们就一定具有 DICOM 功能。设备厂商能够实现他们自己私有的数字影像获取、存储和显示协议。这些厂商私有协议能够满足设备的单独运行，或与同一厂商设备的连接。因此，对于厂商来说，私有协议的使用涉及到方便和营销两方面的问题。方便方面是因为厂商能够在私有协议环境下，内部实现所有他希望实现的设备功能——不由任何标准或任何特定需求所限制。营销方面是因为私有标准要求你必须依靠该厂商——你不可能方便地连接其他厂商的器械和软件。

这听起来像个阴谋论，但是以实用主义观点来看，任何私有格式不可避免地分割了医疗设备市场。我们会在书的最后（第 12 章）仔细聊聊这些细节。

在许多案例中的底线是你必须明确需要购置作为选件的 DICOM 功能。如果不购买 DICOM 功能选件，则只能在设备上打上“具有 DICOM 能力”的标签，如果买了 DICOM 功能选件，这个设备就能实现 DICOM 功能了。

从另一个角度看这个问题则可以发现，从根上讲，DICOM 就是一个设备接口标准。他在外部连接设备，而不是在内部驱动。DICOM 提供设备输出（保证各种设备能够相互连接）的标准。当你购置 DICOM 选件时，你购买的是一种转变的、一种一致的、以及一种以 DICOM 格式输入数据的能力。

如果你对 DICOM 成像工作流程非常重视，那么你就必须认真对待厂商提供的所有 DICOM 选件列表。

如果任何选件都没有，你应该怎么办？明智的选择是：尽快走出急迫感和愧疚感，绝不要试图自己解决这些问题。如果你有一个缺少选件的，但具有 DICOM 能力的设备，唯一的补救方法是联系设备厂商。在任何情况下，让厂商协助解决 DICOM 问题是最合理的办法。不只是因为厂商会提供最好解决方案（即使你必须购买这些），而且能够为设备带来保证，并且这个方案对于器件本身没有任何伤害。

DICOM 兼容问题与前面所讲的类似，一般出现在较旧的成像器件上。如第 4 章所见，DICOM 标准已经出现很长时间，并且已经经过了一定的演化。考虑到 DICOM 设备的使用时间，他们是需要由厂商进行最新 DICOM 软件升级的。总的来说，这种情况和之前讲的差不多，但是有一个坎：设备的厂商可能已经倒闭了。如果真有这种情况出现，你最好采取以下的补救措施。

1. 原来的厂商可能被其他的厂商收购，这将会帮助你。
2. 原来的厂商关门了，但是许多 DICOM 装置仍然由其他厂商卖升级和保修。

从经验来看，数字医学设备用了超过 10 年就应该更换。除了 DICOM 标准方面的进步以外，整个数字医疗技术已经有了大踏步的前进。一定时间以后，较旧的装置不仅看起来有点原始，而且他们会缺少许多功能和特性，其中的一些甚至没有用现代科技来考虑 DICOM 标准。当这些成为了问题，不要再给恐龙们打补丁了；你最好把他们换成当代产品的型号。

3.3 在那遥远的、不靠谱的地方

还有一个实现 DICOM 功能的常见案例：医疗设备就是不支持 DICOM 标准。不管数字医疗科技，以及在医学成像工作流程中的 DICOM 化发展的有多么迅猛，不支持 DICOM 的装置仍然很多，并且有许多来源：

1. 数字医疗设备是使用非 DICOM 接口生产的。
2. 普通的非医疗设备（数字的或模拟的）。

3. 模拟医疗设备且没有数字输出。
4. 老旧, DICOM 之前的医疗设备。

最后一个情况, 即 DICOM 之前的装置不用太关注, 我们将在下面的 DICOM 历史章节中细致讨论。通常, 如果你的设备比 DICOM 要老 (实在是太老了), 确实是时候应该考虑买个新的了。

第一种情况很典型: 说到当代的数字医学成像设备, (比如许多流行的胶片扫描仪) 绝不表示它就具有 DICOM 接口。这通常要么出现在设备厂商想要与医学成像域保持一定距离的情况下; 要么就是由于厂商从事的并不是这个领域, 再或者是由于厂商考虑医学成像太过复杂且做出承诺太过麻烦。比如, 如果 CT 机必须服从 DICOM, 那么一个简单平板断层扫描仪就不一定会遵从 DICOM 了。

非 DICOM 数字设备的问题比缺少 DICOM 输出的设备要严重的多。这些设备有可能不能适应临床工作流程。比如, 每个 DICOM 图像必须包含病人姓名和编号 (ID); 这对于准确的图像路由和标识非常关键。如果你用一个胶片扫描仪来扫描病人的胶片, 想在非 DICOM 的数字图像格式 (比如 BMP) 中输入病人或检查的信息; 这简直是不可能的, 因为非 DICOM 图像格式不支持 DICOM 标签, 根本没有地方录入。另一个问题则是诊断图像的质量。当 DICOM 影像设备在 DICOM 图像的标签中自动存储所有临床数据时, 与之相当的非 DICOM 设备则需要人工处理这些信息, 这经常导致意外错误和数据丢失。

说了这么多, 如果你依然要购买非 DICOM 接口的数字医疗设备, 至少考虑一下以下的预防措施:

1. 尝试从 DICOM 厂商那里买这个设备, 这些厂商可能提供设备厂商无法提供的 DICOM 接口。
2. 不要把非 DICOM 设备设置在你临床工作的核心。

相同的逻辑适用于使用普通的、非医疗设备。比如, 许多皮肤病学家喜欢使用现成的数码相机获取病人皮肤的形态, 感染以及其他情况 (这是非常合理的临床做法)。显然, 现成的数码相机肯定没有 DICOM 功能, 以后也不会有。因此如果你想要将这些图像放入你的 PACS 中或者进入普通的医学成像工作流程中, 你也同样会遇到图像格式转化和图像标识的问题。因为这些设备就不是给医疗用的, 你从厂商或 DICOM 厂商获得 DICOM 援助的时机是在获取图像之后。去找找现成的 DICOM 软件, 看看是否有图像导入选项。许多当代的 DICOM 工作站和文件浏览器都具有 DICOM 导入功能, 让你能够将常规的数字图像 (BMP, JPEG 等) 转化成标准 DICOM 图像。再说一次, 你不得不手工输入缺失的信息 (病人姓名, ID, 检查日期等), 但作为你的回报, 你将能够将这些图像引入你的临床工作中。在更通常的情形下, 你都可以用这种方法对付来自任何地方的非 DICOM 图像文件。

许多医疗设备仍然是模拟的并且还将保持模拟状态一段时间 (比如没有成像装置的心电图 ECG, 或者某些特定的影像设备, 比如超声也通常不配备数字接口)。如我们所知, 一个设备必须先数字化才能实现 DICOM。因此, 模拟图像必须被数字化。静止图像可以转化为数字格式而视频则可以打散成一组数字图像。数字化和 DICOM 转换的工作通常由 DICOM 转化器实现, 它是具有数字化、DICOM 化模拟图像、视频、以及电影胶片的小盒子状设备。此外, DICOM 转化器通常具有医院信息系统接口 (来获得病人数据, 避免人工录入) 并且能够往 PACS 或其他 DICOM 设备发送转化后的图像, 因此它也支持 DICOM 网络功能。DICOM 转化器主要针对的是残存的模拟成像设备市场, 但是总的来说可以用在任何一个模拟成像设备上。另一个好事是, 一些 DICOM 转化器事实上很聪明, 能够用在一些主要的医疗制造商的某些私有成像格式上, 并能很好地转化为 DICOM 格式。除 DICOM 转化器之外, 越来越多的当代 DICOM/PACS 产品已经开始提供先进的图像输出组件了, 主要是为了适应非 DICOM 的数字图像。

最后总结一下，一般的、简陋点的、不符合 DICOM 的成像设备还是可以用来参与 DICOM 工作流程的。尽管如此，如前所述，图像质量和记录管理都比较成问题。此外，这些工作流程中掺入的人工处理会让工作更迟缓和容易出错。因此，这种临时的 DICOM 解决方案更适合用在低业务量、次要的成像方案中，或者是作为老旧系统到新 PACS 中的临时传输方案。但事实上，医疗产业中，医疗成像工作流程必须使用 100%符合 DICOM 的设备的。

第4章 DICOM 简史

由于 DICOM 标准已经有 20 岁了，所以它的历史已经成了它存在的主要部分；了解 DICOM 的历史可以帮助你解答许多现在的问题。此外，虽然时常进行改版，但是标准并没有发生根本性改变。为了适应当前的形势，DICOM 标准依然在不停地演化和调整，然而它仍保留着许多原来的历史容貌。

正常情况下，DICOM 设备的制造、销售和使用将跨越许多年（影像设备很贵，医院管理员比较保守而且精打细算，希望能让设备发挥最大效用），有时甚至达到了不散架就不放手的程度。我们是否应该考虑所有废弃的、修复的以及普通的老装置在许多工作中仍然要作为一种负担得起的选择。这些状况导致医院不太喜欢进行彻底的更新，而是要求更新必须能够兼容较旧的设备。

简而言之，如果你在当代的、复杂的临床环境中工作，你将不得不使用多个时代（符合或不符合 DICOM）的设备，并且你一定要时常进行一些“考古”发现，发现你正在应付 DICOM 历史的不同层次。

4.1 这一切是如何开始的？

1983年，由美国放射学院（ACR）和美国电气工程师协会形成的联合委员会构思了DICOM标准。最初的目标是创建一个不依赖于特定设备厂商的数字医学成像标准，这对于数字成像和PACS的发展起了推动作用。当我们看着其他产业还在为了谁听谁的而争夺的不亦乐乎的时候，我们应该对构思DICOM的人们的远见敬佩不已，因为早在计算机和网络普及之前，他们就在考虑数字医疗应用架构了。

联合委员会被命名为ACR-NEMA数字成像和传输标准委员会，开始通过回顾许多其他标准来建立自己的标准。虽然委员会在寻找能够明确符合他们要求的内容方面一无所获，但是从研究中收获了一些有价值的建议。当时，美国医学物理学家联合会（AAPM）已经放弃了一个在磁带上记录图像的标准。在这个标准中，AAPM试图通过数据元素（data element）序列（sequence）来编码信息，不同元素有不同的变量长度并且通过唯一的名称（标签，tag）加以区分。这种用标记的数据元素序列展现数据的方法被ACR-NEMA所采用。如果你对超文本标记语言（HTML）有一定的了解，最好对可扩展标记语言（XML）也有了解，那么你应该马上就能够找出这些当代流行标准之间的相似之处。用数据元素作为小的组件来表达任何具有复杂性的数据的方法被证明是特别有用和强悍的（如图4）。

数据=[数据元素 1]+ [数据元素 2]+……+[数据元素 N]

比如：

病人=[姓名]+ [年龄]+ [体重]+……+[性别]

图4 将数据打散为数据元素

第一个版本称作ACR-NEMA 300-1985或ACRNEMA 1.0，在1985年出版，并且在北美放射协会（RSNA）的年会上发表。原始的ACR-NEMA标准被提议作为准则，而NEMA不承担任何强制实行和解释说明的责任。标准化的目标是做成牢固的和必须的，并且在医疗共享方面，服从标准已经变成事实上必须去做的事了。

作为第一个版本，ACR-NEMA 1.0包含一些错误和不完美。委员会很快意识到标准需要在不断努力和更好的组织结构中进行深入的工作。因此，AR-NEMA采用了工作组（Working

groups WGs) 的办法, 工作组是独立的子委员会致力于提高标准的特定部分。为了继续提高 ACR-NEMA 1.0 第一个工作组 VI (现在所知的 WG-06, 基础标准) 成立了。工作结果是第二个在 1988 年诞生——ACR-NEMA 2.0 (或称作 ACR-NEMA 300-1988)。修订版已经足够强大到让医疗设备厂商去采用, 缓慢而稳妥地开始在医疗设备接口中使用。甚至现在, 你也能够找到一个老 CT 机或数字归档正在运行 ACR-NEMA 2.0。在现在 DICOM 标准的基础兼容性中仍然保证 ACR-NEMA 2.0 的延续, 不管他变得多么过时。

要是没有计算机网络, ACR-NEMA 2.0 还能统治医疗世界更长时间。ACR-NEMA 2.0 所提出的设备之间医疗数据通讯是很受限的。比如, 用户能够发送图像给移动设备, 但是标准没有规定这些设备能够如何处理这些图像。这些功能上的差距, 在 80 年代的网络科技大发展时期暴露无遗, 因此只是简单地打补丁已经无法解决问题, 大家急需一个新的修订版。

这个新的修订版需要考虑的另一个问题是需要适应数字设备和他们通讯协议的快速变化。不只是设备需要一个新的或更加抽象的方法来看待数字信息流程, 他们同样需要一个坚实的数字医疗信息模型。

为了对于变化做出响应, 最基础、最原始的第三版的 ACR-NEMA 标准雏形在 1992 年的 RSNA 上面世了。之后一年每月进行工作组会议。最早完成的新 ACR-NEMA 标准的前 9 个部分在 1993 年 9 月完成并且在 1993 年的 RSNA 上展示了具有更多功能的版本。这个经过翻新的标准被称作 ACR-NEMA DICOM, 或者被称作 DICOM 3.0, 因为之前已经有两个 ACR-NEMA 标准了。DICOM 3.0 就这样诞生了 (虽然根本就没有 DICOM 2.0 和 DICOM 1.0)。基于这个原因, 3.0 经常被省略, 一般都被大家习惯称作 DICOM。

ACR-NEMA 对 DICOM

几年之后, 我承担 PACS 项目时, 要将 GE 1.3 Advantage 工作站连接到中心成像服务器。由于工作站年事已高, 不符合 DICOM 3.0。而是使用“史前”的 ACR-NEMA 2.0 标准。然而, 除了一些小的不顺利以外, 该工作站是能够连接 3.0 服务器并推送图像的。我们感觉自己非常幸运, 事实上我们的实验恰恰证明了一个重要的事实: DICOM 3.0 从较早的 ACR-NEMA 前辈那里继承了足够多的架构, 因此我的这个项目才有可能完成。此外, 组织 DICOM 3.0 的工作流程时没必要把 DICOM 3.0 之前的装置排除在外。

某种意义上, 用一个 ACR-NEMA 2.0 装置, 比用一个没有医疗数字化的设备强多了。此外, 一些厂商可能仍然提供软件升级和打补丁服务, 可以实现将老旧的 ACR-NEMA 装置升级到 DICOM 3.0 (当然是收费的)。尽管如此, 也千万不要为了节省预算而买一些有 15 年工龄的旧的 MR 装置, 而不是去买个新的。劝你还是断了这种只为了省钱的装置升级和 PACS 连接的想法吧。

时间胶囊

以前我曾经帮助一个著名的PACS公司通过DICOM连接一个古老的MR。同平时一样，这个任务“很简单”：我们需要进行设置，使一个MR与我们提供的新DICOM服务器建立连接。经过长时间的查找，MR的所有者找到了设备的支持团队，我们约定与MR的实施工程师在上午9点见面，然后5分钟内解决战斗。

工程师准时到达，跟MR战斗了数小时，然后彻底崩溃了。客观事实是，这个工程师是在厂商停止提供MR相关服务后才被雇佣并接受培训的。所以他马上找来了另一个资深专家，于是之前的故事又被重复了一遍。几个小时的电话求助后，第三个工程师出现了，他在这行干的足够久总算知道如何登陆到MR的配置界面了。在一系列的头脑风暴之后，终于把MR配好了。整个配置只用了5分钟，但是寻找方法为我们带来了一天的精神和身体上的摧残。

这个故事的启示是：当心老设备！许多情况下，能够了解设备应该怎么使用的人有可能比设备本身还更稀有，更有价值（当然也可能更贵）。

DICOM 3.0永远不会被DICOM 4.0替代，而是每年由DICOM设计工作组进行评估，并发布更新过的DICOM 3.0以及新的补充内容。标准修订版的卷名为PS3.X-YYYY，其中3是标准编号，X为卷编号，YYYY为该版本的年份。比如PS3.5-2008表示DICOM 3的第五部分且为2008修订版，它是PS3.5版本的接替者。然而，每个新的修订版仍然参考DICOM 3.0标准。真可称得上是“万变不离其宗”了。

命名方法如此接近，会导致修订版和标准编号之间的混乱。曾经我问起过我们公司研发的软件：“你支持DICOM 2003吗？”其实，根本就没有DICOM 2003，只有DICOM 3.0的2003版。然而相似的提问是非常普遍的，也会出现在相同的DICOM演化背景下：

1. 你所需要处理的所有DICOM设备本质上是它被开发时使用的DICOM版本的快照。设备之间会完全不一样。
2. DICOM装置必须在一起工作。因此保持新DICOM设备向下兼容之前的型号比追逐最新的DICOM功能还重要。由于这个原因，DICOM厂商并不因为出了最新的DICOM版本而兴奋，他们当然也不急于用更快、更新的DICOM协议去更换较老的。甚至大多数最新生产的DICOM装置还仍然具有90年代中期的功能呢。
3. 如果你不得不在一个复杂的环境中实现一个DICOM方案，一定要选择最合适的而不是最华丽的，这两者可不是一回事。比如，新的DICOM版本采用了许多先进的图像压缩协议，允许设备更高效地传输和存储医学图像。然而，在DICOM世界中设备一定要成对工作，传输新压缩协议的图像给另一个老设备，老设备是根本无法接收的，因为这个设备只支持基本的、非压缩的图像格式。

这并不意味着你只能一直用老设备；相反，毋庸置疑，你应该认真考虑去购买一个新的装置！但是如果你仍然有一些较老的设备，你应该优先考虑这些设备需要向下兼容的问题。

DICOM装置兼容性的层次总会在DICOM一致性声明中体现出来（可能已经指出或提到过很多）。DICOM设备能够支持许多可选的和半可选的特性，并且更广泛的这些特点值得成千上万的新的DICOM设备接入进来。

ACR-NEMA 2.0

虽然ACR-NEMA 2.0是在十多年前被事实上是翻新而成的DICOM 3.0正式代替，但是许多医疗实践仍然使用它，并且一些DICOM软件厂商仍然保留了对ACR-NEMA 2.0的兼容，以保证能够接受老式的数据。将老的或私有的格式转化为DICOM 3.0已经成为了单独的一门生意，这都要归功于DICOM软件公司们的努力。

所有DICOM兼容性问题最终都可以通过查阅当前版本的标准进行了解。如前面提到的，NEMA在其官网（[http:// medical.nema.org](http://medical.nema.org)）维护最新版本的DICOM 3.0版本，你可以从哪里免费下载或订购纸质版。

介绍还满意吗？下面让我们一起来看看DICOM是如何工作的吧。

第二部分

DICOM 和临床数据

只要你需要实现医疗工作流程，你就要面对两个重要任务中的一个：收集临床数据和处理临床数据。DICOM 可以在这两个方面帮助你，为你提供一个一致的、符合临床需要的合理方法。为了在数据如何表示方面保值一致性并消除模棱两可，标准使用一系列数据表示和编码的标准规则。本书第二部分就是为了向您介绍这些规则。

许多情况下，这些内容听起来像来自 DICOM 星球的大眼机器人说的外星语。人类不会觉得这些语言美妙，但是这些却另 DICOM 对话更加轻松惬意。它的长寿恰恰证明了它的价值：从 80 年代中期 DICOM 刚刚出现起，医疗和电脑工业已经经历过大革命时期的日新月异，然而我们现在仍在使用它，并且时常升级的词汇促使它还将被大家使用很多年。

了解基本 DICOM 语言是我们进入 DICOM 世界冒险的不可缺少的本领，因为大眼机器人可是会喜怒无常的。像学习其他语言一样，DICOM 学习也要从字典和语法介绍开始。

第5章 DICOM话，你会说吗？

在本书中（第2章），我曾经简短地触及了DICOM数据表示。DICOM将所有现实世界的数
据一段段地切分成了标准的属性（详见DICOM数据字典）并且用这些属性的选集来描述现实
中的对象，比如IOD（信息对象定义）。在本章，我们将细致地介绍这一过程。

5.1 IT训练营

由于DICOM全都是数字的，因此我们一起来复习一下计算机基础知识吧。在我们日常生活中，我们使用十进制系统时：我们会数到10，而且我们有10个数字来表示他们0-9。计算机数据是通过二进制存储和处理的。二进制，或二进制系统，其实是通过0和1两个数字表达所有数值的。一个比特相当于二进制系统中的一个数字，因此每个比特只能是0或1两个值中的一个。

一个字节就是8个比特（8位二进制数）。如果你写出8位二进制数的所有组合，你会得到256（ $2^8=256$ ）个二进制数：00000000，00000001，00000010，00000011，00000100 …… 11111110，11111111。换句话说，1个字节能够储存从0到255个数值。任何计算机硬件（硬件驱动器，随机存储器，闪存等）都用字节存储、读取和记录二进制数据。比如，存储13比特的数据，一个计算机必须指派整2字节的内存。普通显示器（或显卡）表示一种基色（红、绿、蓝）的256个色阶都只使用1字节。所以，在放射领域中灰度色阶也只用一个字节来表示。这就意味着你可以在普通显示器上使用256个灰度色阶。然而，特殊的放射专用显示器和硬件突破了这种限制，因为其可以指派更多的字节给显示灰阶使用。

一个字节也足够存储所有拉丁字符（小写、大写、标点符号），因此一个字节通常由单字母单元（半角）显示。比如，存储12个字母，一个计算机使用12字节的内存；1个字节存储一个字母。

大数据卷，比如图像，会需要上百万的字节进行存储。因此，二进制系统用更大的数字计算字节： $2^{10}=1024$ 字节相当于1千字节（kilobyte (KB)），1024 KB意味着1兆字节（megabyte (MB)），以此类推。表1中总结了这种计算方法。

表1 字节的倍数

前缀	名称	二进制含义	十进制含义	二者大小差异
K	Kilo (KB)	$2^{10} = 1024^1$	$10^3 = 1000^1$	2.40%

M	Mega (MB)	$2^{20} = 1024^2$	$10^6 = 1000^2$	4.86%
G	Giga (GB)	$2^{30} = 1024^3$	$10^9 = 1000^3$	7.37%
T	Tera (TB)	$2^{40} = 1024^4$	$10^{12} = 1000^4$	9.95%
P	Peta (PB)	$2^{50} = 1024^5$	$10^{15} = 1000^5$	12.59%

事实上，字节的倍数有两种略微不同的计算方法。信息技术（和DICOM）使用 $2^{10}=1024$ 的倍数，这可以从前面所讲的二进制系统的模式中清晰地看到。硬件厂商在使用千（kilo）、兆（mega）的时候使用的不是二进制，而是十进制，即1000的倍数。当他们卖给你一个1GB的闪存时，闪存只能保存 $10^9=1000^3$ 字节（十进制GB），从表1中可以看到比 $2^{30} = 1024^3$ （二进制GB）少了7.37%。看来还是卖硬件更赚钱啊！

当小写“b”用在Kb、Mb、Gb时，b指的是“比特（bit）”而不是“字节（byte）”。网络方面：网络带宽（和计算机存储不一样）通常用“千比特每秒”（Kbs）和“兆比特每秒”（Mbs）来测量，用来表示一秒内网络能够完美传送多少的数据。比如，拨号网络速度最高为56Kbs，相当于 $56/8 = 7\text{KB}$ 每秒（KBs）。一个标准计算机T1线路带宽为1.544Mbs；DSL是10Mbs；一个好的PACS网络需要在1-10Gbs的带宽上运行。

十六进制是2个连续字节的速记法，可以表示65536（ $256^2=65536$ ）个数值。由于他们是16比特（2字节），十六进制数字能够使用一个基于0-9、A-F的16个基本数字的计数系统进行书写；其中A-F分别表示10-15。这很像我们所使用的进制数字，不同的是我们需要16个字符来表示十六进制数字（我们增加A-F作为新的数字）。为了区别于文本和十进制数字，我们用0x的前缀进行十六进制数字的标记。

你如何读取十六进制数字呢，比如0x007F？用数字乘以16的幂（就像表示十进制数字一样）。因此，0x007F应该表示为：

$$0 \times 16^3 + 0 \times 16^2 + 7 \times 16^1 + F \times 16^0 = 7 \times 16 + 15 \times 1 = 127$$

这就是十进制数字。许多当前的计算器（包括Windows系统中的）提供二进制到十进制再到十六进制的转换功能。并且能够用A-F的按键来输入十六进制数字。

在十进制系统中，我们有10个数字，2个数字能够表示 $10^2=100$ 个可能的数值：从00、01、02一直到99。相似地在十六进制中有16个数字（0-9、A-F），从0x00到0xFF的2位十六进制数字，能够表示 $16^2=256$ 个数字。但如我们所知，256个值的范围正好是一个字节。比如，十六进制0x7F表示一个字节，而十六进制0x007F则表示两个字节：00和7F。这就是为什么十六进制数字中有那么多的零：零并不影响数值（ $0x7F = 0x007F = 127$ ），但是它告诉计算机存储设备需要多少空间储存这些数据。

DICOM标准同样使用0x前缀或H后缀，来区分十六进制数字和十进制数字。比如，数字12在DICOM世界中相当于十进制的12，如果是12H或0x12则代表是十六进制的12或十进制的 $1 \times 16 + 2 \times 1 = 18$ 。H和0x在书中只有需要的时候才出现：在默认十六进制的环境下一般就省略前后缀了，在容易混淆时则会将其加上。以零开头或包含A-F这样的数字也说明这是十六进制。通常情况下我们应该记住，在DICOM中，所有相近的数字数据都是用十六进制存储的。

5.2 文本 对 二进制

对于数据格式而言，任何数据都能以文本形式或者二进制形式存储。文本格式多用在姓名、日期、ID 等文本字符串中。二进制格式则用在单一数值或数值序列（图像像素等）编码中。之前所提到过，在更加协议化、面向计算机的表达方式中，二进制格式有存储数字的先天优势，因此，数字数据用二进制格式存储是一个自然选择的结果。而另一方面，二进制数据编码具有严重的不足：它由计算机硬件决定。

不同的计算机系统使用不同的字节顺序来表述相同的多字节数字。一些系统记录数字从最低有效字节开始（小端字节序 Little Endian order），其他的系统记录同样的数字从最高有效字节开始（大端字节序 Big Endian order）。比如，小端字节序计算机（MS Windows OS）使用 0x007F 存储两个字节，而大端字节序计算机则（Mac OS）使用 0x7F00 记录这两个字节。当一些数据在不同顺序的两种系统间传送时（当你从 Mac OS 发送一个 DICOM 影像给 Windows OS 工作站进行存储时），字节序类型必须转换正确（因为二者的字节顺序是颠倒），否则数字会被读反，这会导致数值完全错误：0x007F = 127 而 0x7F00 = 32,512。为了避免数据传输错误，DICOM 应用之间总是校验字节序类型，并且任意两个相连的 DICOM 装置在初始化网络握手期间都会达成字节序类型的共识。为了让这种共识长期有效，DICOM 采用小端字节序作为它的默认字节序类型。这意味着所有 DICOM 应用，无论运行在什么系统或硬件上，必须能够理解和处理小端字节序。

大端/小端字节序计算机的争论更多的是针对计算机硬件的历史，而在技术上的实际贡献就少多了。如果你去参与任何 DICOM 开发，那么处理不同的字节序类型将是你实现产品跨平台兼容性的一个重要职责。如果你是一个 DICOM 用户或管理员，请记住字节序问题，当你连接不同字节序类型设备（如 PC 何 Mac）时一定要小心。

相比数字形式的数值，文本数据的每个字母都是独立存储的，因此无论是什么硬件总是保持相同的顺序。考虑到数据类型，DICOM 既使用文本格式又使用二进制格式。如果你在文字处理软件中打开 DICOM 文件，你会发现它是由含有某种意义的文本字符串和完全读不懂的符号混合而成的，而后面的部分仅仅就是二进制编码的 DICOM 数据片段了。

图 5 是屏幕截图，显示了一个 ACR-NEMA 2.0 文件的片段，是在写字板中打开的。你总是需要特别的 DICOM 软件来读取和阐述 DICOM 数据中到底存了什么。这就是 DICOM 和现在最新的、同样成系统的标准（比如 HTML 或 XML）的区别，这些文件能够用任何文本编辑器来读取和修改。

以上就是我们早该仔细看看的内容——DICOM 文本和二进制数据是如何格式化的。

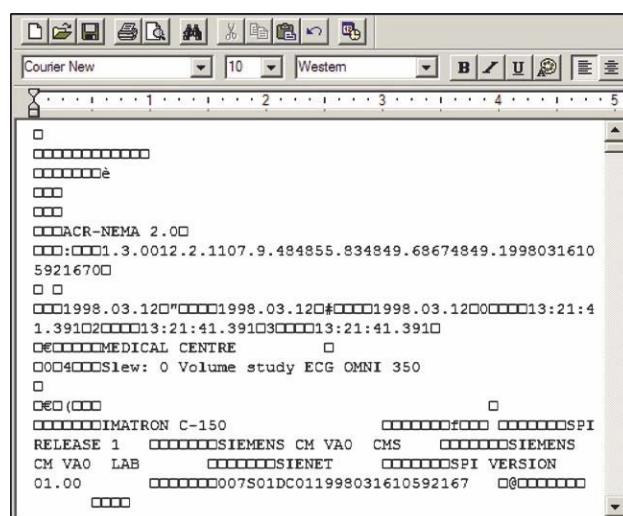


图 5 ACR-NEMA 看上去的样子

5.3 DICOM语法：值表现（Value Representations）

临床数据会有许多格式，距离用毫米；时间用秒；病人姓名用文字字符等等。DICOM 是如何处理众多格式的呢？DICOM 标准在 PS 3.5 中定义了 27 个基本数据类型，就是所谓的值表现（VR）。值表现是用来封装所有可能的临床数据类型的。在 DICOM 中写任何东西必须符合 27 个类型中的一个。每个 VR 都有他自己两个字母的缩写；表示内容的定义；数据中允许出现的字母描述；以及规定的长度。

在表 2 中，DICOM VR 依据他们数据格式和复杂程度排序。许多 DICOM 问题和难题的根源通常出自 VR。一定要重视表 2，当我们讨论一些涉及 VR 的重要问题时，可能还要时不时地返回来参考这个表格。

表 2 VR 定义

VR 名，缩写和全名	VR 内容的定义	允许的字符	数据（值）的字符长度
文本			
CS Code String 代码字符串	开头结尾可以有意义的空格的字符串，比如 “CD123_4”	大写字母 ,0-9 ,空格 ,以及下划线字符	最多 16 个字符
SH Short String 短字符串	短字符串，比如：电话号码，ID 等		最多 16 个字符
LO Long String 长字符串	一个字符串，可能在开头、结尾填有空格。比如 “Introduction to DICOM”		最多 64 个字符
ST Short Text 短文本	可能包含一个或多个段落的字符串。		最多 1024 个字符
LT Long Text 长文本	可能包含一个或多个段落的字符串，LO 相同，但可以更长。		最多 10240 个字符
UT Unlimited Text 无限制文本	包含一个或多个段落的字符串 ,与 LT 类似		最多 4294967294 个字符 (2 ³² -2)

VR 名，缩写和全名	VR 内容的定义	允许的字符	数据（值）的字符长度
设备命名、人和实例			
AE Application Entity 应用实体	标识一个设备的名称的字符串，开头和结尾可以有无意义的字符。比如 “MyPC01”		最多 16 个字符
PN Person Name 病人姓名	有插入符号(^)作为姓名分隔符的病人姓名。比如 “ SMITH^JOHN ” “Morrison- Jones^Susan^^^Ph.D, Chief Executive Officer”		最多 64 个字符
UI Unique Identifier (UID) 唯一标识符	一个用作唯一标识各类项目的包含 UID 的字符串。比如 “1.2.840.10008.1.1”	0-9 和半角句号 (.)	最多 64 个字符
日期和时间			
DA Date 日期	格式为 YYYYMMDD 的字符串；YYYY 代表年；MM 代表月；DD 代表日。比如 “20050822” 表示 2005 年 8 月 22 日	0-9	8 个字符
TM Time 时间	格式为 HHMMSS 的字符串。FRAC ； HH 表示小时（范围 “00” - “23” ）； MM 表示分钟（范围 “00” - “59” ）；而 FRAC 包含秒的小数部分，即百万分之一秒。比如 “183200.00” 表示下午 6:32	0-9 和半角句号 (.)	最多 16 个字符
DT Date Time 日期时间	格式为 YYYYMMDDHHMMSS.FFFFFFFF，串联的日期时间字符串。字符串的各部分从左至右是：年 YYYY；月 MM；日 DD；小时 HH；分钟 MM；秒 SS；秒的小数 FFFFFFFF。比如 20050812183000.00” 表示 2005 年 8 月 12 日下午 18 点 30 分 00 秒	0-9，加号，减号和半角句号	最多 26 个字符
AS Age String 年龄字符串	符合以下格式的字符串：nnnD, nnnW, nnnM, nnnY；其中 nnn 对于 D 来说表示天数，对于 W 来说表示周数，对于 M 来说表示月数，对于 Y 来说表示岁数。比如 “018M” 表示他的年龄是 18 个月	0-9, D, W,M, Y	4 个字符

VR 名，缩写和全名	VR 内容的定义	允许的字符	数据（值）的字符长度
文本格式的数字			
IS Integer String 整型字符串	表示一个整型数字的字符串。比如“-1234567”	0-9，加号（+），减号（-）	最多 12 个字符
DS Decimal String 小数字符串	表示定点小数和浮点小数。比如“12345.67”，“-5.0e3”	0-9，加号（+），减号（-），E，e 和半角句号（.）	最多 16 个字符
二进制格式的数字（与文本格式中的数字一直，但是以二进制存储）			
SS Signed Short 有符号短型	符号型二进制整数，长度 16 比特		2 个字符
US Unsigned Short 无符号短型	无符号二进制整数，长度 16 比特		2 个字符
SL Signed Long 有符号长型	有符号二进制整数		4 个字符
UL Unsigned Long 无符号长型	无符号二进制整数，长度 32 比特		4 个字符
AT Attribute Tag 属性标签	16 比特无符号整数的有序对，数据元素的标签		4 个字符
FL Floating Point Single 单精度浮点	单精度二进制浮点数字		4 个字符
FD Floating Point Double 双精度浮点	双精度二进制浮点数字		8 个字符
OB Other Byte String 其他字节字符串	字节的字符串（“其他”表示没有在 VR 中定义的内容）		
OW Other Word String 其他单词字符串	16 比特（2 字节）单词字符串		
OF Other Float String 其他浮点字符串	32 比特（4 个字节）浮点单词字符串		
其他			
SQ Sequence of Items 条目序列	条目的序列		
UN Unknown 未知	字节的字符串，其中内容的编码方式是未知的。		

5.3.1 VR 长度

VR，如我们所知，定义了 DICOM 数据类型，其中数据大小（VR 长度）是定义中的一个

非常重要的部分。DICOM 用两种方法来了解数据大小。首先，正如我们马上在 5.5.1 中看到的那样，DICOM 总是在记录数据大小时捎带着数据值；这就是 DICOM 了解每个数据元素起始和结尾位置的方法。其次，对于一些 VR，他们的数据长度总是固定的和有限的，正如你在表 2 中最后一列看到的一样。

VR 定义对于数据长度的强制限制，经常在医学成像软件中被忽视，这很容易导致 DICOM 兼容程度不一致的程序之间出现不兼容的情况。如果你正在进行 DICOM 研发，请确保你的产品有足够的 DICOM 兼容度，比如将所有 VR 数据以正确的大小格式化，并且要让它精明到能理解另一个 DICOM 应用发来的大小不正确的 VR。

DICOM 软件开发者要关注 VR 长度的另一个原因是二进制（数字）VR。就大、小端字节序（见 5.2）来说，不同的计算机系统为基本的数字数据准备了大小不同的存储空间。因此，DICOM 中的 VR 大小说明也是为了针对计算机硬件和软件设计的不同来保护 DICOM 数据。

是否使用固定长度？所有 DICOM 数据元素都应该为偶数长度，他们必须是偶数字符（如果是文本）或字节（如果是二进制数字）。为了保证这一点，DICOM 在奇数大小的字符串（比如短文本，ST）后追加空格；或者在奇数二进制数字串后面追加一个空 NULL 的字节，这样就能保证长度为偶数了。比如，姓名为“Smith^Joe”将总是被 DICOM 在内部保存为“Smith^Joe ”，尾部增加的空格使整个字符串长度变为 10。

一方面，DICOM 数据要求凑偶数长度的做法是有优点的。作为奇偶校验它可以来验证数据串的有效性，可以认为所有奇数的内容都是不正确的。然而，这种方法也确实太古老了，并且优点是有代价的，代价就是所有 DICOM 软件都需要增加或截断没有实际意义的结尾空格。此外，由于 DICOM 应用程序总是需要与其他软件（比如用在 PACS 归档的数据库；放射科信息系统（RIS））协作，因此忽视或者去除结尾空格的问题会在整个流程链条中蔓延。最安全的方式是在 DICOM 数据被解码后，并且在尚未发送到非 DICOM 应用程序之前就处理好 DICOM 数据。当截取空格的问题被忽视，存在 DICOM 中的“Smith^Joe”就会按照“Smith^Joe ”被提取出来。很多人可能觉得看起来没什么区别，但是这可能被那些不了解 DICOM 凑偶长度问题的软件错误理解成其他姓名（甚至其他病人）。

5.3.2 字符：外国字儿和通配符

显而易见，不同国家的用户会希望在本国的语言环境下使用 DICOM 数据。DICOM 是具有这种本地化支持的。但是，语言的选择（对操作系统本地特性的设置）会影响到 VR 数据类型中字母和字符的选择。DICOM 将字符的选择称为“字符系统（character repertoire）”。

最新的 DICOM 设备都会使用拉丁字母表（相当于默认 DICOM 字符系统，依据“ISO IR-6”）。这是 DICOM 标准在创建时最先应用的字符集，并且这个字符集适合许多他国语言。当字符集不支持某国语言时，常用的处理方法是使用音译的方式来表示非拉丁字符集。如果字符读音类似，那么就通过附加的软件补丁将该国字符映射为拉丁字符，以此避免更改 DICOM 中的默认拉丁字符集。无论语言相关问题是来自于 DICOM 标准还是来自其他什么地方，这种替换字符的方法都是一种常规做法；也是给用户的一种自行处理语言相关问题的方法。

许多国家可能不会使用这种音译的语言处理方法，因为那些国家在临床应用中规定必须使用本国语言。可是这就要求所有 DICOM 厂商都必须支持这些字符，但是这些厂商目前所能提供的语言本地化支持还远远没有达到完美的程度。

界面本地化

曾经有一颗国际研究卫星由于参与国 A 使用毫米而参与国 B 使用英寸，最终导致卫星在外太空消失了。

请记住 DICOM 使用十进制单位：长度用毫米，重量用公斤等。这些单位是 DICOM 标准指定的，并且为了避免歧义，他们会在任何一个 DICOM 界面上出现。

DICOM 日期格式已经成为医疗界面中最常见的绊脚石了。DICOM 标准使用 YYYYMMDD 即“20080201”表示 2008 年 2 月 1 日。然而，最适合内部存储的方式并不是最适合显示的方式。无疑地，“2008 年 2 月 1 日”在任何 DICOM 用户界面都会比“20080201”或“2008.02.01”看起来更顺眼。但当厂商不这样做或做的不对的时候，不同国家的用户将会对日期产生完全不同的解释。比如，我见过为美国人开发的界面这样表示日期“01.02.2008”，当然他们也很可能弄成这样“02/01/2008”。

除了五花八门的语言和格式外，有一些 DICOM 中保留字符是有特定含义的，并且需要按照 DICOM 的规定使用。比如 DICOM 允许使用通配符代替字符串。星号 (*) 通配符表示任何字符序列；问号 (?) 通配符表示任何单个字符；反斜线符号 (\) 表示“或”。例如 DICOM 需要查找 CT 或 MR 的检查，那么就需要用“CT\MR”字符串来查找，这代表“CT 或 MR”设备。一般来说，通配符对用户非常有用。如果你不记得完整的病人姓名，你可以输入几个首字母后面跟着星号 (*)，就能够获取所有相似的病人了。所以，若输入“Smit*”则每个 DICOM 软件都能找到这些匹配项：Smith、Smithson、Smithsonian 等。

然而，通配符容易造成混乱甚至导致 DICOM 错误。DICOM 软件用户可能不小心在 DICOM 报告中键入了通配符，或者在文件名中包含了反斜线符号，这些都有可能最终存入 DICOM 数据元素中。可以猜得到，这些报告和文件名很容易在通配符查询时被误解，最终导致信息错误或丢失。

5.3.3 文本VR: CS, SH, LO, ST, LT, UT

文本 VR (CS, SH, LO, ST, LT, UT, 代码字符串, 短字符串, 长字符串, 短文本, 长文本, 无限制文本) 比较简单：他们的用处就是存文本字符串，可以说是最不用费劲去处理的数据类型，因为使用他们极少需要处理（除了向其他应用程序传送数据时，需要在结尾处理之前所讲的空格以外）。最重要的一定要记住文本 VR 是有长度的。比如，如果 CS 能够存储 16 个字符，那么超过这种限制，小则造成软件错误；大则造成整个 PACS 故障。显然，任何对字符串的长度检查都需要 DICOM 软件来实现，并且所有字符串若想超越限制要么截断字符串要么将 VR 类型转变为长字符串类型。

5.3.4 日期和时间: DA, TM, DT, AS

DA (日期)、TM (时间)、DT (日期时间) 类型是比较一目了然的：它们通过字符串格式存储日期和时间。最重要的问题就是去了解日期和时间的正确排布顺序。同样，较早的 DICOM 版本和 ACR-NEMA 使用的日期和时间格式与当下版本比起来稍有不同，那时是使用半角句号 (.) 和半角冒号 (:) 分隔的；比如，较早版本中的 18:32:00 这个时间串，在当前版本表示为 183200。好的 DICOM 系统开发者会考虑到这些问题，提供这方面的向下兼容性是很重要的。

DA 和 TM 类型的另一个问题是，单一属性 (single attributes) DT 类型总是需要将其分割为 DA 和 TM 两种类型。拿“病人出生日期 (Patient's Birth Time)”属性打个比方，它是一个 DA 类型的属性，很可能会补充上“病人出生时间 (Patient's Birth Time)”属性。那么我建议还是将两个属性合并成一个 DT 类型比较好。将涉及多个属性值的单一属性分割会造成属性

同步方面的问题；当一个属性变化时，其他的属性也需要进行更新。

DA、TM和DT格式还有一个问题就是不支持时区信息，比如你想使用时区，那么DICOM会告诉你“与标准时区同步是在本标准范围之外的。”我同意在90年代这样做，对于那些只在本地运行的系统来说这样做是一件好事，但是那时候所有交互都只发生在一个医院的两台设备之间。现在呢，基于DICOM的远程放射诊疗已经开始扩展到多个国家和大陆了，当病人们不去跑路的时候，DICOM应该帮助病人们，去仔细处理时区信息的问题。

如果你问我为什么DICOM非得要AS这个类型，说实在的我也不清楚。在当前标准中超过2000个属性里面，只有一个属性（“病人年龄（Patient Age）”）使用它。然而更常用的“病人出生日期”属性（使用DA类型）绝对能够通过简单的处理来获得病人年龄，其实所有PACS也正是按这种方法做的。所以，这事要是我说了算，我倾向于从DICOM标准中去掉AS VR。

5.3.6 二进制格式的数字：SS, US, SL, UL, FL, FD, OB, OW, OF, AT

这些类型实际上和IS、DS这类文本数字串差不多，只不过是使用二进制格式存储的。SS、US、SL、UL、FL、FD通常用来表示单个数（有时可能将这些单个数组合在一起使用）。OB、OW、OF则通常用在长数字串上。怎么才能储存一个数字图像的像素序列呢？比如，这个例子中，序列中的每个数都会有相同的字节大小（分别是1、2、4字节），他们将串接成一个长二进制序列。只有一个类型，OB使用长度为1字节的数。其他类型都比1字节要大，因此他们都会受到大小端字节序问题的影响。

最后，AT（属性标签（attribute tag））存储一对两字节的数。这种数据类型是用来为所有DICOM属性存储标签的，我们马上将在5.4中看到他的作用。所以说，AT类型和其他数字类型不一样，他的唯一的作用就是枚举DICOM数据属性。

5.3.7 PN：存储病人名字

PN（病人姓名）VR编码整个病人姓名。很不幸，DICOM使用一个单一字段保存这些值。因此，整个病人姓名（名、姓、中名等（主要指欧美国家的名字））会用一个PN类型VR存储。可想而知，这经常在治疗流程和软件方面造成混乱，因为“John Smith”可能写成“John Smith”，也可能写成“Smith^John”；或者甚至写成“Smith, John”。为了消除这种不确定性，DICOM规定了如下的姓名顺序：

姓^教名^中名^名前缀^名后缀

所有的字段之间都会用补字符（^）分开。这和我们在VR表中的示例是一样的。然而，在纷乱的医疗环境中，这种顺序常遭到改变，导致永久的信息丢失或病人识别错误。

有两个补救的办法：

1. 标识病人，一般都是使用病人ID而不是病人姓名。使用病人ID就不会受到拼写错误影响了。
2. 在PACS或任何DICOM软件中，需要查询（病人）姓名时，最好使用通配符比如星号（*）充当“任何文本”。正如我们所知，通配符查询是DICOM明确规定的。在病人姓名查询框中输入“*Smith”，将一定能返回名字中带有“Smith”字样的所有病人，所以你的病人信息不会搞错或者受到姓名顺序的影响。事实上，一些DICOM程序会在你的查询过程中自动增加通配符，并返回所有看起来差不多的姓名。

此外，某些DICOM程序已经聪明到超越通配符匹配的程度，他们实现了发音相似匹配来查找名字听起来差不多的病人，即使拼写不一样；比如，能够将“Nelson”和“Neilsen”（都叫尼尔森）匹配。这种特性对临床流程非常有利：这么做是在消除错误，而不是在制造麻烦。

5.3.8 AE：命名应用实体（Application Entity）

AE VR是用来表示DICOM应用实体的。AE本质上是DICOM设备和程序的名字，用来唯一区分它们（在你的PACS网络中，绝不允许存在两个一样的AE）的。对于任何DICOM网络或PACS来说，基于他的用途，所以AE VR可谓最总要的VR。

即使DICOM没有对命名AE规定严格的要求，但AE仍然只会使用数字和大写字母，且没有空格、标点或其他字符。事实上，DICOM装置只接受大写字母的AE的事情并不稀奇。这就必须引入一个新的问题——大小写敏感；为了简化输入，通常避免使用大小写敏感的名称。大多数的DICOM设备无法区分“Workstation1”和“WORKSTATION1”，但是某些其他的设备就会比较挑剔了。关于AE名称的选择，还是取决于你，但是成败与否还是在于使用明确的、好理解的应用实体名称。比如这样做是为了明确表达实体功能（CTARCHIVE，CT归档）或地点（MR1FLOOR，一层MR）。AE会在7.1中详细介绍。

5.3.9 UIDs: 唯一标识

UIDs VR是用来对唯一标识进行编码的，唯一标识主要是用来标识特定DICOM数据（对象）实例的。AE只要求在本本地是唯一的（在网络内）；但无论何时何地，DICOM UID必须全球唯一。比如，当你从一个DICOM装置向另一个复制DICOM检查时，第二个装置需要修改该检查UID（Study UID）属性值，这主要是为了表示现在操作的是同一个检查的不同实例（其实就是新建了一个除UID以外信息完全一样的新检查，之所以是新检查，是因为检查UID不一样）。那么，在第二个检查实例中做任何改变（修改检查图像等）都不会和原始图像搞混了。5.5.8会进一步介绍UID的。

5.3.10 SQ: 序列化数据集

SQ VR是对数据集序列的编码，每个数据集可能包含多个数据属性。此VR支持最为复杂的DICOM结构，他允许在VR中嵌套其他VR。这对于将许多相似元素打包成一个单一数据块来说是有好处的，这样可以很好的处理他们，省的把它们弄的到处都是，很难整理归纳。因此，数据块是整个呈现的，不用担心会有单独的块元素。

在DICOM文档中包括一个叫做一致性声明（conformance statement）的东西，它的序列层级是通过大于号（>）来表示的。比如，若你进入一个DICOM数据中，你会发现它很像图6中的样子。在图6的数据表格中，第一行的“参考的检查序列序列（Referenced Series Sequence）”属性附带有两个属性，前面都有个大于号（>），这表示“参考的检查序列序列”属性有一个SQ VR，并且如这个序列所示，其中有“检查序列实例UID（Series Instance UID）”和“参考的实例序列（Referenced Instance Sequence）”。更加有趣的是，“参考的实例序列”属性还跟着两个其他元素，他们前面都用双大于号（>>）表示，这说明“参考的实例序列”内部还有一个序列，并且包含两个属性“参考的SOP类UID（Referenced SOP Class UID）”和“参考的SOP实例UID（Referenced SOP Instance UID）”。这么说来，“参考的检查序列序列”包括四个子元素；来自他自己的第一层序列的元素和来自他子序列的元素。

序列化导致一个复杂，非线性的DICOM数据格式因为一些VR是从另一些中扩展出来的。序列化同样需要更加小心地设计DICOM软件。比如，图6例子中，根元素“参考的检查序列序列”的VR（数据）长度需要正确地计算所有嵌套的元素数据。基于这个或者同样的原因，SQ元素会变得很难实现和处理，并且在软件层面总是会包含执行bug。如果你正在开发DICOM产品，处理SQ元素一定要小心，并且要有节制地使用它们。更多关于SQ数据编码的细节我们会在5.5.5中讨论。

DICOM data fragment: data nesting with sequencing

Attribute Name
Referenced Series Sequence
>Series Instance UID
>Referenced Instance Sequence
>>Referenced SOP Class UID
>>Referenced SOP Instance UID

and its SQ graph:

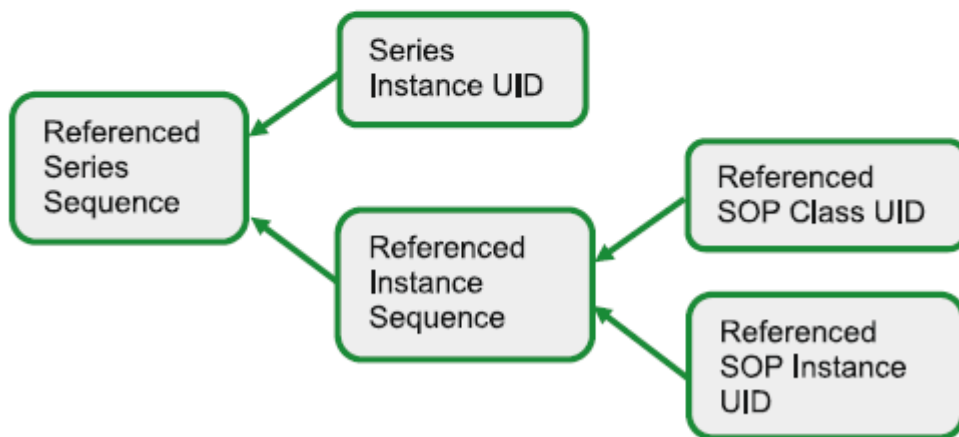


Fig. 6 Nesting DICOM data

5.3.11 UN: 表示未知值

UN VR编码未知值。在许多情况下，UN VR多用在那些无法匹配26个VR的值上。UN最常见的作用是供厂商特定（私有）数据使用，这些数据无论如何都无法让标准去判别了。

我建议每个从事DICOM应用程序开发的人都需要避免使用UN VR。第一，其余的26个VR足够表达几乎所有数据类型了。第二，使用“未知”属性类型仿佛是一种绝望的做法，为什么不使用结构化的数据类型定义呢？大、小端字节重拍序就是一个很好的例子：你不能在UN属性上执行重排序的操作，因为你都不知道这个数据是文本的还是二进制的，甚至也不知道是否真的需要重排序。

最后，VR在DICOM数据结构化方面发挥了无可比拟的作用。他们帮助DICOM与外面世界连接在了一起。VR是DICOM能够理解和沟通的一种构词形式。但是我们如何把现实世界的数项翻译为VR呢？答案就是DICOM数据字典（Data Dictionary）。

5.4

DICOM 数据字典 (Data Dictionary)

完整的 DICOM 数据字典收录在 DICOM 标准的 PS3.6 中, 用来对所有 DICOM 属性进行编码。除了数据字典, DICOM 供应商可以对其私有的数据属性使用它们自己的字典。DICOM 供应商也可以对于私有的数据属性使用它们自己的字典结构。具体规则在本节中会有回顾。

5.4.1

标准的 DICOM 数据字典

说白了, DICOM 数据字典就是在数字医疗方面, 所有标准数据项 (属性) 的注册表。据我们现在所知, 这些项目需要用 27 个 VR 属性进行格式化。

为了把这些超过 2000 个项目按照一定顺序排列, 所有项目首先被分成编号的项目组 (group) (如果项目内容的大概相似就分为一组)。项目组是由单独的元素组合在一起的。因此, 每个项目都有其自己的编号“(项目组, 元素)”, 这就是所谓的元素“标签 (tag)”。所有进行标签的元素都称作“属性 (attribute)”, 或者 DICOM “数据元素 (data element)” 或简称为 DICOM “元素”。

如在 5.1 中所见, 项目组 and 元素都是用十六进制数字编号的。表格 3 中是 DICOM 数据字典的一个摘录。我们可以看到, 第一列中的内容是十六进制“项目组, 元素”标签。第二列“属性 (数据元素) 名称”列, 可算是最重要的部分了, 它描述了元素中所保存的现实世界的的数据是个什么样子。显然,“(项目组, 元素)”标签唯一地对应属性名称, 因此我们可以使用它的标签(0010,0010)或者属性名称 (“Patient Name”)来查询数据元素。这是一个非常重要的 DICOM 应用习惯; (项目组, 元素) 标签更短, 固定长度, 且一定是十六进制, 这些特点更适合在计算机环境下使用它。所有 DICOM 应用程序查询数据元素时, 都使用它们 (项目组, 元素) 的标签而不是使用需要进行描述的属性名称。

如之前章节 (图 7) 看到的那样, 字典中的 VR 列规定了每个数据元素的格式。比如, “病人出生日期”元素(0010,0030)必须使用 DA 格式 (这是一个八位的数据串, YYYYMMDD)。

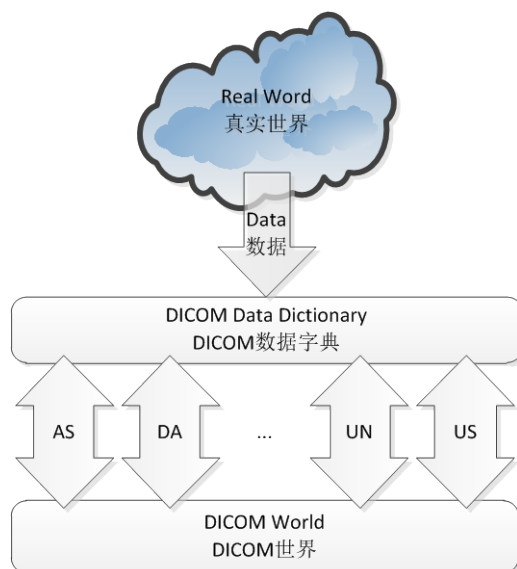


图 7 从真实世界到 DICOM 世界

数据元素值多样性 (VM, Value Multiplicity) 定义了相关元素包含的内容是单值还是多值。比如, “病人其他姓名”元素(0010,1001)能够清晰地展示不止一个姓名, 所以他的多样性中标示的内容是“1-n”, n 表示任何数字。我们如何将许多值放在一个元素中呢? DICOM 将多个值串接起来形成一个存放多值的单值元素。若这些值是二进制的 (有二进制的 VR), 那么

串接工作会很轻松。单值二进制 VR 的长度是已知的和固定的（比如，SS VR 的长度是 2），这些固定长度的内容，能够方便的读取。在另一方面，文本值通常不是固定长度的，因此他们用反斜线(\)作为串接分隔符。比如，如果病人 John Smith 使用其他的名字，比如“Dr Jekyll”和“Mr Hide”，那么(0010,1001)中的内容将会是“Dr^Jekyll\Mr^Hide”，对应的值多样性应该是 n=2。之前我们说过，反斜线(\)在 DICOM 中有特殊的含义，在多值时它是“或”的概念（见 5.3.2）。因此，不要把反斜线(\)用在除此之外的地方（文件名、日期等）因为他可能给你的 DICOM 软件带来混乱。

表 3 DICOM 数据字典中的一些行

(Group,Element) tag (项目组, 元素) 标签	Attribute(data element) name 属性(数据元素) 名称	VR	VM	Retired status 是否退休
(0008,0001)	<i>Length to End</i> <i>到结尾的长度</i>			<i>RET</i>
(0008,0005)	Specific Character Set 特定的字符集	CS	1-n	
...				
(0010,0010)	Patient Name 病人姓名	PN		
(0010,0020)	Patient ID 病人 ID	LO	1	
(0010,0021)	Issuer of Patient ID 病人 ID 创建者	LO	1	
(0010,0030)	Patient's Birth Date 病人出生日期	DA	1	
(0010,0032)	Patient's Birth Time 病人出生时间	TM	1	
(0010,0040)	Patient's Sex 病人性别	CS	1	
...				
(0010,1000)	Other Patient IDs 其他病人 ID	LO	1-n	
(0010,1001)	Other Patient Names 其他病人姓名	PN	1-n	
...				
(FFFE,E00D)	Item Delimitation Item 项目定界项		1	
(FFFE,E0DD)	Sequence Delimitation Item 序列定界项		1	

在 DICOM 数据字典中的“RET”标志表明属性已经退休（弃用），这些是较早前版本标准中的属性，未来 DICOM 将不再支持。这些元素不能被重新定义新的用途，并且它自己的功能已经由其他新的、更好的字典元素所替代。退休的项目在 DICOM 数据字典中使用斜体标出的。客观的说，DICOM 修订版需要经过很多年才会普及到工业产品中，因此 2005 年退

体的项目可能要到 2010 年或者更晚才会在某些 DICOM 装置上使用。事实上, 由于所有 DICOM 厂商更注重对其老的设计保持向下兼容, 所以退休项目会永远放在字典中 (但是会越来越少用到), 但永远不会完全弃用。好的 DICOM 应用程序必须正确地处理这些项目。

手头有了一些数量可观的 VR 和 DICOM 数据字典知识, 我们终于可以开始“说 DICOM”了, 但是我们还只能在非常简单的属性层面聊一聊。比如, 造个句子:

“病人 John Smith, 男性, 剩余 1945 年 8 月 6 日”。

请看表 3 中的数据字典摘录, 我们看到这个句子中有三个数据元素: “病人姓名”(0010,0010)、“病人性别”(0010,0040) 和 “病人出生日期”(0010,0030)。这些元素分别使用 PN、CS 和 DA 三个 VR。因此, 用他们的标签替换了名称, 并且使用 VR 格式, 这句话我们应该这么说:

(0010,0010)Smith^John (0010,0030)19540806 (0010,0040)M

这就是最原始的、经过 DICOM 编码的句子, 简单吧!

5.4.2
私有的 DICOM 数据字典

以前提到过, 标准的 DICOM 数据字典包含 2000 个条目, 这些条目是从医学成像行业仔细收集来的。你可能会预料这种做法对于编码一个不在其中的条目来说机会不太大了。但是这一切要是可行的那又会怎么样呢:

事实上, 需要在 DICOM 编码数据中增加私有 DICOM 属性的各路 DICOM 和 PACS 厂商天天都会这么干。我们可能设计了一些 DICOM 软件, 并且希望以单独的项目存储病人的中名。标准 DICOM 数据字典不包含“病人中名 (Patient’s Middle Name)”属性。DICOM 提供一个非常简单的解决方案。所有双数组编号提供 DICOM 数据字典的供标准数据使用; 所有单数子编号供私有数据使用。

在这个“病人中名”的例子中, 我们能够创建自己的、与标准字典互补 DICOM 数据字典并前存储一个如表 4 中列出的那种条目。我们可能使用一个奇数“私有”组编号比如 0009, 而元素编号, 就随我们挑了。

由于我们补充的字典不是标准的, 其他 DICOM 应用程序可能完全不了解(0009,0010)标签的含义, 但是他们会知道, 这个组使用奇数组号 0009, 因此它是私有标签。按照 DICOM 要求, 不认识的标签应该忽略掉; 因此当其他应用程序读取我们的 DICOM 数据时, 它们会很自然地略过(0009,0010)这个属性的。不幸的是, 事实上没这么简单。其他应用横须可能会有他们自己的私有字典也是用(0009,0010), 里面存着其他元素 (比如, “技师的姓 (Physician’s Last Name)”)。在这种状况下, 我们就进入了私有标签不兼容的一个经典陷阱之中。我们的 (0009,0010)标签将会被另一个 DICOM 提供者接受, 并且内容将被完全误解。在 DICOM 标准的 PS 3.5 (本文 7.8.1) 中提到 DICOM 给出的避免这个问题的方法: 保留确定的标签作为私有创建者元素; 对特定私有字典的制定者进行编码。但是在实际操作时, 这些问题仍然让人感到很棘手。

基于这些原因, 许多好人至少试图追踪那些有名的 DICOM 供应商以及他们的私有字典, 但是这太难了。这些字典很少公布并且总是在修改。然而, 当你打开 DICOM 文件时, 里面可能多半是私有数据项目, 这一点都不罕见。请记住, 私有标签的使用的问题太严重了, 如果有一天你要进行与私有元素有关的工作, 那么请极其小心地对待那些私有元素吧。

表 4 Making private data elements 创建私有数据元素

(Group,Element) tag	Attribute(data element) name	VR	VM
---------------------	------------------------------	----	----

（项目组，元素）标签	属性（数据元素）名称		
(0009,0010)	Patient' s Middle Name	PN	1

5.5

DICOM 对象

还记得 5.4.1 结尾处简单 DICOM 造句吗？我们现在再看一看吧；这可不是个无意义的练习，因为你实际上已经建立了一个 DICOM 数据对象。我们使用以下元素：

“病人 John Smith，男性，生于 1954 年 8 月 6 日”

我们将元素名称换成了他们的标签并且将内容依据 VR 要求的格式进行了编排，变成了：
(0010,0010)Smith^John (0010,0030)19540806 (0010,0040)M

DICOM 对象是 DICOM 模型标准结构中最必不可少的部分。它们是对 DICOM 信息和命令进行编码、传输和存储的实际词汇和句子。所有 DICOM 数据（比如医学图像、命令和报告）总是被打包在 DICOM 对象格式中。在这个格式中，信息通过 DICOM 对象这种载体在各类 DICOM 设备之间通过 DICOM 网络传导，而 DICOM 对象又是通过 DICOM 文件进行保存的。事实上，DICOM 文件可以认为是 DICOM 对象的信息转存。

当我们学习 DICOM 数据字典时，我们了解到 DICOM 将所有现实世界的的数据打散为不可分的原子信息，即所谓的数据元素。这些数据元素是用 27 个现成的 VR 类型进行编码的。一个 DICOM 对象说白了就是一个数据元素的集合——这里没有分离的“DICOM 头（DICOM header）”和“DICOM 图像（DICOM image）”。作为一个数字医学图像，它将会有许多属性，比如图像宽度、高度、颜色（调色板）、图像产生的日期等。所有这些属性都能在标准的 DICOM 数据字典中找到，并且会被翻译为具有标签和数值的 DICOM 数据元素。这些元素的序列，被用来从整体上描述整个图像。这些元素的序列就是这个数字医学图像的 DICOM 对象了。

然而，DICOM 对象可以变得比简单的元素序列更加复杂。之前学习 VR 时，我提到过一个特殊的 VR 是用来提供序列的，叫做 SQ。SQ VR 是用来保存数据元素集序列情况的，每个数据元素集实际上就是一个 DICOM 对象。这些 DICOM 对象也能够包含其他的 SQ VR，也就是说 DICOM 对象中能够包含一组其他的 DICOM 对象。这种 DICOM 对象的递归或嵌套产生了更加复杂的树形结构，使 DICOM 对象看起来像个数据树，其中 DICOM 对象是枝干而数据元素是树叶（图 8）。

DICOM 是如何写出所有复杂的分支数据的呢？它使用了一个最简单的编码规则，下面我们来回顾一下。

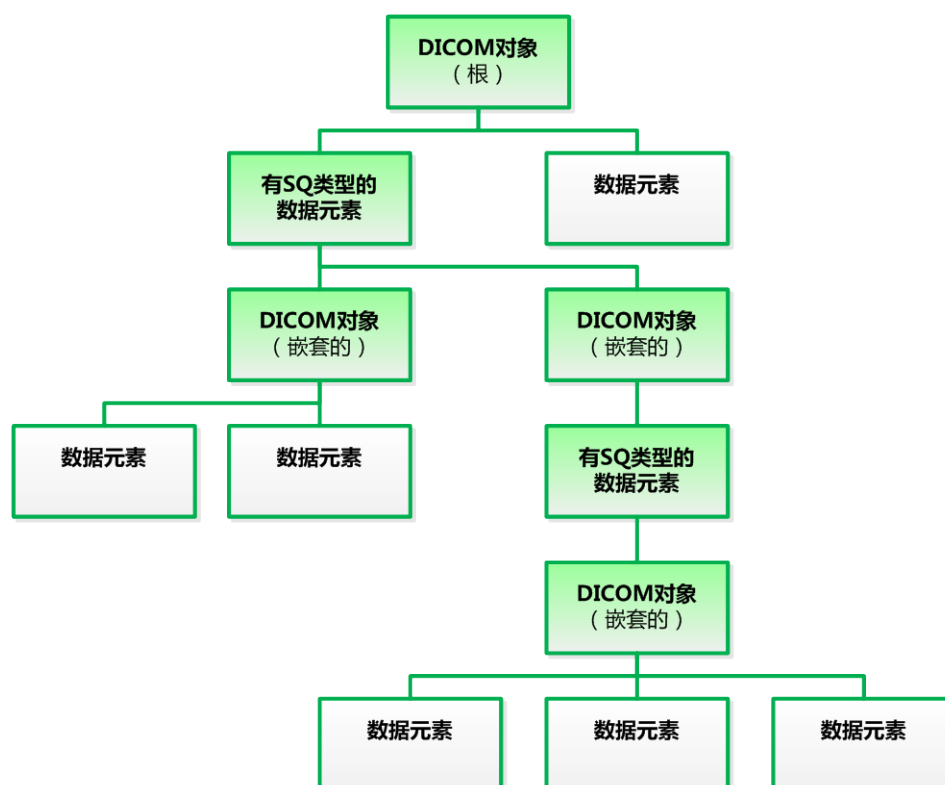


图 8 DICOM 对象嵌套

5.5.1

编码数据元素

用 DICOM 编码就是用 DICOM 规定的格式来书写属性数据，将较复杂的 DICOM 属性值转化为字节的序列。为了描写整个的数据对象，所以必须了解如何编码单独的数据元素。DICOM 标准的 PS3.5 中定义了两个主要的编码类型：隐式 VR 编码（implicit VR encoding）和显式 VR 编码（explicit VR encoding）。隐式 VR 编码是最简单的一种编码类型，并且是 DICOM 默认的编码类型，在表 5 中有详细定义。

举个例子，还是他吧，我们最钟爱的病人——Joe Smith。在 DICOM 数据字典中病人姓名条目的组编号是 0x0010，而元素编号是 0x0010。“Smith^Joe”字符串的原始数值长度是九，但是 DICOM 需要凑偶（VR 长度必须是偶数，参见 5.3.1），因此需要在结尾加一个空格，将姓名字符串变成了“Smith^Joe ”。姓名长度现在变成了 $L = 10 = 0x0A$ 个字符，病人姓名属性的编码情况参看表 6。需要强调一个重要的细节：DICOM 的默认字节序是小端字节序（见 5.2），这意味着多字节数字是从低字节开始写入的。因此组号 0010，最低（最右）的字节 10 在前面，最高字节 00 在后面，“元素”和“长度”编码也采取这种方式。在“二进制”那一行中的 18 字节真实再现了究竟 DICOM 是如何将 Joe Smith 这个病人姓名进行编码的。

对多重数据元素编码（这是种常见情况），DICOM 会将各自独立的编码简单地连接起来，组成一个二进制串；我们将会在第 7 章看到一个非常复杂的例子。由于每个数据项的长度是依据编码内容而定的：“组”、“元素”、值长度都是确定大小的，因此你可以方便地将串接在一起的元素一个个地分解。比如，表 6 中所示的隐式字节序编码，读取前 $2 + 2 = 4$ 字节可以得到“（组，元素）”标签，继续读取后 4 个字节可以得到值长度为 L 的数值，再往后读取 L 个字节得到的就是元素值了。在这之后就是下个元素的起始点，而整个读取过程将和读取上一个元素一样，一次次地重复下去。

显式数据编码和隐式编码很相似。有两种图表类型。其中第一种是用来编码除 OB、OW、OF、SQ、UT 和 UN 以外所有 VR 类型的，详见表 7。

使用显式 VR 编码时，需要在编码中包含两个 VR 类型的字符（参见表 2）。在隐式编码中使用的 4 字节 VR 长度字段在此处被劈成了两半，2 字节存 VR 类型，2 字节存 VR 长度。因此病人姓名编码的例子会变成表 8 中所示的样子。

对于 OB、OW、OF、SQ、UT 和 UN 这些 VR 类型，显式编码会采用些比较不同的方法。两字节的保留字段（值设置为 0000）跟在 VR 类型后面，并且在这之后分派 4 字节来存储 VR 长度（就像隐式编码一样），详见表 9。

比如，使用 OB 类型为一个 256×256 的图像编码像素缓存（每个像素使用 1 字节，全部 $256 \times 256 = 65536 = 0x00010000$ 字节），我们当然可以使用表 10 中所示的方法。

Table 5 隱式 VR 编码

标签		值长度	值
组编号 (2 字节 无符号整型)	元素编号 (2 字节 无符号整型)	4 字节整型 L	偶数长度为 L 的字节, 包含数据元素值
2 bytes	2 bytes	4 bytes	L bytes

Table 6 隱式编码实例

[illegible]

Table 9 显式 VR 编码（用于 OB、OW、OF、SQ、UT、UN）

标签		VR		值长度	值
组编号 (2 字节 无符号整型)	元素编号 (2 字节 无符号整型)	VR (两字符) 如 OB、OW、OF、SQ、UT、UN	保留字段 (2 字节) 预设值为 0000	4 字节整型长度为 L	偶数长度为 L 的字节，包含数据元素值
2bytes	2bytes	2bytes	2bytes	4 bytes	L bytes

Table 10 显式编码实例

字节#	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	...	65547	65548
十进制	224	127	16	0	'O'	'B'	0	0	0	0	1	0	0	3	0	...	10	10
二进制	E0	7F	10	00	4F	42	00	00	00	00	01	00	00	3	00	...	0A	0A
	g = 7FE0		e = 0010		VR 类型		保留字段		VR 长度 L= 0x00010000				VR 值 (像素，每个像素使用 1 字节，L 个字节)					

一定不能将隐式和显式 VR 编码混用在同一个 DICOM 对象中；必须坚持使用同一种编码方式。虽然显式编码看起来有些冗余，但是它在许多情况下很有优势。

1. VR 名称的冗余有利于避免数据编码错误。
2. 随着 DICOM 的演化，一些 VR 类型会发生改变并且可能与当前版本 DICOM 字典中的样式不一致。这种情况下，显式编码保留了原始类型名称并且额外提供向下兼容。
3. 有些非标准的 VR 在标准的 DICOM 数据字典中是找不到的。显式编码对于这种非标准 VR 编码来说是非常重要的。

然而，DICOM 规定隐式 VR 使用小端字节序作为默认数据编码方法。由于显式和隐式 VR 编码不能混用，因此使用哪种编码技术的问题必须在数据传输之前就敲定；DICOM 应用程序在交换任何数据之前都能够进行编码类型的商定。我们会在 9.4 中学习 DICOM 传输语法时看到更多关于数据交换方面的内容。

显式、隐式转换

通常 DICOM 数据不得不依据当下所需的数据传输类型，从隐式 VR 转换成显式 VR，然后再转换回来。显然，从显式转换为隐式比较容易：只用干掉 VR 类型字段就搞定了。反向转换就实在是太困难了：你需要在 DICOM 数据字典中找到相应的属性标签以确定属性的 VR 类型。许多情况下这根本不可能实现。例如，如果属性是私有的（组编号为奇数）根本无法在标准的 DICOM 数据字典中找到。当你迟早邂逅了 UN VR 时，用 UN 来替代所有无法识别的 VR 类型将是唯一的解决方案。

无论在什么情况下，DICOM 一直保持数据项长度为偶数，就像我们编码 Joe Smith 姓名时一样。VR 长度在 DICOM 数据读取（解码）中还有另一个重要作用，它有助于略过未知元素。如果你的 DICOM 应用程序遇到它无法理解的元素，它大可以跳过 VR 长度大小的字节去读取下一个元素。最终，当把未知大小的数据用标准的分隔符进行标记后，相当于由 DICOM 提供一个使用未定义长度的选项。这通常出现在 SQ 项目中，所以我们会在之后的 5.5.5 中再来考虑这个问题。

5.5.2

编码数据组

我们已经知道，DICOM 数据字典中的所有 DICOM 元素都是以数据组为单位组织的，并且都用一对（组，元素）来表示。比如，在数据字典中的 0010 组包含的所有元素都与病人（姓名、ID、尺码、大小、年龄等等）有关；0028 组则用来描述图像信息（宽度、高度、比特和颜色深度）；7FE0 组则只由一个元素组成：像素数据。奇数编号的组不会在数据字典中出现，因为他们是留给厂商储存自己私有数据用的。

当 DICOM 数据元素被编码成为 DICOM 对象时，他们会严格遵循（组，元素）标签的顺序形成 DICOM 对象，并且从标签号较小的开始排序。因此，组内元素是按照升序排列的；而对象中的组也是按照升序排列的。比如，元素（0008,0012）可能会记录在元素（008,0014）之前。元素（0010,0010）可能会在他们之后，因为他的组编号更大。

DICOM 开发者：DICOM 中的元素顺序

DICOM 对象中的元素必须按照规定的元素标签升序进行排列，这对于 DICOM 软件设计来说很重要。首先，这是一个主要的验证工具。如果读取 DICOM 对象时，元素（0010,0010）出现在了元素（0008,0012）之前，那么说明 DICOM 对象出现了很严重的问题。最有可能的是数据已经被破坏得面目全非需要拒绝掉。

其次，一直以来习惯于将最基础的数据放在数字较小的组号中，比如病人数据在 0010 组。这可以帮助 DICOM 应用程序在只希望了解基础标签时不用去读取全部的或可能较大的内

容。从顶端读取，可以在刚刚处理数据组之时就停止读取或跳过其他部分。

除了（组，元素）标签顺序以外，就没有什么需要在编码 DICOM 数据组时特别注意的了，但是还有个问题需要注意：组长度标签。比如某个 DICOM 组“gggg”，这个组中最靠前的保留元素(gggg,0000)是用作保存所有之后 gggg 组中数据元素整体长度的。由于所有数据和标签长度在 DICOM 中都是偶数的，所以从(gggg,0000)元素结尾到下一个组开始的长度 L 也是偶数字节。当编码 DICOM 对象时，元素(gggg,0000)可能会用 UL VR 格式记录在每个新 gggg 组的开头，内容是截止到下一个组之前的所有 gggg 组元素的整体长度（见图 9）。

(组,元素)	VR	长度	值
.....（组 0010 之前的数据元素）			
(0010,0000)	UL	4	L 字节
(0010,0010)	PN	10	Smith^John
(0010,0030)	DA	8	19540809
.....（更多的 0010 组元素）			
(0010,4000)	LT	12	无注释
(0012,0000)	UL		
.....（余下的数据元素）			

L 个数据字节

Fig. 9 组长度元素的例子：在 0010 组最前端的元素(0010,0000)中保存的数据值等于 L，L 是 0010 组的数据元素（正好在(0010,0000)元素之后的）编码后的字节总数。(0010,0000)中的字节数不包含在 L 中。

对组长度进行编码的原因与对元素长度进行编码是完全一样的。如果你的应用程序不需要读取组 gggg 的数据，它可能只从标签(gggg,0000)中读取组程度 L，并且立即快进 L 字节来处理下一个组。由于 gggg 中的元素数量可能很大，因此实际上快进能够提高 DICOM 对象的读取（一种技术，称为部分解析（partial parsing））速度。这种方法用来处理私有的、奇数编号的组非常有效。这类组的内容只有他们的厂商能够解释并且其他人都可以忽略这些数据。如果元素 (gggg,0000)提供了组长度，那么跳过这些组就是小菜一碟。

组长度也可以作为一个基本的安全特征来使用。组长度（就像数据的校验和）使人很难在不破坏整个数据对象结构的情况下修改其中的某部分数据。然而，天底下没有免费的午餐，这种方法也有自己的缺点。为了能够完成(gggg,0000)标签，你的软件应用程序必须了解 DICOM 对象内的 gggg 组中所有数据元素的长度 L。因此，就需要了解写在组长度(gggg,0000)之后的所有 gggg 元素的总长度。这意味着所有 gggg 元素必须能够提供他们最终值并且必须在编码(gggg,0000)元素之前就需要完成编码。对 DICOM 软件开发者来说，这意味着两件事情：

1. 百忙之中完成 DICOM 对象几乎是不可能的，因为所有数据必须先收集起来。
2. 修改 DICOM 对象中的任何数据元素（比如，用另一个名字替换病人姓名）不可避免地会影响元素组的长度，于是这种情况下还需要去更新长度。

附加的操作非常完美地证明了我们生活在一个软件开发者的懒惰世界中。因此许多 DICOM 软件会忽略(gggg,0000)标签，或更糟，随便在里面写个错误的值。考虑到这个原因，DICOM 标准定义组长度属性为可选的：你的应用程序不是必须写清楚他。然而，如果其中提供数值的话，他还是能被别人读取的。因此，大部分 DICOM 软件读取(gggg,0000)标签但不使用他们。典型的做法是无论是否需要，软件会处理每一个组中的每一个数据元素。然而，最好的办法还是避免在(gggg,0000)标签中对组长度值进行不一致的书写或不正确的更新。在 DICOM 标准的 PS3.5 中 7.2 章节中有对组长度元素的细致讨论。

5.5.3

实例：元素和组长度

对元素长度编码是表达 DICOM 数据的一个非常重要的方面。让我们来看一个简单的 C-Echo-request (C-Echo-Rq) 的对象，这个对象在 DICOM 标准中的作用是校验 DICOM 网络的连接情况。当我们在 7.2.2 中学习 DICOM 服务时会仔细了解 C-Echo，但是此时此刻我们需要了解这个对象中到底有什么。C-Echo-Rq 中的编码情况可见图 10。这个对象是一个 DICOM 命令对象，这意味着它只包含来自 0000 组的属性并且使用隐式 VR 编码方式（所有 DICOM 命令对象的默认方式）进行编码的。

元素标签	值长度	值	元素长度
(0000,0000)	4	56	
2+2 字节	4 字节	4 字节	=12 字节
(0000,0002)	18	1.2.840.10003.1.1	
2+2 字节	+4 字节	+18 字节	=26 字节
(0000,0100)	2	0030	
2+2 字节	+4 字节	+2 字节	=10 字节
(0000,0110)	2	0001	
2+2 字节	+4 字节	+2 字节	=10 字节
(0000,0800)	2	0101	
2+2 字节	+4 字节	+2 字节	=10 字节

如图 10 中所示的表格，对象只包含 0000 组中的 5 个元素：元素 0000、0002、0100 和 0800。我们在表格中对每个元素长度进行了计算。“元素”标签(组和元素编号，如(0000,0002))总是使用 $2 + 2 = 4$ 个字节，并且数据字段的长度将总是用 4 个字节进行编码。

跟在(0000,0000)后面的四个元素的总长度是 $26 + 10 + 10 + 10 = 56$ 字节。因此，56 将作为 (0000,0000) 元素的值。如果还有其他的组在 0000 组的后面，我们能够跳过(0000,0000)元素结尾之后的 56 字节，然后开始处理下一个组。

5.5.4

对 DICOM 数据对象编码

如果你，我亲爱的读者，能够精通数据元素编码部分，那么你就已经无惧了。DICOM 对象没什么复杂的，其实就是一连串 DICOM 数据元素（见图 11）。

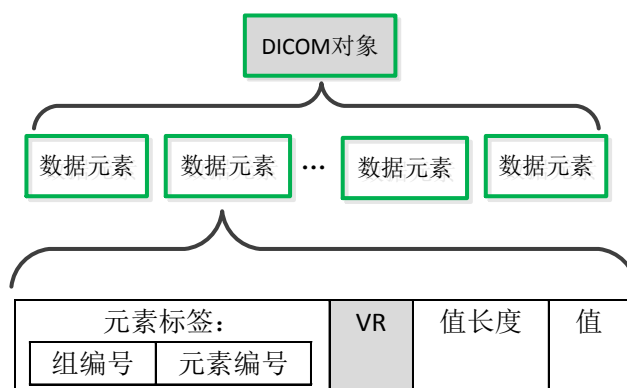


Fig. 11 没有 SQ 元素的对象：一个简单的数据元素序列

我们之前讲过，一个数据元素可能包含一个 SQ 项目，这与嵌套的 DICOM 对象有关。这会增加复杂性，但是不会改变编码的原则。其中，一个规则决定了元素顺序：DICOM 对象内部所有数据元素必须按照（组，元素）标签编号来排序。这种排序方法主要为了实现至少两个实际的目标：

1. 帮助校验数据完整性（就像偶数数据长度所起的作用）。在 DICOM 对象中一个接一个地读取元素时，如果发现读取的下一个数据元素标签编号比前一个还小，那么就说明 DICOM 对象损坏了。
2. 帮助实现将数据按照使用的顺序进行排序。比如，图像宽度和高度的元素标签编号比实际的图像像素缓存要小。这意味着图像宽度和高度是首先被读取的，并且在你读取像素之前就已经知道组织图像像素矩阵应该使用什么样的宽度和高度了。

5.5.5

SQ：对 DICOM 对象序列编码

如之前所学，SQ（序列）这种 VR 序列，在 DICOM 数据编码方面起着非常特殊的作用。它允许我们在一个 SQ VR（见段落 5.3.10）中来存储 DICOM 对象的整个序列。如果对象中介入了 SQ 类型，那么 DICOM 对象的布局会发生戏剧性的变化。SQ 元素本身没有数据，而是存储了一个 DICOM 对象序列。在父 DICOM 对象（这个对象包含 SQ 元素）中这些对象会变成在其内部嵌套的对象。此外由于在 SQ 序列中的 DICOM 对象也服从相同的 DICOM 对象格式，因此他们也可能具有 SQ 元素。所以，我们可能会发现一个像图 12 那样的多重嵌套层级。只有到了被嵌套的 DICOM 对象不再具有 SQ 元素的那一层，嵌套才会停止。这样，对象序列化的整个概念就显得很自然了。如果你还是无法理解，那么想想这本书的结构把。书有章节（数据元素），但是有些第一层的章节内部有第二层章节（子段落），并且一些第二层章节还有他们的子章节（第三层）等等。因此，如果我们认为书是一个很大的 DICOM 对象的话，那么章节和子章节就应该是我们的 SQ 元素。事实上，许多当前数据表现语言（比如 XML）也和 DICOM 一样使用了数据嵌套模型。

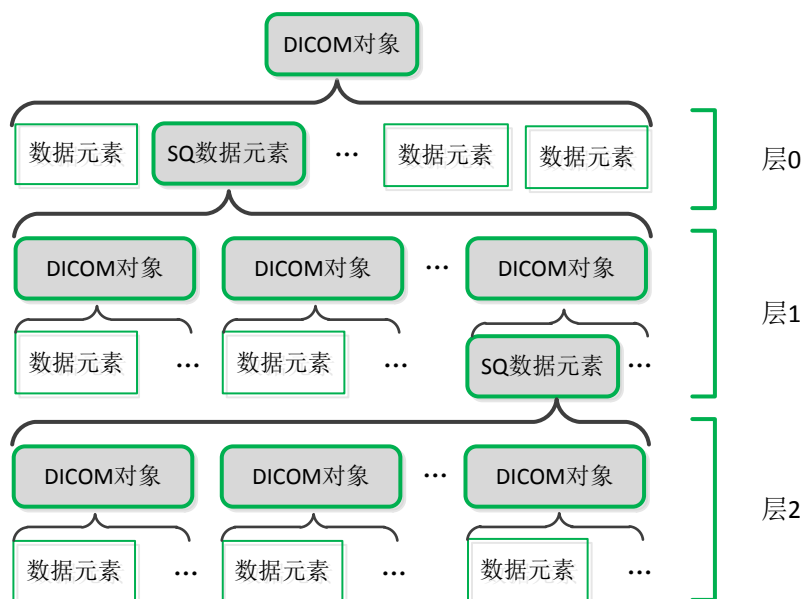


Fig. 12 有 SQ 元素的对象：嵌套着 DICOM 对象

我们如何编码 SQ 元素呢？要是没有 SQ 所有问题都会变得简单了。一个 DICOM 对象是他字节数据元素的列表，所以我们需要按照一个列表来编码它。对于 SQ 元素，DICOM 提供一个特殊的 SQ 编码语法，它能在一个 VR 中囊括 DICOM 对象序列。这很像我们在 5.5.1 中所见到的，整个过程依赖 VR 编码方法（隐式或显式）以及 SQ 数据项目分隔符的选择。表 11 中的三个例子涵盖了所有可能的 SQ 编码方法（嵌套的 DICOM 对象用灰色框标出）。如果你对比三个例子，你自己就会发现 SQ 编码规则。规则是：

1. 在 SQ 序列中的 DICOM 对象是以序列项目的形式进行编码的。
2. 每个在 SQ 序列中的对象项目（object-item）前面都会有个(FFFE, E000)的标签（作为项目分隔符项）。这个项目后面还会跟着以下之一：
 - a) DICOM 项目的显式长度（比如，表 11 的例子 1 中的第一个和第二个项目）。这个长度是用来定义为了获得后面的对象所要读取的字节数。
 - b) 隐式（未定义）长度，设置为十六进制数字 FFFFFFFF（表 11 的例子 3 中的第二个项目）。那么，我们需要用项目分隔符项(FFFE,E00D)来标记 DICOM 对象项目的结尾。这个分隔符项是零长度的，因为他只是用来标记 DICOM 对象的结尾并且自己不含有任何数据。
3. 与项目相似，整个 SQ 序列能够显式长度或未定义长度：
 - a) 如果使用显式长度（例子 2，长度 00000A00），那么这个长度等于所有编码的 DICOM 对象序列（后面马上跟上长度标签）的总长度。在例子 2（见表 11）中，我们读取 00000A00 字节并且根据(FFFE,E000)标签将他们分割成 DICOM 对象。
 - b) 我们也可以使用隐式（未定义）长度，在 DICOM 中会标记为 FFFFFFFF。因为长度是未知的，所以我们不得不用零长度的序列分隔符项(FFFE,E0DD)来标记 SQ 序列的结尾。

Table 11 SQ编码

例子1：使用显式VR的SQ数据元素被定义为未定义长度的项目（VR=SQ）的序列，包含两个显式长度的DICOM对象（项目）。

标签	VR		值长度	值(DICOM 对象的序列)						序列分隔符项	
(gggg,eeee)	SQ	0000	0xFFFFFFFF (未定义长度)	第一个项目			第二个项目			序列分隔符标签 (FFFE, E0DD)	长度 00000000
				标签: (FFFE, E000)	长度: 0x1234	值: DICOM 对象	标签: (FFFE, E000)	长度: 0x1000	值: DICOM 对象		
4 字节	2 字节	2 字节	4 字节	4 字节	4 字节	0x1234 字节	4 字节	4 字节	0x1000 字节	4 字节	4 字节

例子2：使用隐式VR的SQ数据元素被定义为有三个显式长度的项目序列。

标签	值长度	值(DICOM 对象的序列)					
(gggg,eeee)	0x00000A00	第一个项目			第二个项目		
		标签: (FFFE, E000)	项目长度: 0x000004F8	值: DICOM 对象	标签: (FFFE, E000)	项目长度: 0x000004F8	值: DICOM 对象
4 字节	4 字节	4 字节	4 字节	0x04F8 字节	4 字节	4 字节	0x04F8 字节
		长度计算: 0x04 + 0x04 + 0x04F8 + 0x04 + 0x04 + 0x04F8 = 0x0A00					

例子3：使用隐式VR的SQ数据被定义为无长度的项目（VR=SQ）序列，包含两个项目，其中一个是显式长度而另一个项目是未定义长度

标签	值长度	值(DICOM 对象的序列)								序列分隔符项	
(gggg,eeee)	0xFFFFFFFF (未定义长度)	第一个项目			第二个项目					序列分隔符 标签 (FFFE, E0DD)	项目长度 00000000
		标签: (FFFE, E000)	项目长度: 0x0000 17B6	值: DICOM 对象	标签 (FFFE, E000)	项目长度 0xFFFF FFFF (未定义长度)	值: DICOM 对象	项目分隔符标签 (FFFE, E00D)	长度 00000000		
4 字节	4 字节	4 字节	4 字节	0x17B6 字节	4 字节	4 字节	(未定义长度)	4 字节	4 字节	4 字节	4 字节

如果你对比 SQ 和 XML 的分隔符的话，那么未定义长度和 XML 所采用的方法很相似，SQ 标签(gggg,eeee)的作用和 XML 的小于 (<) 标记作用相当；而(FFFE, E0DD)分隔符与 XML 的大于 (>) 标记作用相当。与这相似，在对象/项目层面(FFFE, E000)相当于 XML 的小于 (<)；而(FFFE, E00D)相当于大于 (>)。放着分隔符与 XML 的类似性不谈，未定义长度分隔符的美丽之处在于你不需要计算你的 DICOM 对象或序列的整个长度。因为，如果需要计算整个长度，那么你在长度值方面犯了点小错，也会造成整个序列都将无法读取。因此，未定义长度分隔符看起来更可靠也更易实现。

另一方面，如果你对阅读 SQ 内容没啥兴趣，那么显式长度（表 11，例子 2）允许你跳过整个 SQ 元素或者它的单个项目。比如你在例子 2 的对象中已经读取 00000A00 序列长度之后，你可能快进过 0x00000A00 = 2560 个字节直接去读取下个元素，而不管整个 SQ 元素，无论因为什么，至少你根本不需要去处理这些字节的数据了。总之，你提高了效率，这在 1983 年是个大事，可如今已经没那么重要了。现在，制作精良的 DICOM 软件去读取最复杂的 DICOM 数据也只需要几分之一秒，而执行效率更多的是受到网络速度和图像编码方式的影响而不是使用隐式长度项目。因此，如果你来实施 DICOM 软件，我建议坚持使用隐式（未定义的）长度分隔符，因为对于编写任何的 SQ 元素来书，他们都更简单易懂。

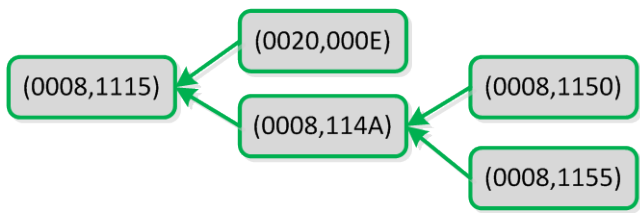
然而，请记住如果在编写你自己的 DICOM 序列时你可以在显式和未定义长度之间做出选择的话，那么为了能让你读取你的 DICOM 对象，你必须将两者全都实现。换句话说，你可以随使用，但是请对那个可能已经使用了另一种长度编码类型的 DICOM 厂商更礼貌些吧。毕竟你也有责任去读取他们的对象呀。

还要记住，在我们三个例子中的每个 DICOM 对象项目都会轮流包含 SQ 对象，因此他们的编码规则实际上是一致的。我们来回顾一下 5.3.10 中提到的嵌套对象，它会以 SQ 元素的形式进行存储，如图 13 所示。在这个例子中，每个 SQ 元素只包含一个 DICOM 对象。三种编码例子中的每一个都可以使用不同的显示长度并且从图中的底部开始编码。首先，我们在 2 层上编码一个 DICOM 对象（它没有 SQ 并且我们以一个普通 VR 序列的方法对它编码）。之后编码的对象放在了(0008,114A)的 SQ 元素中，并且用显式或未定义长度进行编码，此时我们也就进入了 1 层编码。最终，这个编码过程会再被重复，继续创建一个 1 层的 DICOM 对象并且放在(0008,1115)元素中。(0008,1115)元素会成为唯一的根 DICOM 对象元素，用来存储和网络传输。如果你在设计 SQ 编码，那么你能简单地按照递归的方式实施，形成各种树形数据结构。

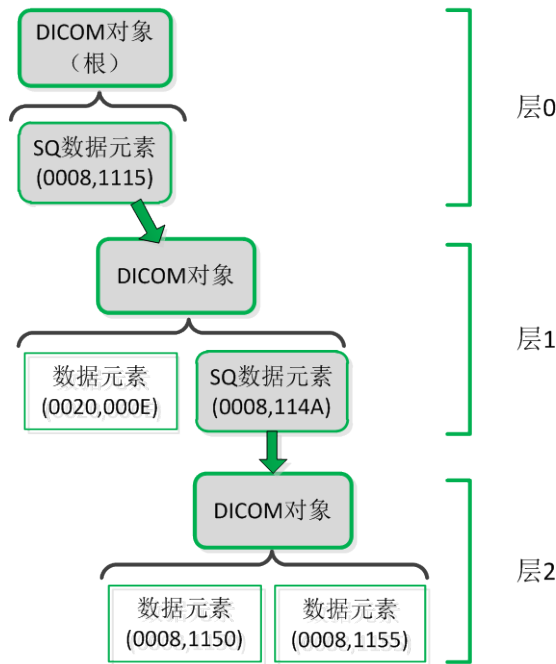
DICOM数据片段

属性名称	标签
参考的检查序列序列	{0008,1115}
>检查序列实例 UID	{0020,000E}
>参考的实例序列	{0008,114A}
>>参考的 SOP 类 UID	{0008,1150}
>>参考的 SOP 实例 UID	{0008,1155}

数据片段的SQ框图



数据片段，SQ嵌套编码的表现



5.5.6 必须的和可选的数据元素

我们需要注意，即使不去管 DICOM 对象的复杂性，也需要对放在里面的内容多加小心。用 CT 亨氏灰阶（Hounsfield grayscale，以发明 CT 的那个人命名）初始化超声影像的存储过程将让所有人费解。DICOM PS3.3（“信息对象定义”）在这方面有很细致的描述，规定了每个特定的 DICOM 对象类型所必须的标签和元素，比如影像设备图像。

这将导致另一种对所有数据元素的分类方法：必须的（required），受限的（conditional）或可选的（optional）（DICOM 标准 PS 3.5 段落 7.4）。为了确保准确，DICOM 定义了以下属

性类型，见表 12。比如，在 DICOM 工作流中的病人是通过他们 ID 识别的而不是用名字。因此，“病人 ID”属性(0010,0020)在任何真实的 DICOM 数据对象中都会是类型 1。“病人姓名”属性(0010,0010)一般是类型 2。“图像有损压缩”属性(0028,2110)通常是 1C——这只有当图像已经被进行无损压缩后才会指定。

当像数字影像设备这样的纯 DICOM 设备在他们处理的对象中自动包含必须的元素时请一定小心，当你使用标准化稍差的 DICOM 转换软件时可能会出现问题。一个胶片扫描仪将你扫描的胶片打包在 DICOM 格式中时可能会完全忘记必须的标签的存在，这并不罕见。DICOM 对象（包括 DICOM 文件）丢掉必须的元素是非法的，其他 DICOM 设备可能会拒绝这个对象或完全错误地处理它。咱们拿病人 ID 丢失做个例子：如果没有指定这个元素，那么对象将变为无效的。或者更糟，如果指定了元素但是里面没有值，它会被理解成为“任何”（空元素会被理解为通配符）。这意味着你完全可以在胶片扫描仪上扫描许多不同的病人，然后将他们合并成一个病人！别不相信，这可是医疗数据归档中最常见的一种错误。

5.5.7

存储图像数据

图像也遵循同样的数据编码规则而且可以很容易的将图像放到 DICOM 对象中。你可能会想，在 DICOM 数据字典中每个医学图像都有许多与属性有关的重要元素。其中包括：

1. 图像高度：相当于“行”属性(0028,0010)。
2. 图像宽度：相当于“列”属性(0028,0011)。
3. 图像像素数据：相当于“像素”属性(7FE0,0010)。

后半部分则是实际的图像，在许多情况下这部分占了 DICOM 对象大小的 95%（医学图像很大）。

此外，你能把图像帧的序列装进一个 DICOM 对象里，DICOM 就是用这种办法保存像超声电影回放这种数字视频的。事实上，我们可以用更开阔的思路来看看这种处理方法：以数字化的信号序列表达数字图像。这会让他们看起来和其他信号数据一样，比如音频，你可能想记录数据并把它们放到 DICOM 对象里。只要数据字典用的好，这些都是有可能的。你可以方便地用多帧图像和音频将你最喜欢的电影编码成一个 DICOM 对象。

我要音乐电视！

我们前面讨论过，DICOM 将视频存储为单个图像帧的序列。比如，一个超声电影回放，有 10 秒长，用 25 帧/秒的频率记录，那么在 DICOM 中，这段电影将以 $10 \times 25 = 250$ 相邻图像帧进行数字化并存储在一个 DICOM 对象中。

当在 DICOM 软件中显示这些视频时，它们仍然会以 250 个静态图像序列的形式显示。换句话说，你总是不得不去按一个“播放”按钮来让他们像视频一样播放。我已经见过许多人被这个概念所困扰：如果这个电影回放是视频，为啥不让他自动播放呢？简单解释是，回放是你的浏览软件的功能而不是 DICOM 的功能。DICOM 存储了带有(0018,0040)“每秒帧数”属性的 250 个静态视频帧，就是为了告诉你这些帧应该按照视频来播放。

5.5.8

唯一标识

无论你是否会用到基于对象的应用程序，你都必须熟悉“实例”的概念。简单地说，实例就是某时刻对具有数据值的对象拍摄的一幅快照。对象数据的改变将会产生另一个对象的实例。对于医学图像来说 John Smith 的头部 X 线检查，存储在 DICOM 对象中。如果你复制一个 DICOM 对象，你实际上是创建了另一个图像实例。当一个原始的图像（实例）保持

着原来的面貌时，他的一个复制品可能已经被裁切、变色、标记等；总之，它可以被改的面目全非，即使它仍然被称作“John Smith 的头部 X 线检查”。放射医生到底在使用哪个实例呢？如果你参与实际医疗工作，那么你就会发现你最感兴趣的事就是区分各个实例。在 DICOM 中，归档时实例是有自己的 UID 的——唯一标识符。

图 14 描绘一个原始图像生成多个实例的过程，其中一些实例和原始图像其实是一致的，只不过是放在了不同的地方或者用在不同的地方。这就可以解释，为什么 UID 常用来标识这些实例，而且 UID 需要全世界唯一。图像实例能够被发送到离原件非常远的地方（比如，在其他国家进行远程放射诊断），在那里它会和一些其他图像一起存储在另一图像归档中。DICOM UID，就像人们的 DNA，用来唯一标记一个特定的实例，无论这个实例变成什么样。也因此，UID 在 DICOM 中不仅是用来标记独立的图像，而且是用来标识图像序列、检查、设备、数据交换协议的语法以及许多其他你不希望弄混的东西。

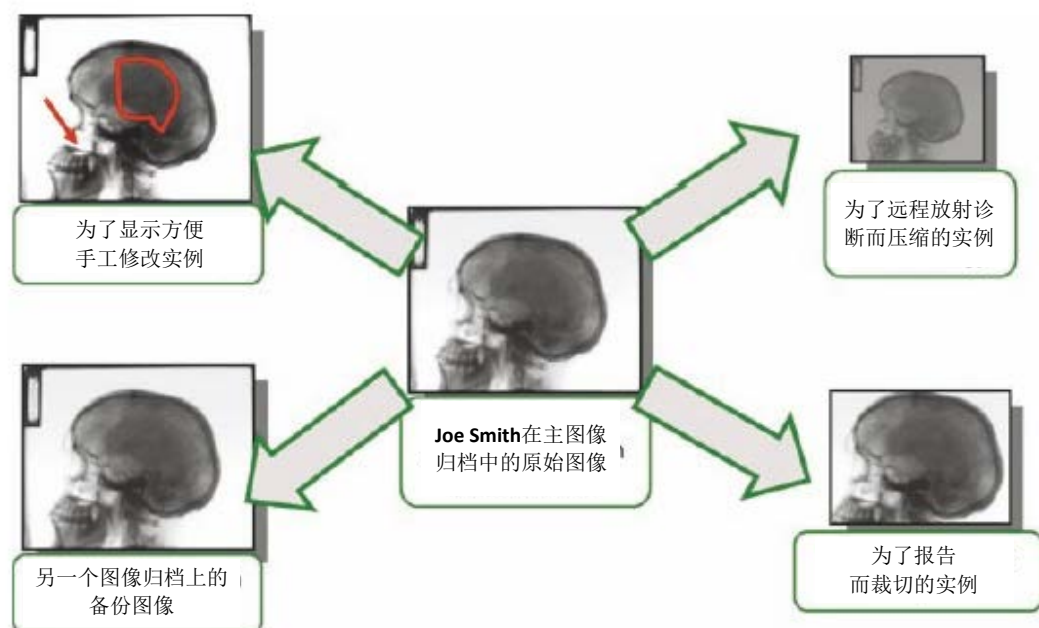


Fig. 14 相同图像的实例，每个需要不同的 UID

DICOM UID 是像“1.2.840.10008.1.2”那样由数字组成并由逗号分隔的字符串；所以他们符合 UI VR 类型（见 5.3 中表 2 的 UI 类型）。UID 字符串具有全球唯一性，能够保证在多个国家、地点、厂商和设备间区分病人。在我们全世界，无论图像来自何处，这是唯一的方法来确保是你的图像以及确保不会在传输过程中有被其他图像混淆的风险。因此 DICOM 使用下面的 UID 编码惯例：

UID = <org root>.<suffix>

其中“<org root>（组织 根）”部分唯一标识一个组织（即厂商、研究机构、NEMA 等等）。理想的做法是，每个组织应该申请并且使用他自己的根 ID（也不是总能这样）来保证这个根从来没人用过（见 DICOM PS3.5 的附录 C，了解注册规则）。“1.2.840.10008”字符串是 DICOM 事物 UID 所使用的全局保留“<org root>”，且不能在别处使用。

提示：认识 DICOM 文件

在写字板中打开一个未知的文件，在里面搜索 1.2.840.10008。如果找到这个段字符串，那说明你可能正在处理一个 DICOM 文件。在文件顶端的四个 DICM 字母（从文件开头数起第 129 – 132 个字符）是另一个分辨 DICOM 文件的方法。然而，一些较老的或不正确的 DICOM 实现方法中可能没有写这些字符，所以你会更容易找到像 1.2.840.10008 这样的 UID 前缀。

“<suffix>（后缀）”也是由许多数字组成的一部分 UID，并且“应该在<org root>范围内是唯一的”（见 DICOM PS3.5 部分 9）。相对来说根字符串比较短的，后缀字符串则是用来保证实例的唯一性的。比如，如果你生成了一个这样的后缀：

<病人 ID>.<检查 ID>.<当前日期>.<毫秒级的当前时间>

你可以完全相信在你的机构中不可能再有重复的 UID 了。这就是为什么所有长度为 64 个字符的 UID 类型迟早都会有用的；用的越多，UID 重复的可能性就越小。

基于同样的原因，UID 总是用作 DICOM 文件名。这可不是 DICOM 要求的（见 10.1.4）：真的 DICOM 文件名应该由 8 个字符组成，字符包括：大写字母、数字和下划线字符（比如：DIR1\SKW12AB5）。然而，对 DICOM 应用来说，如果 8 字符格式是强制的向外输出文件的命名形式的话（在外部的或移动的媒体上写入 DICOM 文件，比如 CD/DVD 或闪存，见第 10 章），那么当 DICOM 应用在自己的硬盘上存储他们的图像时，用 UID 命名内部的文件就很普遍了。用 UID 命名 DICOM 文件是合理的：每个 DICOM 图像对象包括“图像 SOP 实例 UID(Image SOP Instance UID)”，属性标签为(0008,0019)。这对于 DICOM 图像来说是个好名字。当你看到像这样的文件名时：

1.2.804.114118.2.20040909.125423.3692976692.1.1.1

你可能正在处理 DICOM 文件。其实 DICOM 文件只不过是在你硬盘上存储的 DICOM 对象而已——DICOM 对象的信息转存。

UID 解析？

即使 UID 名总是按照特定逻辑创建，也不要试图解析他们或者使用他们名称传递一些附加信息。比如，即使你知道一个 UID 名可能包含病人 ID 或检查日期，仍然不要去摘录这些数据。DICOM 明确地警告过不能使用这种方法。

UID 名不意味着任何数据交换，他们唯一的目的是区别多对象实例。如果只是为了区别，那么 DICOM 应用程序就可以自由地改变或更新 UID 名，只要它使用某种一定的附加命名逻辑。除 DICOM 所指示的做法之外，你的 DICOM 厂商总是会想办法维护正确的 UID 发布策略的。在 DICOM 遵从性软件的帮助下，这些厂商必须跟踪图像改变并且为同一图像的每个不同的临床实例创建一个新的 UID。

5.6

DICOM 信息等级 (hierarchy)

在我们回顾 DICOM 数据之前，有必要快速看看 DICOM 信息结构。当然，DICOM 数据字典中的属性在将现实世界数据映射到 DICOM 标准方面的确起了非常重要的作用。但是，这些属性实在太多了，难道我们真的不需要按照一定的顺序来排布他们吗？

这是在 DICOM 中归档时使用的顺序，病人-检查-序列-图像等级（图 15）。

1. 一个病人可能有多个检查（Study）。
2. 每个检查可能包括一个或多个图像序列（image series）。
3. 每个序列有一个或多个图像（image）。

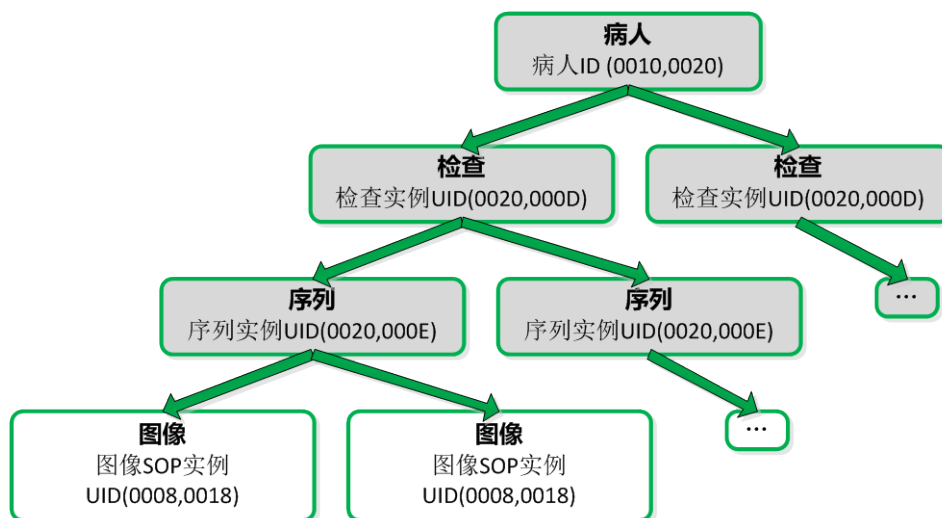


Fig. 15 DICOM的四层信息等级。通过关键元素来分别每个层次

这个等级天然地反映了在真实世界中当一个病人进行了医学影像检查后的情况。John Smith 来到医院，预约了一些检查（比如，MR、CT 和超声检查）。许多后续的检查迟早会进行。每个检查可能有多个图像序列（冠脉、轴扫、是否增强、有变化的成像协议等等）。并且每个序列，很自然地，有一个或多个图像。现在，如果我们需要找到或排序一个病人的图像，那么我们可以基于他的病人、检查、序列和图像属性来实现。比如，我们可能希望找到一个病人几年以来做过的所有 MR 检查。

为了实现这种等级，DICOM 为每个等级设置了一个关键层级 ID。在病人层，设置了病人 ID（Patient ID）（所有病人都有自己的 ID，用来唯一地标识他们）。“病人 ID”元素存储在 (0010,0020) 的标签下，这个标签支持任何成像类型。在病人-检查-序列-图像等级中，除病人层外其他层次，也同样使用这种原则：在检查层，每个检查有其唯一的“检查实例 UID（Study Instance UID）” (0020,000D)；在序列层，每个序列有其唯一的“序列实例 UID” (0020,000E)；而在图像层，每个图像有其自己的“SOP Instance UID” (0008,0018)。检查、序列和图像 UID 使用 UI VR 类型格式，用 64 字符长得唯一（实例）标识符定义，其值只是用数字和逗号。病人 ID，由于经常用在需要人类读取的接口上，因此可能会遵从低结构化的 LO（长字符串）VR 类型。它最多仍然只能使用 64 个字符，但是几乎可以使用任何字符。

所有 DICOM 命令和大多数的 DICOM 数据属性总是绑定这种四层信息模型。因此，四个 DICOM 等级属性起着重要作用：就像它们的名字所说，它们唯一标识它们的数据。比如，如果两个检查有同一个检查实例 UID 值，那么会认为它们是同一检查，并且在序列和图像层中包含同一个 UID 集合。此外，如果两个病人有相同的病人 ID，那么会认为它们是相同的病人。如果恰当的运用，那么这种标识将从本质上提高数据识别能力。否则，则很容易导致错误。

5.6.1

病人 ID 相关问题

病人 ID 的使用意味着消除了使用病人姓名来分辨病人所带来的比较明显的问题。简单地说，病人姓名作为病人标识不可靠，原因如下：

- 1. 可能会被拼错（字母、符号、甚至空格）。
- 2. 可能输入的不正确（比如，名字和姓录反了）。
- 3. 可能会经常改变（由于婚姻、法律问题等）。
- 4. 很难用本地文字表示其他语言（在不支持某种外语字符的 DICOM 装置上输入外语姓名；假设在英文系统的 CT 机上录入日文姓名）。
- 5. 侵犯病人隐私。

考虑到以上原因，所以使用像 12345XYZ 这样的 ID 来标识某个病人显然是更好的选择。在我们这个不完美的世界中，没有什么东西是生来就绝对没问题的。

首先，世界上没有中心病人 ID 库来产生或维护一致的病人 ID，所以也没有普遍的 ID 规则。当一个病人出现在医院时，医院的政策通常是一定要先给病人分配一个病人 ID。一些医院可能使用病人的社保号（美国适用），生日或病人特定的一些其他信息；既然这样，至少下次病人就诊时可以使用同一个病人 ID。另一些医院可能只是简单地使用字母数字编码或更糟，试图使用一个循环使用的数字来使其显得牢靠。“安全”是指 ID 需要地点唯一（不能在其他地点使用同一个病人 ID）和事件唯一（可能是在同一地点的晚些时候）。大多数机构可能只是简单地采用病人姓名作为病人 ID，那就相当于抛弃了所有病人 ID 的优点。

此外，一些医院会依据病人检查设备不同，或者依据病人图像会由哪个放射诊断读取部门的不同，而改变病人 ID。他们很可能是以“123.ForDrSmith”或“123.Followup”结尾，但其实都是同一个病人 123。如果使用了这种病人 ID 命名法，那就是一场噩梦的开始；不管为了什么目的，但它一点都起不到标识的作用，至少它相当不可靠。此外，ID 是一个唯一的数据证书，不是用来编码一些不相干信息的。

我们都是兄弟姐妹

如之前所说，DICOM 认为所有使用相同病人 ID 的病人就应该是同一个人。我最近见过一个著名医院出现的滥用病人 ID 的案例。所有病人来到医院都没有 ID 号，只是使用一个“W/I”病人 ID，表示“没有 ID（Without ID）”。每天有大量这样的检查，对于 DICOM 来说，实际上是把所有这样的病人合并成了一个病人。因此，PACS 软件根本没办法来区分这些“没有 ID”病人到底是谁。

如果不能提供 ID，那么用病人生日（比如 JS19670102）也比填成“W/I”要好，至少重复率会低一些。

你可以在 DICOM 数据字典中找到另一个属性，叫做“病人其它 ID（Other Patient IDs）”（0010,1001）。用来减少一个病人多个 ID 所带来的麻烦，但是 DICOM 软件很少会使用这个属性。基于一个很明显的原因：对于一个病人来说，确保单一 ID 一致性比处理不同 ID 的列表更有效（可能会和其他病人重叠在一起）。

有什么处理病人 ID 一致性的其他办法呢？肯定有。好的 DICOM 软件从来不会只相信病人 ID 属性，并且总是执行一些附加逻辑来校验病人身份。附加的逻辑的实现是基于 DICOM 数据元素所提供的其他数据信息来实现的。病人姓名、生日、性别、最后的检查日期、体重、体型、和一些其他的标签能够用来得出一个关于病人相似性和身份的智能结论。比如，如果有两个 ID 不一样的病人，但是他们名下的信息中这些参数是一致的，那么可以基本肯定这两个病人是同一个人，应该重置其中一个 ID（最好是先使用的那个）。相同的情况也被用来

对姓名拼写错误或者有其他数据错误的病人进行合并。不同的病人合并功能通常都是由 PACS 软件执行的。

如果病人被合并错了（比如，如果相同的病人 ID 被错误地分配给了两个不同的病人），相同的方法也会用来拆分病人。我记得一个影像中心使用的所有病人 ID 都是 3 个数字；但是你只能用 000 到 999 来区分一千个病人。当你的病人量超出一千个人以后，你又要从头开始，循环使用这些编号，将它发给另一个病人，因此造成错误地合并了不同的病人。使用更短小或更简单的 ID 是不可能让你的工作变得更简单的。

病人拆分和合并是不应该自动完成的。当一个 DICOM 应用程序发现需要进行拆分或者合并时，程序应该标记可能的记录并通知人类管理员来处理。无论程序逻辑有多复杂，都仍然有可能导致错误的合并和拆分，并且由人工处理来避免合并和拆分的错误是很重要的。

随着分布式 PACS 作用的提升，地区或国家性的系统渐渐替换掉了本地的系统，纠正病人标识和确保其一致性的要求也随之提高。使用智能卡、射频标识、以及生物病人标识（可在不违反病人隐私的前提下实现）来编码和分辨不同病人的技术已经在医疗信息技术方面获得了应用。

真实案例：双胞胎

曾经，我遇到了一个双胞胎姐妹的案例。姐妹俩在同一时间去看同一个医生，并且右踝也有同样的问题。姐妹俩有相同的姓（你可能已经猜到了），并且更有趣的是，他们的名字有 20 个字母长。当第一次在技师工作站上的“病人姓名”框中输入她们的姓名时，根本没有地方输入她们的名字了。值得庆幸的是，姐妹俩被分配了不同的 ID；但即使这样，由于 ID 只能标识信息而无法描述病人，因此连医生都不能区分他到底在看谁的图像。

千万别以为这只是个特殊情况。

5.6.2

检查、序列和图像 UID 的相关问题

对于检查、序列和图像 UID 来说，有一点比病人 ID 要强，就是不像病人 ID 那样，几乎不需要手工输入或分派 ID 值。这些 ID 是由影像设备自动生成的，并且自动将数值插入到图像的 DICOM 标签中。

另一个好处是，检查、序列和图像数据是在扫描同一个病人的时候产生的。若是病人 ID，我们就需要考虑在不同检查或者不同地点保证它一致。检查、序列和图像 UID 则是在同一个连贯的成像过程中产生的，因此很容易与其他相区分。这同样也简化了生成检查、序列和图像 UID 的方法。比如，这些 UID 内容可以是当前的日期/时间字符串，其中时间的准确度可以到秒级如 5.5.8 中所述。这个 UID 生成方式可以使整个过程更加的可靠，但是不可避免有一些常见错误。

最烦人的问题常常出现在由技术原因造成的 ID 错误。我已经见过很多次以下情况了，检查设备会将不同的序列实例 UID 分配给了同一个序列中的图像。在任何 DICOM 软件中，图像是通过唯一的序列 UID 组成图像序列的，因此同一个序列具有不一致的序列 UID 值可能会将一个序列中的所有图像分割成多组不相关的图像。比如，一个 CT 序列有 2000 张图像本应按照成像时间和三维成像位置来排序，但是你却看到了 2000 张无法看出任何关联的，被打散的图像。如果图像显示软件能够根据成像时间来进行图像排序，那么成像时间至少能够保证图像能够临时按照时间排序，但是这不过是给该问题贴了块创口贴而已。与病人 ID 不同，DICOM 软件很少提供序列的合并功能。因此处理序列 UID 问题的唯一出路是在哪里出了问题就在哪里解决：在影像设备装置上改正该问题。

还有另一个由序列、检查或图像 (SOP) UID 的唯一性所产生的典型但麻烦较小的问题。UID 中的字母 U 表示“唯一 (Unique)” (标识符)，代表标识符值应该是全球唯一的。如果对图像进行了任何改变 (如剪切、旋转或用有损方式进行了压缩)，即使所有临床重要信息还保持不变，但是图像也和之前不一样了。所以，这种修改后图像的图像 (SOP) UID 是需要改变的 (见 5.5.8)。如果看到两个图像序列看起来一模一样，但是仍被分别存储在一个图像归档，请千万不要奇怪，因为只要 UID 不一样他们就是两个不同的图像序列。

在影像设备上自动产生检查、序列和图像 ID 是一个比较好的办法，但是要是这一切没有正常地发生那又能怎么办呢？最典型的例子是二次获取 (Second Capture SC) 的 DICOM 图像，比如扫描 (数字化) 的胶片。当你进行胶片扫描或导入非 DICOM 图像，并将其转化为 DICOM 图像时，你的胶片扫描仪肯定不清楚众多胶片或者图像之间的关系，只有你知道是怎么回事。这些图像属于同一个病人、检查或序列吗？你的 DICOM 扫描仪应该可以提供完备的检查、序列和图像 UID 的手工录入功能，以便在被数字化的 DICOM 数据中记录原来图像之间的关系。不幸的是，不是所有的 SC 设备都会重视实现这些功能。如果你正计划买一个胶片扫描仪或其他 SC 设备，那么请确认它对输入的 DICOM 数据有足够的支持，并且可以将图像分配到相同的病人、检查或序列中。

5.6.3

分级数据和关系数据

为了把 DICOM 病人-检查-序列-图像信息模型说明白，让我们简短地提一下，这种等级表现了真实世界所映射出来的更加复杂的 DICOM 模型的核心部分。这个模型可以在 DICOM 标准的 (PS3.3 第 7 部分) 找到，此外在不改变本质的前提下，还可以在基础的病人-检查-序列-图像的等级中增加更多的项目。事实上，所有 DICOM 设备都在病人-检查-序列-图像级别方面起作用，实现分级数据的查询、提取和处理。

为了识别一个序列，你首先会需要找到这个病人，并且找到序列所属的检查。在所有的 DICOM 界面上都是这个逻辑。这些 DICOM 界面总是从顶端开始浏览其中的数据，即从最高的病人层 (有时也会是检查层) 开始，逐步向下去浏览序列和图像。此外，分级的 DICOM 查询和数据提取也会因为不知道更高级别的 ID 而失败。比如，如果预先不知道序列所属的检查和病人 ID，那么对一个序列进行分级查询肯定会失败。

唯一能够代替分级数据处理过程的是关系数据处理过程。和分级数据处理不同，关系数据处理并不会把数据分为四层，而是允许你用任何可能的方法来查询数据 (而是基于数据之间的关系来实现的)。比如，只要你能提供其他的数据来识别序列 (比如影像设备、日期/时间、UID 等等)，即使你不知道他的检查 ID 和病人 ID 也能够查询到某一个序列。

DICOM 支持关系 (非分级的) 数据处理过程作为另一种处理选择。在 DICOM 装置上，若分级数据处理过程是强制的，那么关系数据处理过程或许可以作为附加选择，但多数情况是不能的。是否提供关系数据处理过程应该在 DICOM 一致性声明中说明。

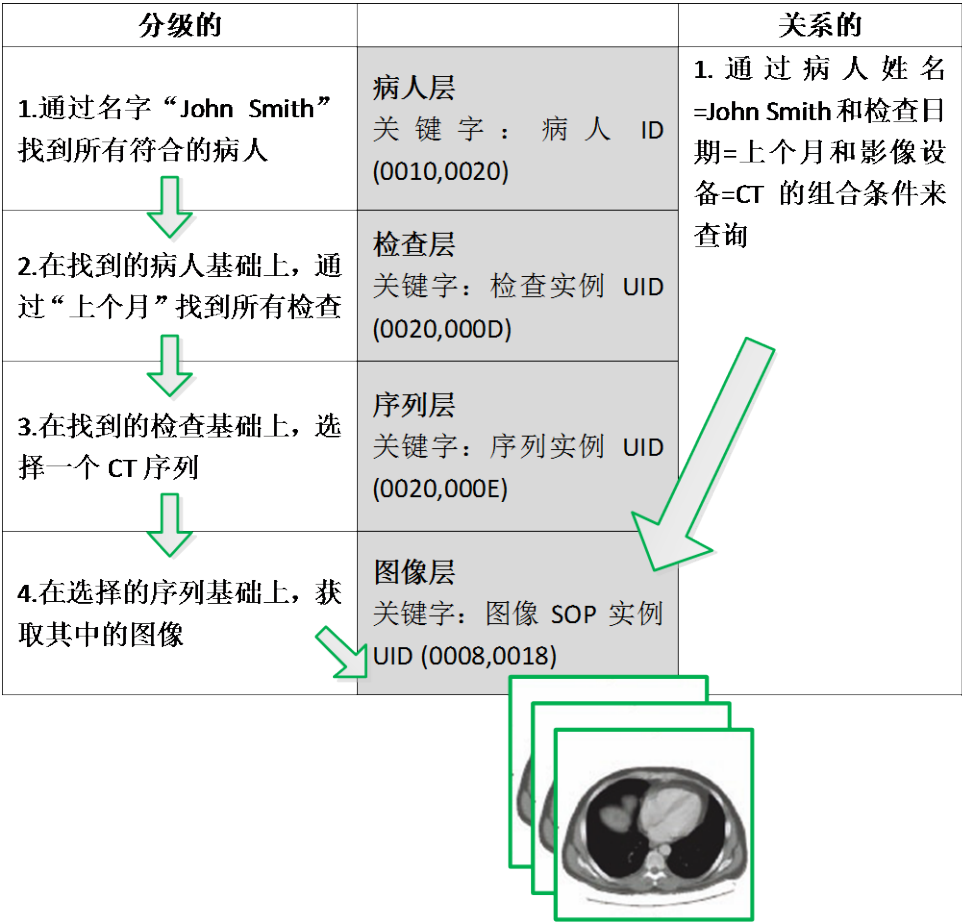


Fig. 16 分级的（左侧）和关联的（右侧）查询实例——查询“John Smith上个月进行的CT检查图像”。分级的查询比较简单而且效率更高，但是步骤较多。

5.7
模块、IOD 和信息实体

用数据元素来生成 DICOM 对象是唯一的正确方法，但是在 DICOM 数据字典中有 2000 多个数据元素，我们应该让它们更加结构化。你不能真的在一些 MR 特有的元素（比如 (0018,0087)磁场强度）后；增加一些 CT 标签（比如(0018,0060)中的 KVP）；插入一些超声图像然后把这个作品称作 DICOM 对象。这种妄想将直接被大部分 DICOM 设备拒之门外，即使是那些具有神奇般灵活性的 DICOM 数据处理机制，也根本没法处理这种混合体。

数据元素是最小的组件，不一定能把它们拼凑在一起。为了让它们更好地排布及使用，我们需要将数据元素组合成更大的组件，并用他们组成更加有意义和更加确定的 DICOM 对象。

DICOM 将这些大块组件称为信息模块（Information Modules），信息实体（Information Entity, IE）和 IOD。像在 DICOM 中的许多事物一样，它们都是有等级相关性的。模块形成 IE，用来创建 IOD。由于 DICOM 数据的定义对于每个影像设备是不同的：模块、IE 和 IOD 的选用，都是由影像设备决定的。所以让我们近距离看看这些组件是如何工作的吧。

Table 13 序列和实例涉及的宏属性

属性名	标签	属性描述
参考的图像序列的序列	(0008, 1115)	项目的序列，每个序列包括一个检查序列的属性。展现为一个或多个项目
>图像序列实例 UID	(0020, 000E)	包含参考实例的序列的唯一标识符
>参考的实例序列	(0008, 114A)	项目的序列，每一个提供一个实例的参考，这个实例是在包裹项目中由序列实例 UID 定义的序列。会有一个或多个项目
>>参考 SOP 类 UID	(0008, 1150)	唯一地标识参考的 SOP 类
>>参考的 SOP 实例 UID	(0008, 1155)	唯一地标识参考的 SOP 实例

5.7.1
宏属性（Macro Attribute）：变得更简单

通常，我们希望将相似的属性组合在一起以便缩短其他模块的引用量。DICOM 称这些共有的规格参数表的组合为“宏属性”（表 13）。用序列化来标记（见 5.3.10）。大于号（>），总是代表下一个元素是以 SQ VR 形式包含在前一个元素之中的。但是即使没有序列化，一些宏也会非常的大并且能够在 PS3.3 中占用好几页的篇幅（比如，图像像素宏）。即使考虑使用信息结构化，宏也不等同于任何对象或数据块；宏只是用来帮助我们重用元素表的。

5.7.2
信息模块：基本数据块

模块，提供数据元素组织归纳的最早的和最必要的层级。比如，病人标识模块（Patient Identification Module）（表 14）将所有病人标识信息归为一组，包括病人姓名、ID、本姓、娘家姓等数据标签。

Table 14 病人标识模型：标识一个病人

属性名	标签	值/描述
病人姓名	(0010,0010)	病人的全名

病人 ID	(0010,0020)	一个病人在当前医院的标识编号或代码
其他病人 ID	(0010,1000)	用来分辨病人的其他标识编号或代码
其他病人姓名	(0010,1001)	用来分辨病人的其他姓名
病人的本姓	(0010,1005)	病人的本姓
病人母亲的本姓	(0010,1060)	病人母亲的本姓
医疗记录定位符	(0010,1090)	用来查找病人医疗记录的一个标识符（比如胶片夹）

模块的内容是可以自解释的：他们获取所有可能与病人标识有关的属性（数据元素）。对于一个特定的病人，显然不是所有的属性都是能够收集在一起的，在一些情况下可能什么都不能确定（比如，遇到一个失去知觉的病人）。然而，模块并不意味着包含它所有属性的全部值。未知的字段是可以留空的，或者如果不是必须的，甚至可以忽略掉。模块的主要目的是用连续的和成体系的方法来收集相关数据属性（元素）。比如，如果我们病人成为一个临床试验的课题，我们可以增加一个临床试验的模块来记录所有的试验属性（表 15）

Table 15 临床试验模块：基础的临床试验属性。

属性名	标签	属性描述
临床试验主办方	(0012,0010)	临床试验主办方名称
临床试验方法 ID	(0012,0020)	标注试验方法的标识符
临床试验方法名称	(0012,0021)	临床试验方法的名称
临床试验地点 ID	(0012,0030)	负责提交临床试验数据的地点标识符
临床试验地点名称	(0012,0031)	负责提交临床试验数据的地点名称
临床试验题目 ID	(0012,0040)	临床试验题目指定的标识符。如果临床试验题目可读取 ID(0012,0042)不可用则会出现。也有可能因为其他原因而出现。
临床试验题目可读取 ID	(0012,0042)	标识盲法验证的题目。如果临床试验题目 ID(0012,0040)不可用则会出现。也可能因为其他原因而出现

为了给你提供一个稍有不同模块实例，让我们在表 16 中看一个电影回放模块吧，这个模块定义了多帧图像的回放参数。如果一个 DICOM 对象包含一个视频多帧的序列（图像的序列就是数字化视频的连续帧），一个电影模块存储这需要播放的序列信息。最必要的参数是“帧周期” (0018,1063)，表示帧之间毫秒级的间隔。比如，一个平均 25 帧/秒的视频频率相当于(0018,1063)中的 $1000/25=40$ 毫秒。然而，如你所见，根据不同的数据内容而定，许多其他属性也是可以支持的。需要注意的是，电影模块使用“代码序列”宏。

像 DICOM 中几乎所有东西一样，信息模块可以是强制的（在 DICOM 中通过大写 M 标明）、有条件的（如果其他特定的模块存在那么就需要，并且用大写 C 来标明）、或者用户定义的（为了私有数据元素设置的；用大写 U 标明）。如何合理地选择模块是由在 DICOM 对象中存储的数据类型决定的。在多数情况下，这个类型要与影像设备（CT、MR、计算机 X 线摄影——CR 等）相符；但是也有非影像设备对象的类型（比如 ECG（心电图），报告或者录音）。例如，病人标识模块对于任何 DICOM 影像设备来说都是必须的；即对于一个数字图像来说，我们必须了解它属于谁。电影模块，对于多帧超声影像（比如超声回放）来说是必须的，但是对于 CT 或 MR（这些设备不会产生数字视频）来说就不是必须的了。

还要注意（尤其在你对数据库设计比较熟时），DICOM 模块不是规范化的；因此，不同模块可能会共用一个属性（有重叠的数据）。比如，像“病人姓名”(0010,0010)就在许多 DICOM

模块中都能见到，并且病人模块（见 DICOM PS3.3）中包含病人标识模块中的大部分数据（见表 14）。事实上，模块中的偶数属性是冗余的。比如，在电影模块中（如表 16），“帧周期” (0018,1063)和“电影速率” (0018,0040)显然是相关的（电影速率=1000/(帧周期)）。这不是结构化复杂数据的最好方法，因为这会使许多模块更多地依赖其他模块。然而，我们可以理解在 DICOM 标准演化的过程中需要兼顾实用性和历史兼容性。

DICOM 标准的第 3 部分包含所有标准模块（大概有 100 个）和内容很多的信息对象模块规格说明书，其中有每个影像设备相关模块的细节。

为 DICOM 开发者
<p>你正在设计 DICOM 软件吗？那么我强烈建议你使用 DICOM 模块作为你的基本数据组件。应该考虑通过创建基本模块类来开始设计，基本模块类使用通用数据元素编码（用显式和隐式 VR 来读写 DICOM 对象）并且可以对每个你需要的具体 DICOM 模块创建子类。千万别把 100 个模块都给实现了！如果进行了清晰的类设计，那么你能够根据你的需要增加一个新的模块而只需要极少的代码量。</p> <p>比如，可以从最基础的设备类型（比如 CT）开始，对所有涉及 CT 成像的模块（从信息对象模型表 PS3.3 中得到）进行编码，如果需要则可以扩展出一个新的影像设备（模块）。对于重叠的模块，则采用增加相关操作员的方法将他们互相转换。对于冗余的数据属性（比如表 16 中的帧周期和电影速率），他们只支持极少的属性子集并且可以用正当的恢复功能来恢复原来的数据。简而言之，完全使用面向对象的编程方法将会使编程更加简单和高效。</p> <p>在你所见到的所有 DICOM 软件实例中，DICOM 模块都已经被忽略了。对象会直接通过数据元素（属性）来创建，这虽然更高效但是有可能带来错误。希望 DICOM 工作组能够持续推敲标准的模块设计，让它们变得更正规、更完备。</p>

Table 16 电影模块：DICOM 视频的回放

属性名	标签	值/描述
选用的回放顺序	(0018,1244)	描述多帧图像所选用的回放序列。计算值： 0 = 循环(1,2,...n,1,2,...n,1,2,...n,...) 1 = 摆动(1,2,...n,n-1,...2,1,2,...n,...)
帧周期	(0018,1063)	每个帧的标称时间（单位：毫秒）。如果帧增长指针(0028,0009)指出了帧周期，那么这个属性就是必须的。
帧周期矢量	(0018,1065)	对于多帧图像，保存着帧之间实时增量的数组。如果帧增长指针(0028,0009)指出了帧周期矢量，那么这个属性就是必须的。
起始帧	(0008,2142)	在播放的多帧图像中，第一帧的帧编号。
截止帧	(0008,2143)	在播放的多帧图像中，最后一帧的帧编号。
推荐的显示帧速率	(0008,2144)	推荐的速率，多帧图像的帧应该以每秒的帧数来显示
电影速率	(0018,0040)	每秒的帧数
帧延时	(0018,1066)	在多帧图像中，从内容时间(0008,0033)到第一帧的总时间（单位：毫秒）。
图像触发延时	(0018,1067)	从触发（比如 X 线脉冲）到第一帧的延时（单位：毫秒）。
有效时长	(0018,0072)	实际获取全部帧图像数据的总时长（单位：毫秒）
实际帧时长	(0018,1242)	每帧数据获取消耗的时间（单位：毫秒）
多声道描述代码序列	(003A,0300)	各种关于多声道的描述。如果传输语法（Transfer Syntax）用过多帧图像编码且包含多声道，那么这个属性就是必须的。比如 MPEG2 传输语法。零或多个项目可能会在序列中出现。
>声道标识码	(003A,0301)	一个用来识别的声道参考值，用传输语法编码的比特流（1 为主声道，2 为第二声道以及 3-9 的互补声道）。
>声道模式	(003A,0302)	一个编码的描述符限定了声道的模式。枚举值： MONO = 单信号 STEREO = 两个信号同时获取的信号（左和右）
>声道来源序列	(003A,0208)	一个声道来源的描述符。在序列中允许只有一个项目。
>>包括“代码序列宏”		定义的上下文 ID 声道来源 3000

5.7.3

信息实体（Information Entities）

DICOM IE 是通过 DICOM 信息模块组建的。通过信息模块建立 IE 是一个非常简单易行的工作。对于每个 IE，DICOM 仅仅列出了一些 IE 可能包括的模块。比如，常见的病人 IE 应该包括病人模块、范例身份证明模块和临床试验课题模块。常见的检查 IE 包括通用检查模块、病人检查模块和临床试验检查模块。当到了序列和图像层，则各类影像设备的区别会变得更加重要，IE 也会变得更复杂，因为有更多的模块参与其中。

如果我们已经有信息模块了，我们还要 IE 有什么用？IE 在 DICOM 信息模型中体现了复杂性的下一个级别。如果模块代表组合相关数据元素（属性），那么 IE 就是用来表达与医疗成像有关的现实实体的。理解 IE 的最好方法去看看 DICOM（PS3.3）中的流程，此处参看图 17。

图 17 中的框框就相当于 DICOM IE。一个病人在设备上进行了某种检查；之后会产生图像的序列，但是为了获取报告数据，可能也包括其他数据（比如，声音的波形、结构化报告（SR），记录图像位置的参考系以及空间和时间的方向等等）。这里有 20 个 DICOM IE，他们的列表由 DICOM WG（工作组）不断地更新，并且你可以在最新版的 DICOM 标准（PS3.3）中找到他们。

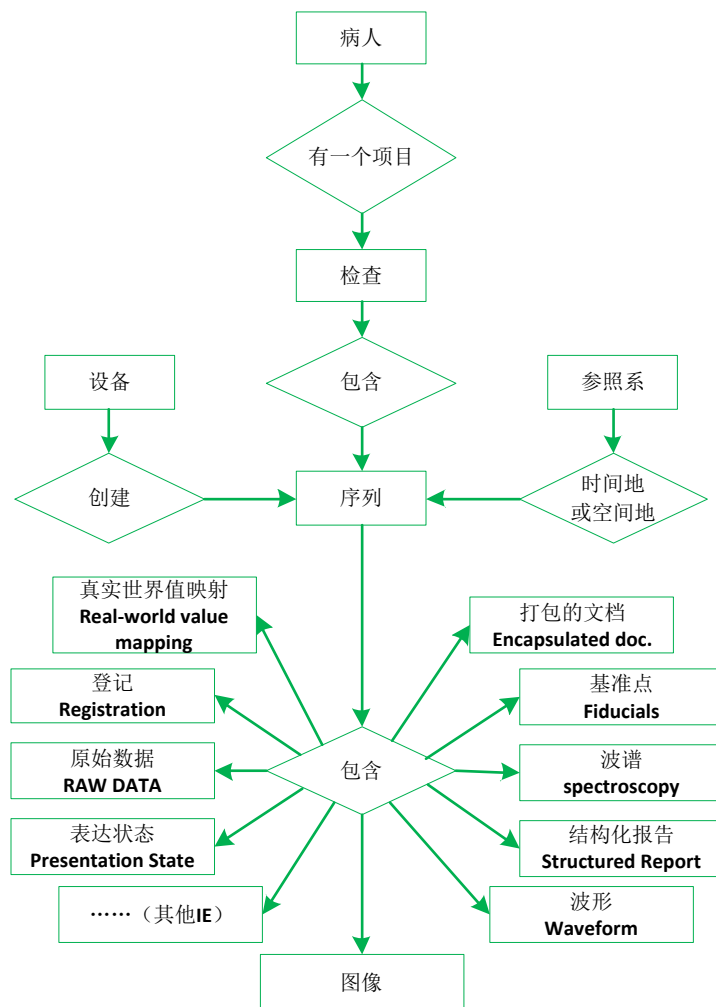


Fig. 17 具有不同信息实体的 DICOM 信息模型

5.7.4

DICOM 信息对象

当信息的结合是有意义的，IE 就组建了 IOD。IOD 是 DICOM 信息等级的巅峰；他们定义

了 DICOM 使用的对象。换句话说，DICOM 的整个数据处理过程是基于 IOD 实现的，并且所有“高层”DICOM 数据必须符合规定的 IOD 类型。

总之，构建 IOD 这种形式就是为了表述数字成像事物中最常见的数据类型，比如来自各种影像设备的图像。表 17 展示了我们如何才能通过适当的 IE 和信息模块的集合来建立一个 CT 图像的 IOD。仅仅需要在这个 IOD 中用“MR 图像”信息模块替换掉“CT 图像”信息模块，我们就可以得到 MR IOD 了。但是如果要获得一个 NM（Nuclear Medicine 核医学）IOD，我们还需要针对核医学的表述特点来进行更多的改变，比如同位素、探测器、多帧等等。

Table 17 CT IOD，通过 ID 和信息模块建立

IE	信息模块	用法（强制的、有条件的、用户自定的）
病人	病人	M
	临床试验题目	U
检查	通用检查	M
	病人检查	U
	临床试验检查	U
序列	通用序列	M
	临床试验序列	U
参照系	参照系	M
设备	通用设备	M
图像	通用图像	M
	图像平面	M
	图像像素	M
	增强及用药	C 如果在图像中使用了增强剂则需要
	CT 图像	M
	覆盖信息平面	U
	VOI LUT ^a	U
	SOP 通常的	M

^a卷的关注查询表

DICOM 标准内部是依据面向对象的设计方法构造的。它试图创建一个自有的真实世界模型，用重要临床属性集合来定义抽象对象，并用抽象对象来表达数据。在医疗环境中的任何真实世界的实体，比如从医疗角度触发，对于病人 John Smith，你可能并不关心 John 的发型或者他家狗狗的名字。但是你一定很想知道他的姓名、ID、出生日期、性别、体重、保险公司以及与健康有关的其他重要临床信息。事实上，医疗流程所关注的，也就是好好收集那些医学成像方面使用的属性，见图 18（当然我们也关心 John 其实是一个人而不是其他东西，但是这不是为了进行社会学研究）。

这使我们必须回到 DICOM IOD 的本意。对于任何真实世界实体（病人、CT 图像、DICOM 胶片打印机）的类来说，他的 IOD 只不过是属性（组合起来存放在模块和 IE 中）的集合，这些属性为 DICOM 用途来描述整个类。当我们有一个特定病人（John Smith）、CT 图像或其他什么，通过在 IOD 属性中赋予特定的值，使这些实体变成抽象 IOD 的实例。

此外，DICOM 将所有 IOD 分为规格化（Normalized）和合成化（Composite）两类。一个规格化的 IOD 表示一个单一的、真实世界的实体，就像我们病人 IOD 表示的病人一样。所有规格化 IOD 的属性都是真实世界实体固有的。例如，DICOM 检查 IOD（DICOM Study IOD）就是规格化的并且只包含固有检查属性，比如检查日期和时间。病人姓名更适宜作为病人的属性，而不是检查的，因此在检查 IOD 中是不会出现的。

合成化的 IOD 是许多真实世界实体或者实体要素部分的混合体。例如 CT 图像 IOD，在 DICOM 中，这个 IOD 将包含许多病人属性（姓名、ID 等，以便标识图像所属的病人）以及 CT 机的属性、病人检查等等。它混合了多种真实世界实体来创建一个 CT 图像 IOD 混合体。你可能会猜到，合成化的 IOD 更适合用来记录关系、联系、过程和上下文；规格化的 IOD 用来表述单一对象。然而合成和规格化之间的界限总是比较模糊的，并且至少我们知道他们都是 IOD，并且 DICOM 对他们的处理是一致的。

综上所述，IOD 是真实世界对象的属性表达：病人、CT 图像等等。就面向对象设计而言，IOD 是对象或类。之前曾讨论过（见 5.5），实际的 DICOM 对象只不过是在 DICOM 格式中记录的 IOD 实例，其中 VR 则是用来记录数据元素名称和值的。

5.7.6

多学点

关于 DICOM 数据组成的基础概念我们已经阐述的很清楚了。现在你应该足够了解 DICOM 中从 VR 到 IOD 的所有知识了。如果你想要了解更多，DICOM 标准（PS3.3）的第 3 部分是你的最终选择。首先，那里会更细致地介绍定义特定对象和数据类型的原则；其次，还会提供 DICOM 信息模块的细节以及通过与 IOD 和 IE 绑定实现放射流程的方法。

记住，第 3 部分是标准中很大的一部分内容；超过了 1000 页每年都会增长 100 页。大多数文档中的改变都会记录在附录 A（Composite IOD，合成化的 IOD）、附录 B（规格化的 IOD）以及附录 C（信息模块）中。持续关注这些附录及其中的内容能够让你成为“吉尼斯纪录保持着”；因为他们一直在增加特定 IOD 类型。所以你需要改为根据需要挖掘 PS3.3 中的数据，比如通过关键词来进行查询。然而，DICOM 著作的演化有时会带来不好的结果；演化不能无限制地通过增长对象定义的数量来提高标准的水平。如果有 DICOM4.0，或许整个 ID、模块和 IOD 的集合会被重新组织成一个更规范、更浓缩的格式。我们一直期待着它的发生。

Chapter 6

DICOM 中的医学图像

我亲爱的读者，让我们假想一下，我们已经在 VR 和 DICOM 对象的土地上徜徉许久了，并且对于 DICOM 处理医疗数据的方法也已经有了深入了解了。这一切已经为我们旅程中下一个重要的步骤铺平了道路：看看 DICOM 是如何与医学图像一起协同工作的。显然，图像具有一些众所周知的属性（properties）（宽度、高度，每个像素的比特数），这些都可以在 DICOM 数据字典中找到，并且用显式或隐式 VR 来编码。但是最有趣的图像属性其实是图像自己，图像像素值的序列，存储在(7FE0, 0010)“像素数据（Pixel Data）”属性中，既使用 OB（用作 1 字节像素样本）编码又使用 OW（用作 2 字节像素样本）编码。

为了存储这些(7FE0,0010)像素 DICOM 支持各种图像格式。这些格式可以大略地分成两大类：

1. **DICOM 指定的：**只有 DICOM 使用这些图像格式。它们也是最古老的图像格式，是在更好的图像格式诞生之前，计算机时代的早期引入的格式。它们类似于原始的 BMP 图像，具有各种像素字节的打包方法。
2. **DICOM 所接受的独立标准格式：**包括这些著名的格式，如 JPEG、RLE（运转周期编码 run-length encoding）、ZIP 和一些名气较小的格式，如 JPEG2000、JPEG-LS。所有这些标准都涉及各类可逆的和不可逆的图像压缩技术，这些技术在医学成像领域尤其有用（主要是为了减少图像数据量）。当主要的标准（DICOM）包含其他标准（JPEG）用作特定任务（图像编码）时，模块化的标准引用是非常方便、和谐的并且非常实用。

我们将从第一种类型开始，因为他是最早的也是最古老的，并且大多时间仍然用它作为 DICOM 默认的格式。之后我们会回顾独立的标准中最重要要点。你可以在别处了解关于独立标准的更多内容，但是他们的一些属性会在医学成像存储和分析方面有绝妙的效果，所以我们会涉及这些内容。

6.1

DICOM BMPs

你可能已经了解过，数字图像是一个像素的矩阵（图片元素），不同颜色的微小点组成了图片。比如，一个典型的 CT 图像宽是 512 像素，高也是 512 像素；那么它总够有 $512 \times 512 = 262144$ 个像素。如果你把这些像素从左上角开始一行接一行地排布，那么你会得到一个 262144 个像素值的序列，你可以将它们存储在一个文件中，事实上这个文件就是你的原始 BMP 图像（图 19）。

现在让我们闭上眼睛，在脑海中列举所有我们认为重要的数字图像属性。列出来了吗？把你的和 DICOM 对比一下吧。

1. **图像宽和高：**显然，你需要了解它们。它们的乘积常常被称作图像空间分辨率，其实就是图像中像素的总数。
2. **每个像素的色标：**图像的每个像素可以是许多色标值的混合物。最典型的情况是一个彩色像素，它包括了三种独立的色标：红、绿和蓝（即 RGB 色空间）。每个色标的浓度相当于像素的亮度，他们混合在一起就产生了颜色。比如，将等量的红和绿混合会得到暗黄色；将等量的红、绿、蓝相当于灰度。灰度图像，通常用每个像素只用一个单色色标，

相当于像素的灰度流明。当使用2字节(OW VR)色标时,这个色标将能提供 $22 \times 8 = 65,536$ 个可能的灰度:在CT、MR、CR和许多其他影像设备上,DICOM能够提供极深灰度的根源所在。无论如何,像素取样的选取对于所有图像中的像素来说是保持不变的,这只由成像技术决定。

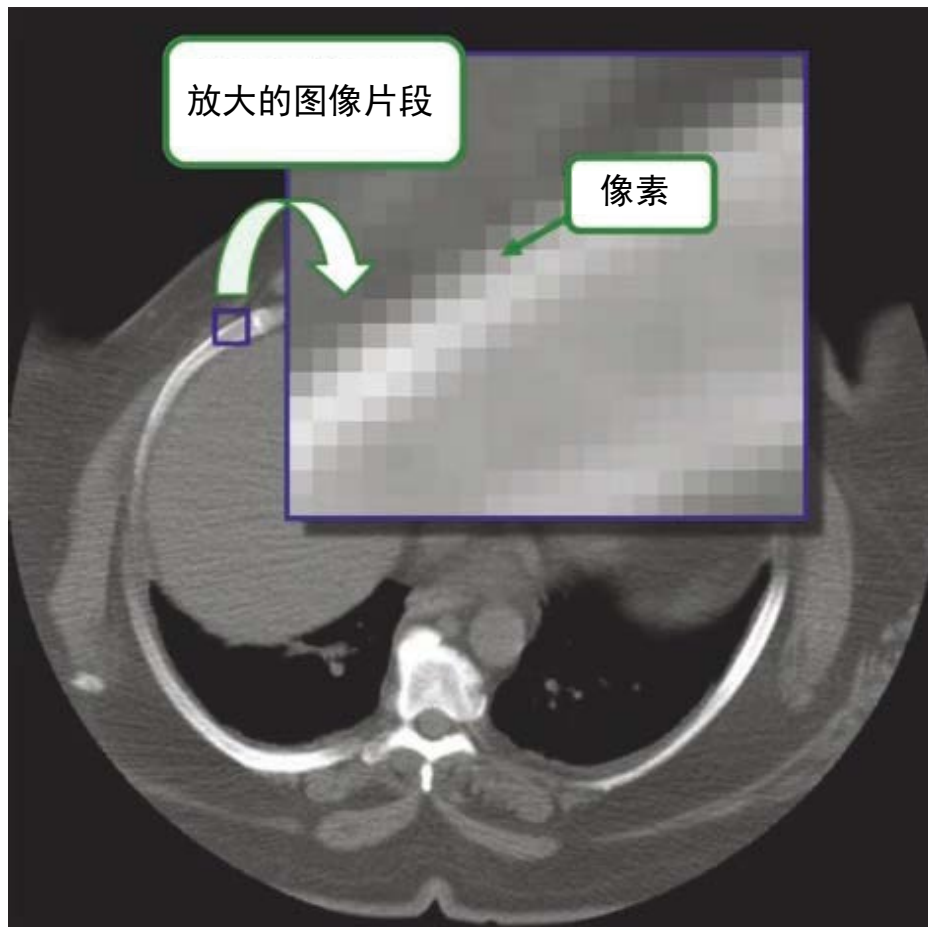


Fig.19 放大的图像像素

3. **比特通常用来存储一个像素样本**, B_s : DICOM将这个参数成为“存储用的比特”。比如,如果你有一个灰阶图像每一个像素用8比特,那么 $B_s = 8$,并且灰度共有 $2^{B_s} = 2^8 = 256$ 。如果你将存储用的比特数增加到 $B_s = 10$,那么每个像素样本将会具有 $2^{10} = 1024$ 个灰度,以此类推。如你所见,存储用的比特数的主要用途是体现图像的颜色深度(比如,彩色图像中的颜色数和灰阶图像中的灰度数);它体现了你的图像到底有多丰富。在谈论亮度时,这个属性与空间分辨率类似(图像宽度和高度),但更活跃。它在数字医疗领域尤其有价值也尤其重要。因为在数字医疗领域你经常需要分辨颜色和亮度的微小变化。
4. **每个像素样本分派的比特**, B_a 实际上是 B_s 的8倍(保证占满整个字节)。这就是计算机内存存储像素样本所需要的空间。显然, $B_s \leq B_a$ 。
5. **最高位的比特**, B_h : 我们很快就会知道,这个比特相当于在 B_a 段中 B_s 像素样本的结尾。

图20总结了多样本DICOM像素的结构:我们的像素是有三个样本组成的(红、绿、蓝),并且每个样本在这个特定的例子中都有 $B_s = 12$ 比特。因为所有计算机数据都是用字节来存储的(1字节=8比特),12比特取整就是 $B_a = 16$,并且 $B_h = 11$ 。通常 $B_h = B_a - 1$ (比特是从0开始计算的)。同样的16比特存储会分派给其他的样本(比如我们例子中的红和蓝),因此整个图片实际上是用他的像素样本的序列组合成的。我希望让大家理解起来不是太困难,但是为了让图

像呈现一个整体，让我给你展示一下其他像素样本编码例子吧，见图21。在这种情况下，一个像素存储在 $B_s=10$ 比特的存储中，却是从二号比特开始计算。因为DICOM需要存储最高位的比特 B_h 和比特存储 B_s 的数值，你可以用 B_h+1-B_s 来找回“比特开始”位置而其他的也以此类推。

在所有的“怀旧”像素编码中，最棘手的部分是DICOM能如何利用那些没有上的比特。请看第二个例子：

$B_a - B_s = 6$ 比特

(0-1比特和12-15比特) 没有包含任何像素数据；因此我们实际上浪费了 $6/16=37.5\%$ 的存储空间。为了补偿这些空间，DICOM会选择在其中存储附加信息，比如图像覆盖的像素（这与其原始的图像像素数据没有什么关系）、实际上，一个信息流的片段将被其他片段塞满，最终还是使用了每个字节中的所有比特。

在DICOM的早期岁月，那时图像压缩仍然处于萌芽期，在像素样本之间存储其他数据是一种非常酷的优化存储方案。你可以想象，事实上它导致了很多人莫名其妙的比特大杂烩，最终导致这种做法被抛弃。现在，我们会通过使用图像压缩技术来节约更多存储，而很少再采用在像素内的比特中插入数据的方法。然而，“比特挤压”技术仍然被DICOM设为标准，并且能够在许多应用程序和数据中找到，特别是当你在摆弄一个比较老的DICOM装置的时候。DICOM PS3.5的附录D中介绍了其他可能的样本编码语法。

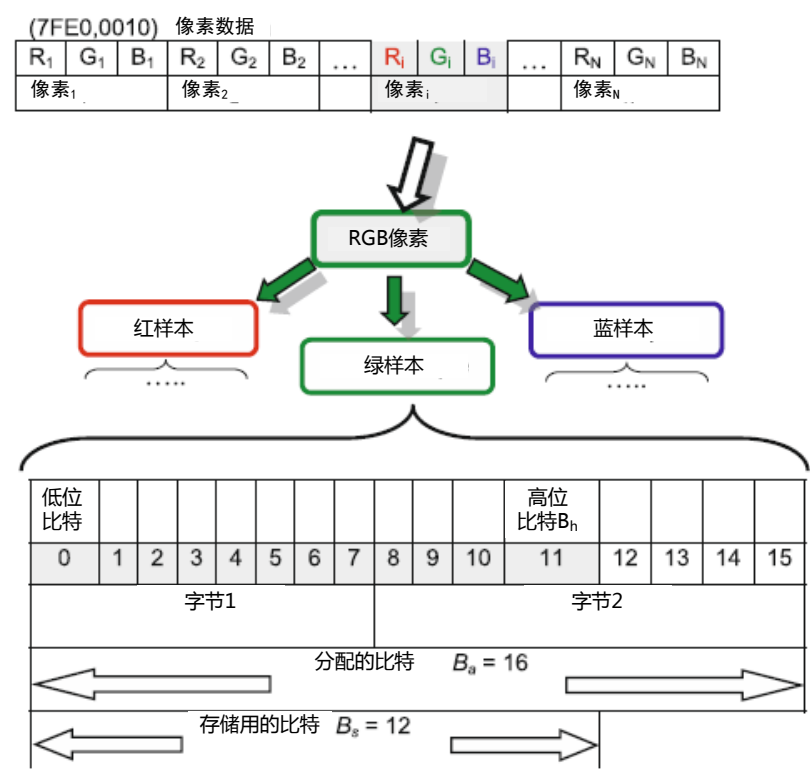


Fig. 20 Storing pixel sample bits: basic and most common case

存储像素样本的比特：基本的和最常见的情况

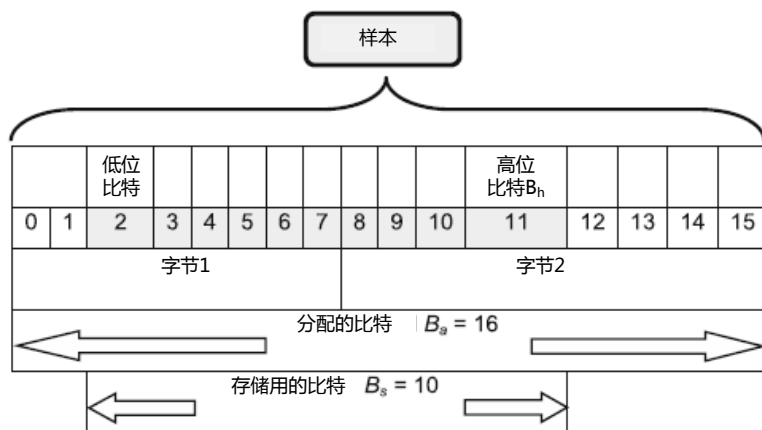


Fig. 21 Storing pixel sample bits: more complex example

存储像素样本的比特：更复杂的例子

如果你能用不同的记录方法来处理图像像素（样本），那么你就知道如何来读写DICOM BMP图像了。如前所述，DICOM对图像的处理方法与其它信息类似，所有重要的图像属性都会转化为DICOM对象中的DICOM VR。这些属性必须在每个图像所属的DICOM对象中体现出来。在DICOM中，这些属性是必须的。表18提供了更细致的重要图像属性的快照。这些属性来自DICOM数据字典。这只不过是所有DICOM像素相关标签的一小部分，但是如果你具有处理其它图像格式的经验，那么你应该希望获得DICOM的全部内容。比如，如(0028,0008)的建议，你可以在一个简单的DICOM图像中存储一个图像帧（特别是视频）的序列。你可以在(0028,0030)中指定像素的实际物理尺寸，这个标签内允许用的物理单位来测量图像对象，比如用厘米。记录像素间距和图像间距（存储在(0018,0088)中，并且MR或CT的所有图像序列都会用到），主要是为了满足三维重建（我们技师说应该叫三维重组）的需要，因为我们需要保证重建对象具有正确的大小。我不确定是否还要继续介绍这个列表，就到这里吧，下面进行一下总结。

Table 18 DICOM数据字典中的重要图像属性

Tag 标签	Name 名称	VR	VM
(0028, 0002)	Samples per Pixel 每个像素的样本	US	1
(0028, 0004)	Photometric Interpretation 光度解释	CS	1
(0028, 0008)	Number of Frames 帧数	IS	1
(0028, 0010)	Rows 行	US	1
(0028, 0011)	Columns 列	US	1
(0028, 0030)	Pixel Spacing 像素间距	DS	2
(0028, 0100)	Bits Allocated B_a	US	1

	分配的比特		
(0028, 0101)	Bits Stored B_s 存储用的比特	US	1
(0028, 0102)	High Bit B_h 最高位的比特	US	1
(0028, 0103)	Pixel Representation 像素表示法	US	1
(7FE0, 0010)	Pixel Data 像素数据	OW/OB	1

DICOM在存储图像数据以及相关参数方面提供的支持极其丰富，除此之外没有一个图像格式能够做到。这就是DICOM能够在医学图像领域成功的原因。同样的，我们也就了解了为什么将DICOM图像输出成为其他“常见”格式总是会出现问题，比如造成丢失关键图像信息或者导致错误理解图像。除此之外，也就没有什么其他新奇之处了。DICOM支持JPEG且其内部支持其他图像压缩方法。这些压缩方法可以将(7FE0,0010)“像素数据”属性中存储的DICOM对象进行像素缓存的压缩，但是并不会影响任何其他的属性数据。让我们看看这些压缩方法到底是怎么回事吧。

6.2

图像压缩

一个典型的 CT 图像，宽=高=512，灰阶的（每个像素只有一个样本），且存储用比特=12。计算机需要多少存储来储存这个图像呢？让我们来计算一下吧。为了存储 12 比特，首先需要分配 2 个字节（16 比特）。像素的总数为 512×512 ，因此存储图像的字节总数是：

$$512 \times 512 \times 2 = 524,288$$

因此我们需要半兆字节来存储像素数据，这些数据会存在 DICOM CT 对象的(7FE0,0010)“像素数据”属性中。然而，CT 图像不是只有一幅的；通常会是一组图像序列或检查，每个序列或检查都会包含成百乃至上千的图像。在本例中，典型的 CT 图像将占用 100MB 的空间。如果用拨号网络下载，则需要将近 6 个小时的时间，在宽带网络上也需要好几分钟（表 19）。几天这样的检查就能轻松地塞满你整个硬盘，挤爆任何远程影像项目。对于这样的图像卷你应该如何应对？

一个典型的回答是“买个更好的计算机”，或“用更快的网络”。就算升级硬件永远是一个好主意，但在本例中它只能解决一个问题：如何帮你花钱。看看在数字图像获取设备的品质和分辨力方面这些突飞猛进吧。数字放射摄影（DR），计算机放射摄影（CR），多层螺旋 CT，数字乳腺摄影，和超声设备一直在提升分辨率和产生的图像总数，这不可避免地带来了更大个的图像和更大个的检查。如果你计划展开一份有竞争力且可持续发展的医疗事业，并且你调研了各种应用程序比如远程影像，那么不使用图像压缩技术的话你会死的很惨的。

图像压缩技术从一开始就已经包含在 DICOM 标准中了。图像压缩瞄准(7FE0,0010)的图像像素字节并且轻巧地将它们重新安放在一些更简短的形式中。这将很明显地减少原始图像的大小（比老式的比特挤压要好得多），它不仅节约了存储空间而且通常会缩短图像下载的时间。我们说起过 DICOM 并没有创造一种其自己的图像压缩技术。而是囊括了几乎所有知名的图像压缩算法比如 RLE，JPEG，JPEG2000，JPEG-LS，和 ZIP。所有这些算法的都是各自独立发展起来的，最终导致了不同的国际标准组织（ISO）标准和应用程序；DICOM 只是简单地采用这些标准和应用程序。所有压缩的(7FE0,0010)像素缓存的 DICOM 数据对象都是使用显式 VR 小端字节序类型编码的（见 5.5.1）

图像压缩的使用会在图像呈现以及 PACS 的整体表现方面带来显著的效果，因此不管你当前在临床项目中的角色是什么，你都需要了解以下压缩的基础知识。数据压缩的根本方法一点都不难。任何压缩算法都是试图找到和修剪最多余和最啰嗦的信息，这样就能使数据变得更短小。修剪的效率是用下面的压缩率来衡量的：

$$R_{\text{comp}} = \frac{\text{原始数据大小}}{\text{压缩后数据大小}}$$

更高的压缩率表示获得了更好的压缩效果。每个压缩算法都会提供它自己达到最大 R_{comp} 的策略，但无论是概念上的还是实际的，所有压缩技术都必须是这两类中的一类：无损的（可逆的）和有损的（不可逆）压缩。

6.2.1

无损压缩

无损压缩并不真正改变图像。在你压缩或解压图像之后，你总是能获得原始的、一个像素挨一个像素的图像。这是通过一种聪明的重组和重命名像素字节的方法实现的。只要抓住一个重点，请看下面的像素值序列：

1000, 1001, 1002, 1002, 1000, 1000, 1001, 1057,....

一种典型的无损压缩算法是找出那些最频繁出现数值的重复性，像 1000 就可以用更短

的符号来代替。比如，如果我们用“a”来代替“1000”，那么上面的像素串就变成了：

a, 1001, 1002, 1002, a, a, 1001, 1057,....

因为“a”就比“1000”要短，我们整个数据串会变得更短，只要我们记住“a”代替了“1000”，我们就一定能解压缩这些字串，得到原始的数值。这种压缩方法就是著名的“可变长度编码”。现在来回顾一下：如果我们将原始数据中重复最多的元素进行处理，其实就是用最短的符号来替换他们。我们能够得到实质上的数据量减少，但是我们将总能够恢复数据。那么，我们实际上就发现了霍夫曼（Huffman）压缩算法，在可变长度编码中最流行的算法。这个准确的方法。所有自然的图像都有重复的内容可用来压缩。

首先，相邻的像素总是趋向于相似的数值；其次，一个图像会有一个具有相同像素值的很大的区域（比如黑色背景）。固有的冗余带来了无损压缩的成功。当在可用的图像维度（通常二维图像为x和y，三位图像集合则为x、y、z，比如CT断层图像）中发现这些冗余时，就可以进行更多的压缩了。然而，探索像素的重复性只能依照明确的方向对普通的医疗图像进行扩展， R_{comp} 的数值将在 2 和 4 之间；表示压缩后图像大小将在原始图像的 1/2 大小和原始图像的 1/4 大小之间。这已经很棒了，但是如果选择了有损压缩选项，那么这个压缩结果就显得不那么光鲜了。

6.2.2

有损压缩

有损压缩，人如其名，牺牲了部分原有信息来达到更高的 R_{comp} 值。这种牺牲可以获得额外的数据冗余，因此是无损压缩步骤的一种有效补充。让我们看看前面那个相同的像素序列吧：

1000, 1001, 1002, 1002, 1000, 1000, 1001, 1057,....

如果像素密度大约是 1000，你真的能在视觉上分辨出 1000 和 1001 的区别吗？估计是不行，因为在像素亮度上只有 0.1%的区别。因此，我们可以稍微修改一下这个序列，选择一个灰阶作为可接受的容差然后将 1001 用 1000 替换：

1000, 1000, 1002, 1002, 1000, 1000, 1000, 1057,....

如果我们对上面这个序列进行无损压缩，那么它就会变成：

a, a, 1002, 1002, a, a, a, 1057,....

这个结果比之前纯粹的无损压缩要短多了。换句话说，如果无损压缩利用了相同的像素，那么有损压缩就将它扩展到了具有不易察觉容差的近似像素值上。

信不信由你，当前最时髦的有损压缩算法可以有高达 100 的 R_{comp} 。对于普遍的医疗图像，在有损压缩中合理的 R_{comp} 的值大约为 10，有时可达到 20。有损压缩中的 R_{comp} 值是由可察觉的容差所决定的。在前例中，我们压缩了一个灰阶，但是如果使用两灰阶容差则会带来更好的压缩效果：

a,a,a,a,a,a, 1057,....

显然，容差不能无限地增大。有些情况下这种容差可能会被人看出来，在有损压缩图像上有可见的加工痕迹。在高 R_{comp} 和可见的图像梯降之间达到有损压缩的平衡点，已经成为了一种艺术，但请记住以下几点：

1. **有损压缩可以导致法律方面的争执。**如果在一个程序中使用有损压缩，DICOM（以及食品药品监督管理局 FDA）要求所有有损压缩的图像本身必须有所标记。
2. **计算机辅助诊断（Computer-aided diagnosis CAD）问题。**CAD 越来越流行了，计算机和计算机软件在医疗图像分析中起到了越来越大的作用。容差不会被人眼察觉，但是将会被 CAD 软件察觉，并且可能导致破坏性的后果。

CAD 和有损压缩

你的CAD程序，可能成为一个评判有损压缩适当程度的客观工具。如果CAD程序在原始图像压缩前后都得到了相同的结果，那么你也许可以用这个结果作为图像没有被过度压缩的象征，至少有损 R_{comp} 没有对CAD分析带来干扰。

目前，有损图像压缩技术在各类远程放射诊断系统中非常受欢迎，因为在这些系统中图像必须远距离传输，但有时不确定的带宽可能会带来网络瓶颈。但即使在远距离传输的情况下，压缩率也是需要根据需要调节的。去年，我见过一个远程放射诊断服务对医学图像交换使用了 70%质量的有损 JPEG 压缩。也许你不知道，即使是你宠物的图像，也最好不要使用这种压缩方式。有损压缩中的加工痕迹会在质量保证层面变得相当突出。如果你计划在你的机构中使用有损压缩，一定要请受过训练的放射影像专家对每种类型或设备的图像进行评估，然后确定这种损失确实是无法察觉的。

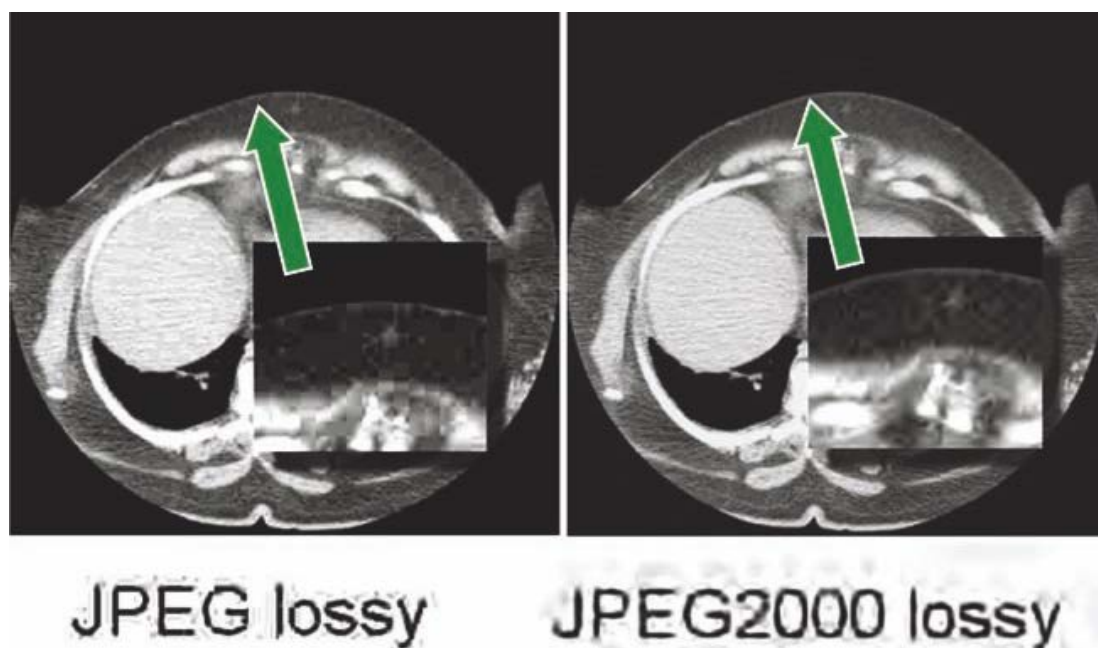


图 22 描绘了过度有损压缩后的图像和文字。过度进行JPEG压缩会带来高可见的块状加工痕迹（左图）；过度进行JPEG2000压缩会带来图像模糊（右）

过多的有损压缩会导致图像较高的可见加工痕迹，如你在图 22 中看到的一样。左边的图像是经过JPEG过分压缩处理的，这导致了臭名昭著的JPEG方格加工痕迹，即可以看到的色彩不连续的众多矩形区域。在右侧的相同图像使用了大剂量的JPEG2000 有损压缩，这种压缩方法以可见的图像模糊代替了方格。两个图像的 R_{comp} 都差不多是 70。如果将压缩程度降到原始图像的十分之一，则会带来更好的效果，表面上看起来将和原始图像一模一样。

记住使用有损压缩的职责在于放射影像专家的权衡；如果由于图像质量的下降造成了错误的解读，那么放射影像专家是应该负责责任的。PACS 厂商只对这些问题负责：

1. 持续提供未压缩的图像并且提供压缩选项。是否选择有损压缩是你的事情
2. 当工作站上显示图像时，必须清晰地标注出那些经过有损压缩的图像。因此，如果你对图像质量不满意，那么你应该能够重新调取那些未压缩的图像（请与你的 PACS 供应商确认这个问题）。

此外，重复应用有损压缩会带来图像质量的严重下降。如果你让一幅原始的CT图像以适度的压缩比率（比如， $R_{comp} < 10$ ）经过了一次先有损压缩再解压缩的过程，那么解压后的图像可能和原始图像看起来一模一样（有经验的放射影像专家可能看不出两者的区别）。然而，

如果你重复对该图像进行压缩和解压缩（比如为了存储或者网络传输），那么质量的损失和压缩加工痕迹实惠累积的。这很容易出现在一长串内部联通的图像设备上，每个设备都启用了有损压缩的功能。当图像从它们这条生产线上经过之后，是根本无法用来诊断的。因此，如果你启用了有损压缩功能，那么一定只在最后一个环节（比如，长时间图像归档或给最终用户显示用的工作站）使用它。绝不要在图像传送链上的中间环节使用有损压缩。

噪声和压缩

噪声较多的图像，比如薄层 CT，是很难进行无损压缩的；噪声破坏了像素之间的相似性（冗余性）。另一方面，有损压缩将像素值平均化，使他们变得更加相似，在很多情况下相当于图像降噪算法。这可以证明有损压缩用在传统的噪声较大的影像设备上可行的，比如超声。

6.2
图像压缩

一个典型的 CT 图像，宽=高=512，灰阶的（每个像素只有一个样本），且存储用比特=12。计算机需要多少存储来储存这个图像呢？让我们来计算一下吧。为了存储 12 比特，首先需要分配 2 个字节（16 比特）。像素的总数为 512×512，因此存储图像的字节总数是：

$512 \times 512 \times 2 = 524,288$

因此我们需要半兆字节来存储像素数据，这些数据会存在 DICOM CT 对象的(7FE0,0010) “像素数据” 属性中。然而，CT 图像不是只有一幅的；通常会 是图像序列或检查，每个序列或检查都会包含成百乃至上千的图像。在本例中，典型的 CT 图像将占用 100MB 的空间。如果用拨号网络下载，则需要将近 6 个小时的时间，在宽带网络上也需要好几分钟（表 19）。只需要几天时间，这样的检查就能轻松地塞满你整个硬盘，挤爆任何远程影像项目。对于这样的图像卷你应该如何应对呢？

Table 19 数字图像和检查的典型大小

影像设备	典型图像矩阵(高、宽、 每个像素字节数)	图像大小,千字节(KB)	一个检查中的典型图像数 ^a	典型的金叉大小,兆字节 (MB)
NM	128 × 128 × 1	16	100	1.5
MR	256 × 256 × 2	128	200	25
CT	512 × 512 × 2	512	500	250
超声	600 × 800 × 3	1400	500	680
CR	2140 × 1760 × 2	7356	4	30
彩色 3D重建 ^b	1024 × 1024 × 3	3000	20	60
数字乳腺 X 线摄影	最大 6400 × 4800 × 2	60000	4	240

^a由于这个变化与设备、检查和扫描协议关系极其密切，因此我们提供了最常规“数量级”的数字

^b “二次获取”重建，现在越来越多地由CT和MR影像设备提供。

一个典型的回答是“买个更好的计算机”，或“用更快的网络”。就算升级硬件永远是一个好主意，但在本例中它只能解决一个问题：如何帮你花钱。看看在数字图像获取设备的品质和分辨力方面这些突飞猛进吧。数字放射摄影（DR），计算机放射摄影（CR），多层螺旋CT，数字乳腺摄影，和超声设备一直在提升分辨率和产生的图像总数，这不可避免地带来了更大个的图像和更大个的检查。如果你计划展开一份有竞争力且可持续发展的医疗事业，并且你调研了各种应用程序比如远程放射诊断，那么不使用图像压缩技术的话你会死的很惨的。

图像压缩技术从一开始就已经包含在 DICOM 标准中了。图像压缩瞄准(7FE0,0010)了图像像素字节并且轻巧地将它们重新安放在一些更简短的形式中。这将很明显地减少原始图像的大小（比老式的比特挤压要好得多），它不仅节约了存储空间而且通常会缩短图像下载的时间。我们说起过 DICOM 并没有创造一种其自己的图像压缩技术。而是囊括了几乎所有知名的图像压缩算法比如 RLE, JPEG, JPEG2000, JPEG-LS, 和 ZIP。所有这些算法都是各自独立发展起来的，最终导致了不同的国际标准组织(ISO)分别掌握这些标准和应用程序; DICOM 只是简单地采用这些标准和应用程序。所有压缩的(7FE0,0010)像素中缓存的 DICOM 数据对象都是使用显式 VR 小端字节序类型进行编码的（见 5.5.1）。

图像压缩的使用会在图像呈现和 PACS 的整体表现方面带来显著的效果，因此不管你当前在临床项目中的角色是什么，你都需要了解以下压缩的基础知识。数据压缩的根本方法一点都不难。任何压缩算法都是试图找到和修剪最多余和最啰嗦的信息，这样就能使数据变得

更短小。修剪的效率是用下面的压缩率来衡量的：

$$R_{\text{comp}} = \frac{\text{原始数据大小}}{\text{压缩后数据大小}}$$

更高的压缩率表示获得了更好的压缩效果。每个压缩算法都会提供它自己达到最大 R_{comp} 的策略，但无论是在概念上还是在实际操作中，所有压缩技术都必须是这两类中的一类：无损的（可逆的）和有损的（不可逆）压缩。

6.2.1

无损压缩

无损压缩并不真正改变图像。在你压缩或解压图像之后，你总是能获得原始的、一个像素挨一个像素的图像。这是通过一种聪明的重组和重命名像素字节的方法实现的。只要抓住一个重点，请看下面的像素值序列：

1000, 1001, 1002, 1002, 1000, 1000, 1001, 1057,....

一种典型的无损压缩算法是找出那些最频繁出现数值的重复性，像 1000 就可以用更短的符号来代替。比如，如果我们用“a”来代替“1000”，那么上面的像素串就变成了：

a, 1001, 1002, 1002, a, a, 1001, 1057,....

由于“a”比“1000”要短，因此我们整个数据串会变得更短，只要我们记住“a”代替了“1000”，我们就一定能解压缩这些字串，得到原始的数值。这种压缩方法就是著名的“可变长度编码”。现在来回顾一下：如果我们将原始数据中重复最多的元素进行处理，即用最短的符号来替换他们，那么我们就能够得到实质上减少数据量，但是我们永远可以恢复这些数据。那么，我们实际上就发现了霍夫曼（Huffman）压缩算法，在可变长度编码中最流行的算法。所有自然的图像都有重复的内容可用来压缩。

首先，相邻的像素总是趋向于相似的数值；其次，一个图像会有一个具有相同像素值的很大的区域（比如黑色背景）。固有的冗余带来了无损压缩的成功。当在可用的图像维度（通常二维图像为x和y，三位图像集合则为x、y、z，比如CT断层图像）中发现这些冗余时，就可以进行更多的压缩了。然而，探索像素的重复性只能依照明确的方向对普通的医疗图像进行扩展， R_{comp} 的数值将在 2 和 4 之间；表示压缩后图像大小将在原始图像的 1/2 大小和原始图像的 1/4 大小之间。这已经很棒了，但是如果选择了有损压缩选项，那么这个压缩结果就显得不那么抢眼了。

6.2.2

有损压缩

有损压缩，人如其名，牺牲了部分原有信息来达到更高的 R_{comp} 值。这种牺牲可以获得额外的数据冗余，因此是无损压缩步骤的一种有效补充。让我们看看前面那个相同的像素序列吧：

1000, 1001, 1002, 1002, 1000, 1000, 1001, 1057,....

如果像素密度大约都是 1000，你真的能在视觉上分辨出 1000 和 1001 的区别吗？估计是不行，因为在像素亮度上只有 0.1%的区别。因此，我们可以稍微修改一下这个序列，选择一个灰阶作为可接受的容差然后将 1001 用 1000 替换：

1000, 1000, 1002, 1002, 1000, 1000, 1000, 1057,....

如果我们再对上面这个序列进行无损压缩，那么它就会变成：

a, a, 1002, 1002, a, a, a, 1057,....

这个结果比之前纯粹的无损压缩要短多了。换句话说，如果说无算压缩是利用了相同的像素，

那么有损压缩就将它扩展到了具有不易察觉容差的近似像素值。

信不信由你，当前最时髦的有损压缩算法可以有高达 100 的 R_{comp} 。对于一般的医疗图像，在有损压缩中合理的 R_{comp} 的值大约为 10，有时可达到 20。有损压缩中的 R_{comp} 值是由可察觉的容差所决定的。在前例中，我们压缩了一个灰阶，但是如果使用两灰阶容差则会带来更好的压缩效果：

a,a,a,a,a,a,a, 1057,....

显然，容差不能无限地增大。有些情况下这种容差可能会被人看出来，在有损压缩图像上有可见的加工痕迹。在高 R_{comp} 和可见的图像梯降之间达到有损压缩的平衡点，已经成为了一种艺术，但请记住以下几点：

1. **有损压缩可以导致法律方面的争执。**如果在一个程序中使用有损压缩，DICOM（以及食品药品监督管理局 FDA）要求所有有损压缩的图像本身必须有所标记。
2. **计算机辅助诊断（Computer-aided diagnosis CAD）问题。**CAD 越来越流行了，计算机和计算机软件在医疗图像分析中起到了越来越大的作用。容差不会被人眼察觉，但是将会被 CAD 软件察觉，并且可能导致破坏性的后果。

CAD 和有损压缩

你的CAD程序，可能成为一个评判有损压缩适当程度的客观工具。如果CAD程序在原始图像压缩前后都得到了相同的结果，那么你也许可以用这个结果作为图像没有被过度压缩的象征，至少有损 R_{comp} 没有对CAD分析带来干扰。

目前，有损图像压缩技术在各类远程放射诊断系统中非常受欢迎，因为在这些系统中图像必须远距离传输，但有时不确定的带宽可能会带来网络瓶颈。但即使在远距离传输的情况下，压缩率也是需要根据需要调节的。去年，我见过一个远程放射诊断服务对医学图像交换使用了 70%质量的有损 JPEG 压缩。也许你不知道，即使是你宠物的图像，也最好不要使用这种压缩方式。有损压缩中的加工痕迹会在图像品质层面上变得相当地明显。如果你计划在你的机构中使用有损压缩，一定要请受过训练的放射影像专家对每种类型或设备的图像进行评估，然后确定这种损失确实是无法察觉的。

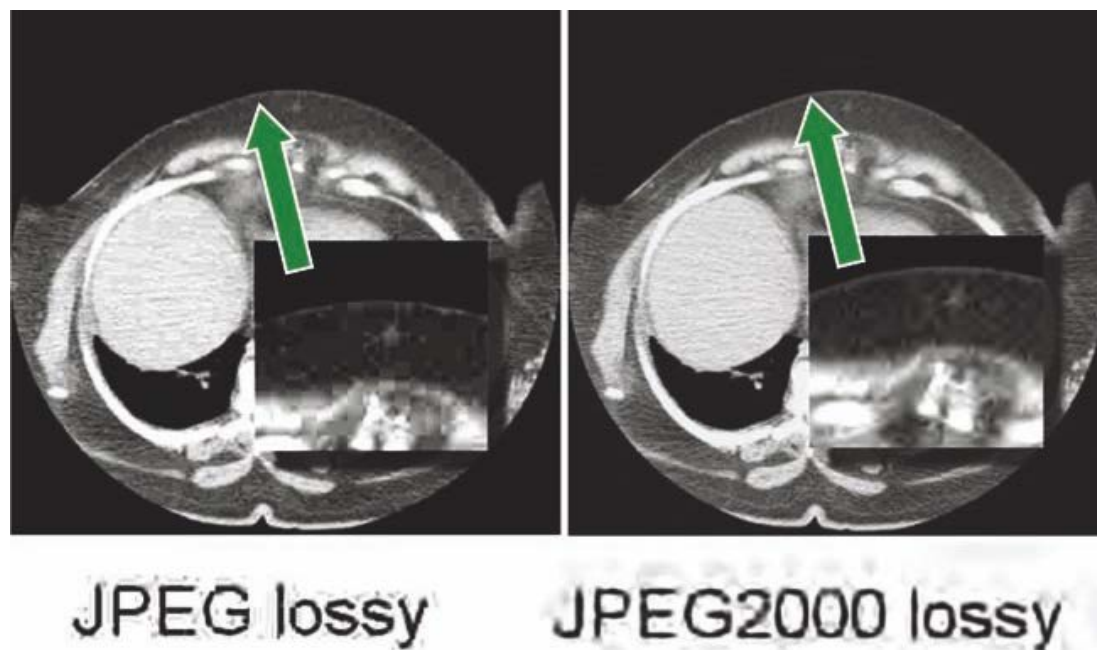


图 22 描绘了过度有损压缩后的图像和文字。过度进行JPEG压缩会带来高可见的块状加工痕迹（左图）；过度进行JPEG2000压缩会带来图像模糊（右图）

过多的有损压缩会导致图像较高的可见加工痕迹，如你在图 22 中看到的一样。左边的图像是经过JPEG过分压缩处理的，这导致了著名的JPEG方格加工痕迹，即可以看到的色彩不连续的众多矩形区域。在右侧的相同图像使用了压缩程度很大的JPEG2000 有损压缩，这种压缩方法以可见的图像模糊代替了方格。两个图像的 R_{comp} 都差不多是 70。如果将压缩程度降到原始图像的十分之一，则会带来更好的效果，表面上看起来将和原始图像一模一样。

记住是基于放射影像专家的权衡才能决定使用有损压缩的具体方式；如果由于图像质量的下降造成了错误的解读，那么放射影像专家是应该负责任的。PACS 厂商只对以下这些问题负责：

1. 不断地提供未压缩的图像并且提供压缩选项。是否选择有损压缩是你的事情
2. 当工作站上显示图像时，必须清晰地标注出哪些是经过有损压缩的图像。因此，如果你对图像质量不满意，那么你应该能够重新调取那些未压缩的图像（请与你的 PACS 供应商确认这个问题）。

此外，重复应用有损压缩会带来图像质量的严重下降。如果你让一幅原始的CT图像以适度的压缩比率（比如， $R_{comp} < 10$ ）经过了一次先有损压缩再解压缩的过程，那么解压后的图像可能和原始图像看起来一模一样（有经验的放射影像专家可能看不出两者的区别）。然而，如果你重复对该图像进行压缩和解压缩（比如为了存储或者网络传输），那么质量的损失和压缩加工痕迹是会累积的。这中情况很容易出现在一长串内部联通的图像设备上，其中每个设备都启用了有损压缩的功能。当图像从它们这条生产线上经过之后，是根本无法用来诊断的。因此，如果你启用了有损压缩功能，那么一定只在最后一个环节（比如，长时间图像归档或给最终用户显示用的工作站）使用它。绝不要在图像传送链上的中间环节使用有损压缩。

噪声和压缩

噪声较多的图像，比如薄层 CT，是很难进行无损压缩的；噪声破坏了像素之间的相似性（冗余性）。另一方面，有损压缩将像素值平均化，使他们变得更加相似，在很多情况下相当于图像降噪算法。这可以证明有损压缩用在传统的噪声较大的影像设备上可行的，比如超声。

6.2.3

流式压缩

流式（渐进），按需压缩看起来是标准压缩的一种智能扩展。流式有点像著名的缩略图，你应该已经在网络上（比如，“[点击这里可以放大图像](#)”）不止一次看到了。考虑到后面的例子：你需要下载并回顾一下 CR 图像。你显示器的分辨率是 1024×768 像素，而标准 CR 图像的大小为 2000×1500 像素；因此，CR 图像大概是你显示器长宽的两倍。你如何放大缩小图像并不重要，主要是依 CR 图像的原始分辨率进行判断，它永远不可能完全适合你的显示区域。如果你的网络不是很快，为什么要为一次性调取整个图像而烦恼呢？实际上你根本不可能完整的显示它。

流式压缩跟踪当前图像显示区域，持续补充正在显示的图像上某部分的高分辨率细节（图 23）。因为高分辨率细节是最难压缩的，只有需要的时候才调取它们可以减少调取图像的大小以及下载图像的时间。刚开始时，流式压缩向用户发送一个可容忍范围内的最低级别细节，仅仅是为了满足用户显示器的分辨率。相较于全细节的原始图像，这样处理的话图像能够被更快速地调取。那么，当用户放大缩小或者移动特定的图像区域时，流式压缩会只针对这个区域立即调取其余的，高分辨率的图像细节。

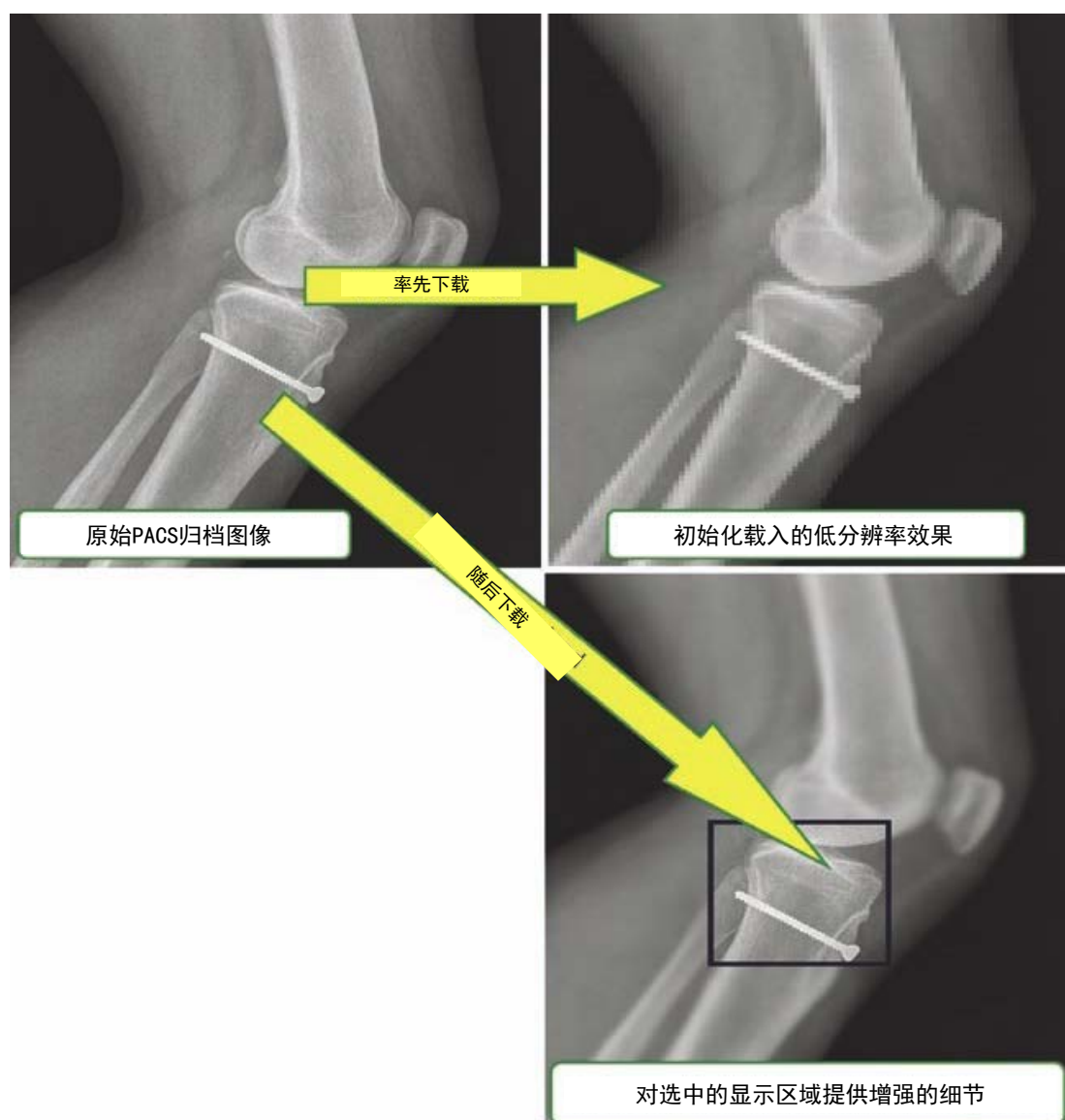


Fig. 23 用流式压缩调取高分辨率图像：首先整个图像呈现低分辨率（减少下载量），之后根据用户选择的显示区域，按要求锐化本地图像细节

这可以在实质上提高图像下载速度，因为：

1. 一个用户通常某时只浏览特定图像区域且并不需要高分辨率的整个图像副本。相似的，浏览一个图像序列时，一个用户每次只会浏览一部分图像。
2. 一个用户几乎立即就能使用图像并开始工作了，不需要等待下载完全部高分辨率图像。
3. 将整个图像下载过程分解为当前所需的小细节块，对于用户来说将明显地缩短下载时间。

这些特性使得流式压缩在远程放射诊断系统方面非常流行，对于较大的图像，远程放射诊断系统能够借此获得额外的（且通常为实质的）时间收益。流式压缩唯一的缺点来自于它的主要优点：在图像显示期间分布式地下载整个图像。当用户想要浏览图像的不同区域时，定位到每一个新的区域都需要进行一个新的、获得高分辨率细节内容的下载过程。这些细小而重复的下载过程将会降低定位效果，甚至更糟，这些下载过程可能持续地改变图像的样子；那些模糊、低分辨率的部分将永远没法变得更锐利，除非新的细节下载完成。对于许多放射影像专家来说这种体验非常不爽，我就见过有人用过流式压缩系统后，又换回标准的非流式

压缩方案。

“是的”，其中一个放射影像专家这样对我说：“我知道我将不得不因为下载非流式压缩图像而等几分钟。但是至少这个问题只在开始时发生，之后我就可以对整个检查做任何操作了。此外，我能够在最初图像初始化和漫长的非流式下载时做其他事情——核实一些报告；在电话里口授一些内容——而用了流式压缩后我不得不在一些微小的但没有任何产出的事物上浪费时间。”

尽管如此，普通大小的图像（检查）在有限的网络带宽下传送时，流式图像压缩还是很有优势的。请一定要在你所在的特定环境中来评判它的潜在益处，然后再决定是否使用它。就在成书之时，DICOM 还不支持任何流式压缩，但是在未来它一定会出现的。某些静态图像压缩标准，比如 JPEG2000，对于不同图像区域可以提供变化的压缩等级，因此可以在压缩时决定哪些区域更加重要以及在压缩时应该保留的最好细节。

6.2.4

选择正确的压缩技术

许多当前的 DICOM 应用程序都能支持各种花里胡哨的图像压缩算法，并且仅仅点击几下就能够开启或关闭这些算法。在你这样做之前应该仔细考虑这个重要的问题。

首先，你必须在有损压缩和无损压缩技术之间做出选择。如果你需要为诊断提供原始图像或者你会将图像进行短期存储（磁盘空间不是问题），无损压缩再好不过了。这可以保证图像保持原样。RLE，无损 JPEG，无损 JPEG2000，无损 JPEG-LS，以及 ZIP 压缩算法都不会更改原始图像。

如果你在读片上要求不高，如果你使用远程放射诊断项目（正在与网速拼搏），或者如果你需要长期存储图像，那么你不妨选择有损压缩。当然，假如你的网络带宽或图像归档大小阻碍了你的业务，那么你就不得不这样做了。在应用有损压缩（比如有损 JPEG 或 JPEG2000）时，经验法则是保持 10:1 的 R_{comp} （比如 $R_{comp} < 10$ ），你可以通过软件控制 R_{comp} 的值。事实上，DICOM 压缩方面的一个主要的缺点是缺少对于 R_{comp} 的 DICOM 控制。在 DICOM 标准中，没有设置或校验有损压缩比率的内容；你只能选择压缩算法。因此对 R_{comp} 的支持工作就交给了你的 DICOM 软件厂商，它们也许会提供这类功能选项。

如何确定你的软件所使用的 R_{comp}

如果你的软件允许（通常都允许）在磁盘上存入两个相同的 DICOM 图像，第一个打开压缩选项，而第二个关闭压缩选项；那么图像文件大小的压缩比例就很容易算出了，未压缩的大小除以压缩后大小，将会是一个获得 R_{comp} 的好方法。

无论你选择什么压缩方法，存储图像时绝不要使用厂商私有的压缩方法；我们不要在未来的数据其迁移问题上找麻烦。

其次，压缩将数据打包为一个更短小的格式，但是不能提供显示功能，你的软件总是不得不解压缩它来得到图像。你常常会听到这样的事：“图像压缩得更小，可以以更快的速度显示图像。”这完全是错误的。

使用图像压缩意味着两件事情：

1. **打包和解包压缩的图像数据是要花费时间的。**它明显会降低图像显示的速度，并且完全能够测算出来，甚至可能降低你计算机上其他运行程序的速度。因此，减少存储量和增加下载次数的优点总是显得不够突出，主要是因为受制于随之增加的处理时间。通常这不是一个大问题（如此看来，总之你有理由购买更快的计算机了！），但是一些压缩技术（比如 JPEG2000）对比其他的技术（比如 JPEG-LS）在压缩和解压缩时需要更长的时间。

2. 当图像到达目的地并且需要进行显示时，它们将必须被解压缩。这意味着所有 (7FE0,0010) 缓存内容将变回它们原始的数据大小，有多大变多大。因此，压缩并不会让你的图像显示工作站更加高效；这些图像仍然需要充足的内存和硬盘空间来处理解压后的数据。图像压缩的主要受益者通常是你的图像归档（存储）和网络，而不是显示工作站。

压缩的暂时状态总是让人难以理解。假如你走进你的超声检查室很有可能听到这样的抱怨：“为什么我的计算机这么慢呀？我的超声文件才只有 30 兆！”是的，这些图像文件是只有 30MB 因为他们是以 10:1 的比率压缩后的图像。当他们在计算机内存中被调取出来并进行显示时，他们会解压缩回原始的大小，即 300MB——足够把你的计算机拖慢的。如果图像压缩在存储方面的优点是显而易见的（通过 R_{comp} 这个因素来减少存储量），那么它在网络方面的压缩增益就显得更狡猾些了。请看图 24

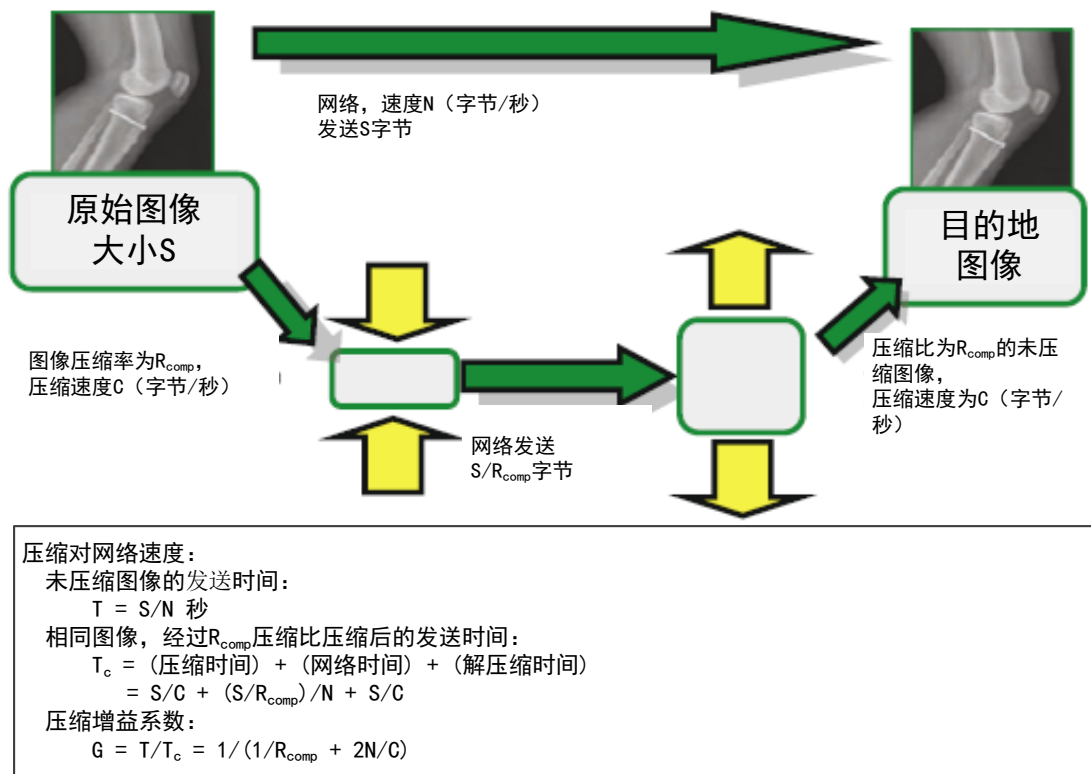


Fig. 24 在网络上使用图像压缩

发送图像时使用 R_{comp} 的比率进行压缩可以减少网络传输时间，但是将会须要额外的时间来压缩和解压缩图像。因此，如果你想要使用图像压缩来加速你的网络传输达到 K 倍，那么根据图 24 你会得到：

$$G = T/T_c = 1/(1/R_{comp} + 2N/C) > K$$

或

$$C > 2N/(1/K - 1/R_{comp})$$

如果我们考虑使用一个适中的无损压缩率 R_{comp} = 3，并且我们希望获得合理的双倍传输提速即 K=2，那么我们会得到：

$$C > 2N/(1/2 - 1/3) = 12N$$

意味着图像压缩算法必须实现 12 倍于网络速度的数据处理能力。即使你使用一个相对较慢的 10Mbps 网络，即需要 120Mbps = 15MB/s 压缩能力的压缩算法来保证你的图像传送速度能够

提高一倍。然而，当前的压缩算法中，通常的处理速度在 1-2MBs 的水平。如你所见，在一个速度适中的网络环境下，这种所发实际上降低了你的速度而不是提高了速度。另一方面，在分布式临床网络中，比如进行远程放射诊断时，整体的网络速度趋向于慢速（通常因为“[最后一公里”问题](#)），这时压缩能够明显提高图像传输速率。

再次， R_{comp} 的值总是依靠图像并且在图像压缩前是无法预计的。比如，图像有大片的单调背景（乳腺X线摄影片）通常压缩的效果比有大量本底噪声的图像（NM核医学）要好很多，仅仅是因为“单调”意味着“冗余”，并且能够被更好地压缩。目前，越来越多的人开始期待 3D 压缩技术——3D JPEG2000（适用于静止图像序列，比如 CT 断层图像），以及 MPEG4（适用于视频，比如超声电影回放）。3D 压缩发掘了图像内部冗余的额外维度，这能够使我们获得更高的 R_{comp} 值。而且，每个压缩算法有其自己的医学图像类型，不同类型的图像使用不同类型的算法将会达到最好的效果。比如，一个 8 比特有损 JPEG，绝对不适合用在 CT 或 MR 图像上¹；但是超声用起来会感觉挺好，因此该方法常用在超声设备上。

我们必须暂时避免更深入的介入到图像压缩领域中去，因为它们已经超越了本书的范围；但是如果你要使用压缩技术，那么在医疗机构中使用任何特定的图像压缩类型之前，务必咨询一下图像压缩方面的专家。

唾手可得的果实

通常压缩的实施比听起来要容易。比如，微软 Windows 当前的版本就支持内嵌的文件压缩；你总是能够右键单击一个文件夹并且将它压缩。这意味着所有存储在文件夹中的文件都将可以使用 Windows 内嵌的压缩功能进行无损压缩，而不去管处理这些文件所使用的程序。

要是这些文件夹就是你的图像存储文件夹又会怎样呢？所有存储在这些文件夹中的图像将会自动地且无损地进行压缩。我曾经试着使用这个方法来自来压缩 CT 的图像归档，而且我还获得了 1.6 的 R_{comp} 。压缩效果不那么令人印象深刻，但是，嘿，朋友们，我几乎啥都没做就减少了 1.6 倍的图像存储空间！

许多当下的 PACS 图像存储的费用是 10-20 美分/MB（注意：是 2008 年的数据），就算你只是压缩到原来的二分之一，那么你也可以省下钱买许多新的图像服务器了。

¹ 8 比特 JPEG 只能保存 8 比特/像素，而 CT 或 MR 图像最高可达 16 比特。最终你会失去色彩/灰阶的深度。一定要选用那些没有任何色彩深度限制的压缩算法。

6.3 使用数字医学图像

6.3.1 图像插值

太多法律和临床上的争论都集中在了有损图像压缩身上，图像内插（数字化放大缩小）的问题看起来完全被遗忘了。一段时间之前我不得不写一个 DICOM 图像输出程序，程序负责将 DICOM 图像输出为常见的图像格式，比如 JPEG 和 BMP。许多医生仍然喜欢这么做，当然不是为了医学目的（我们绝不鼓励这样做），而是为了其他的一些事情，比如写研究论文和保存教学文件。不久之后，我的一个同事（一位资深放射影像专家）问我为什么她的 MR 图像（导出为 JPEG 格式）变得如此之小；比她之前在 PACS 工作站上看到的图像小了很多。这到底怎么了？

我的回答是：“真没怎么”。当处理数字成像数据时，你看到的图像可以说就不是你拿到的那个。在一个典型的 1024×1024 显示器上，一个典型的 256×256 的 MR 图像看起来很小，如果在一个比较好的放射专用显示器上就看起来非常地小了。因此，任何 PACS 显示软件总是会在小图像上进行插值使他们变大，这样才能利用整个显示区域。

插值（数字放大缩小）是一个改变数字图像原始分辨率的过程，即人为地增加图像的像素数。如果你有一个 256×256 的图像，那么你就只有 $256 \times 256 = 65536$ 个像素。如果你想在 1024×1024 的显示器上全屏显示该图像，那么你就只能把它当做 1024×1024 的图像来显示，即具有 $1024 \times 1024 = 1048576$ 个像素。因此，当你放大图像到全屏模式时，多出来的那些像素（ $1048576 - 65536 = 983040$ ）到底从何而来呢？它们是通过一个插值算法（通常是双三次插值(bicubic interpolation)）得到的，做法就是在原始图像像素中间插入些新的像素，使得最终呈现的图像矩阵变得更大。因此，当你对数字图像放大时，你渐渐发现原始图像上原本没有的一些像素点通过插值而添加到了图像上。比如，从一个放大 4 倍的图像上挑出每个 $4 \times 4 = 16$ 个像素，这 16 个像素里面只有 1 个像素来自原始图像，其他 15 个实际上都是程序创建的（图 25）；这是实实在在的增加，不是吗？对图像数据进行实质上的改变比任何适度的有损压缩都要大。

对人造像素的控制力体现了为什么 PACS 开发者和用户以非常严肃的态度对待图像插值，而且在这个领域仍然在进行着大量的研究。因为插值程序是没有权利为了获得更好的图像而再次访问原始对象（病人）的，意味着它只能巧妙地“冒充”所有额外的像素以便让人感觉尽可能的自然。“冒充”的品质是一个很特别的东西，它总是能够用来评判图像显示软件的质量。为了测试一个你可能会购买的 PACS 显示工作站，你可以调取一个小图像（比如 MR 或 NM），将图像放大，看看有没有马赛克、锯齿、虚线以及其他看起来不自然的加工痕迹。如果你发现了前面提到的任何问题，那么就可以断言这个工作站软件使用了一个糟糕的插值方法，并且其与规定的标准品质相去甚远。图 26 描绘了对于一个 NM 小图像两个插值算法之间的差别：左边的插值方法极其差劲，而右边的就自然得多了。

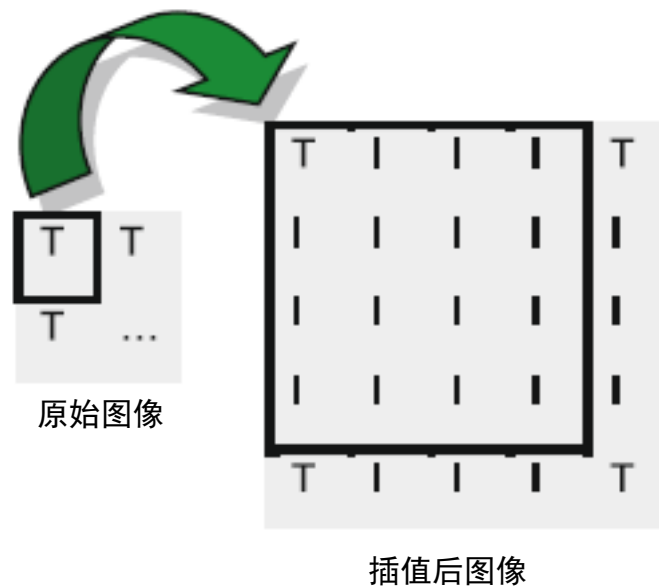


Fig. 25 一个4X 插值。“T”表示原始（真实的）像素，而“I”表示插值后的像素

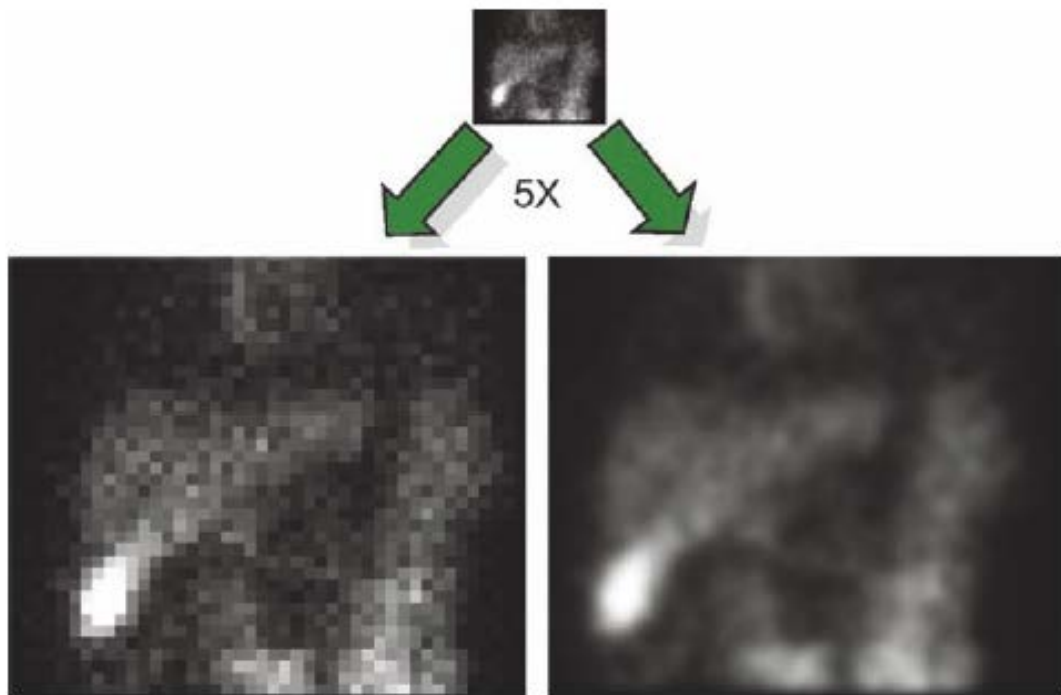


Fig. 26 上面为原始的NM（核医学）小图像。左侧劣质的最近邻居图像插值。右侧较好的双三次插值。截图是从两个不同的商业图像浏览程序上获得的。

插值还有另外一个加工痕迹是不可能避免的。当一个图像被过度放大（如图 26 中的例子）时你就会发现图像看起来有些模糊。这恰恰反映了一个事实——插值不会给图像增加信息不可能增加任何的好效果和锐利细节。也可以得到另一个结论：如果你主要使用小图像，就别花大价钱去买一个价格不菲的具有高分辨率的显示器。典型的 NM 图像最小只有 64 个像素（宽度或高度上）；MR 图像宽和高也就是 256 或 512 个像素；CT 图像 512 像素；超声图像 400 或 800 像素。当今任何普通的显示器都能够应付这些数字了。别为那些冒牌像素花

冤枉钱了。

令人感到相当意外的是，DICOM 通常对图像和显示质量非常在意，但却忽视了数字图像插值的这一部分。

6.3.2

图像重建

虽然我们之前无数次提到 DICOM 在诊断用影像处理方面是必须的，但还是让我们举个例子来看看 DICOM 是怎么搞定这些事的吧。

图像重建——平面、曲面、表面、三维——已经成为时下放射领域最热门的词。它们也是数字成像对比从前的硬拷贝胶片(那时你根本没法后处理)最独特的优势之一。如果 DICOM 数据对象中没有收集各种信息，那么重建后处理是根本不可能实现的。

考虑到一个简单的数字成像，比如 CT 扫描。除它的像素属性之外，DICOM 会记录所有有关的长度、三维坐标和方向。特别是如图 27 所示：

1. **图像像素之间的间距 D_p 保存在 DICOM(0028,0030)“像素间距 (Pixel Spacing)”属性中。**它定义了图像像素的物理大小并且保证了实际距离测量的准确性。比如，如果你知道 x 和 y 轴的像素间距为 0.4mm，那么在图像中的一条 10 像素的线就会有 4mm 的长度。同样，由于你知道图像像素中的宽和高（比如对于普通 CT 来说是 512×512），你就能够找到图像的实际尺寸了： $512 \times 0.4 \text{ mm} = 204.8 \text{ mm}$ 。
2. **“图像位置”(0020,0032)属性， I_p 。**这表示图像最左上角（第一个像素）的 x、y、z 坐标，单位为毫米。这可以让我们了解在三维空间内图像开始的位置。
3. **“图像方向”(0020,0037)属性。**它存储了图像行和列向量 v_r 和 v_c 在三维方向上的余弦。这两个向量，源自属性图像位置 (Image Position) 点 I_p 。其在三维空间完整地定义了整个图像平面。现在，如果我们有一个图像像素 P 在第 r 行、第 c 列，那么我们就能找到它的三维坐标了，方法如下：

$$P_{3D} = I_p + r \times v_r + c \times v_c$$

在 x、y、z 三个坐标上分别计算。

4. **“层间距”(0018,0088)属性， D_s ，**记录了连续各层图像之间的距离，单位为毫米。它提供了和(0028,0030)“像素间距 (Pixel Spacing)”相同的作用，但是只在 z 方向上。比如，如果你考虑在第一和第二个 CT 层面上那个相同的像素(r,c)位置，那么两层面之间的距离将等于(0018,0088)属性的值。

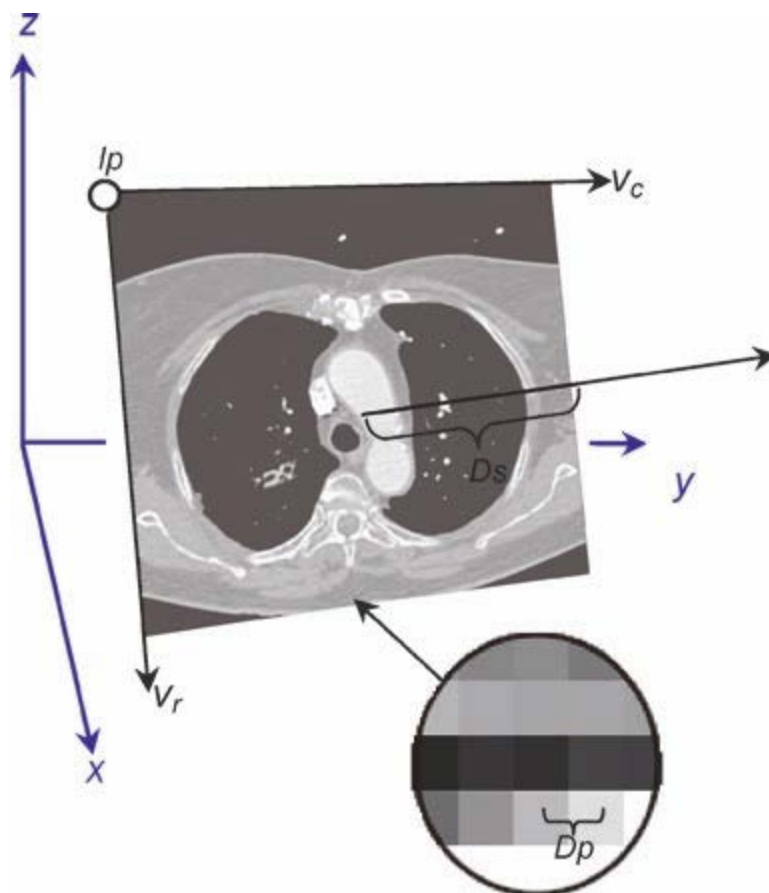


Fig. 27 在 DICOM 中的三维图像坐标

5. 图像时间、层厚、位置和一些其他的 DICOM 属性记录了其他重要的坐标。比如，图像获取时间就是灌注分析必不可少的信息，当我们需要处理来自有时序的图像序列信息时，就需要这些了。

现在想象一下你有一个原始的 CT 序列，而你想要创建一个倾斜的图像，即在相同的序列卷上的不同成像平面上进行分层切割。这真要谢谢 DICOM 属性，是它使你能够实现这个功能，并且仅仅通过一些简单步骤：

1. 为你想得到的斜剖面定义新的图像平面属性：图像位置点 I_{p0} 以及两个图像取向向量， v_{r0} 和 v_{c0} 。在斜面上像素之间的间距更像原始图像，仍然保持着相同的纵横比。基于同样的假设，图像像素大小也会保持原样。
2. 在你新的倾斜图像平面上对每个图像像素的数值进行计算，如下：
 - i. 让每个像素与倾斜平面协调一致， (r_o, c_o) 。
 - ii. 用之前的算式，将它转化到整体的三维空间：

$$P_{3Do} = I_{p0} + r_o \times v_{r0} + c_o \times v_{c0}$$
 这将告诉你在三维空间，新的 P_{3Do} 像素与原始图像序列像素之间的位置关系。
 - iii. 使用原始图像序列，用插值算法找到 P_{3Do} 点的值。
3. 对倾斜图像像素指派数值，并且处理下一个像素。

上面提到的第三步插值，可以使用我们较早前学到的插值方法。当我们计算当前倾斜图像像素 P_{3Do} 三维坐标时，像素将会落在原始序列像素中的某处。最简单的最近相邻插值法会在原始的序列像素上找到最接近 P_{3Do} 像素的位置，并且将它的亮度赋给倾斜面上的那个像素点。更好的线性插值法会利用八个最接近原始像素的点在 P_{3Do} 上进行亮度的线性插值。

这种只进行较小修改的做法用作斜面（多平面）、容积、最大密度投影、和最小密度投影等表现方法（图 28）上。无论什么情况下，最重要的步骤就是在给定的三维像素点 P_{3D} 上计算密度值，这只需要一些DICOM属性就能够轻松实现。你可能会想，如果对大量像素进行重建，那么三维容积处理就需要耗费相当大的计算量和计算时间了，这也说明了为什么重建过程总是在一些特殊的硬件加速设备上进行的，并且会使用高档的显卡。这些加速用的数字线路是用来处理繁忙的插值运算的。这也就是你装配三维演示工作站或服务器时的主要花费。非常有趣的是，这个用在放射领域的演示方法，包括硬件，是来自于电脑游戏产业的。几年前在电脑游戏产业，这种演示方法就已经很完美了。但是很显然，要是没有DICOM属性，这种演示方法还是啥也做不了。从简单的斜面重建到最错综复杂的“虚拟导航”软件，DICOM提供所有必要的信息来实现最复杂的图像重建。

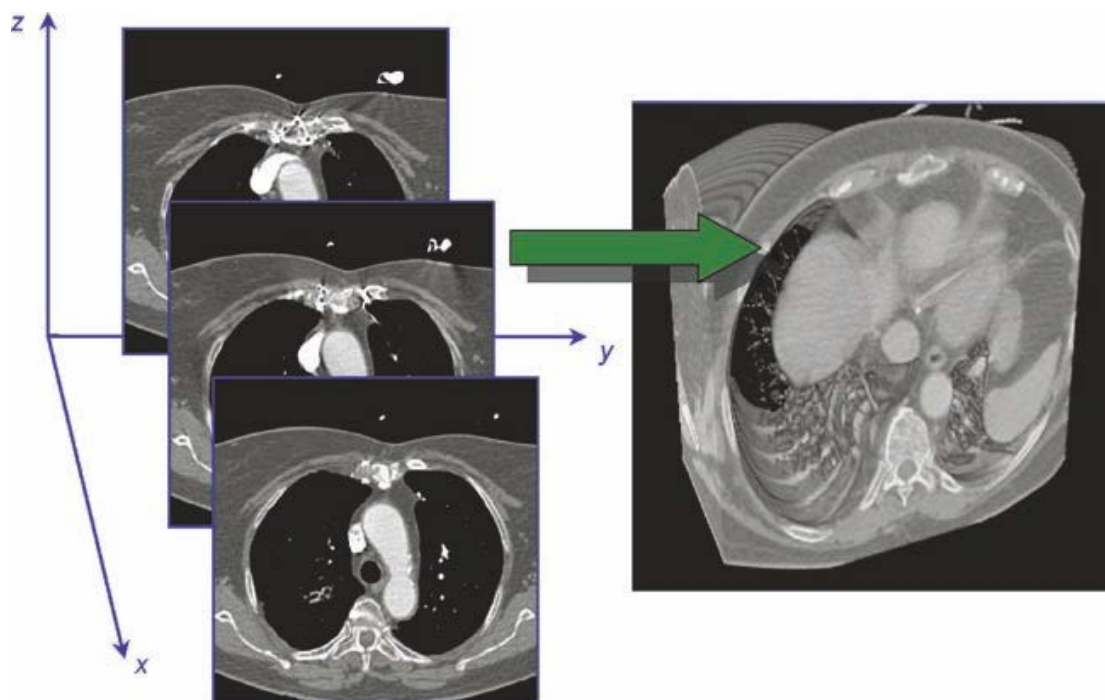


Fig. 28 使用 DICOM 用原始图像序列进行三维重建

6.3.3

灰阶深度

“我的显示器需要支持多少灰阶？”，除了现在我们在彩色/灰度范畴讨论分辨率以外，这个问题很自然地补充了图像空间分比率和插值方面的问题。¹ 这个论题的许多方面经常混淆在一起，使其变得难以理解，因此值得花一些时间来让这些问题看起来更明确。

真正重要的是原始图像色彩深度。如我们在 6.1 中所学到的，一个CT图像存储用比特 B_s 等于 10，那么图像就有 $2^{B_s}=1024$ 个灰色明暗度。只要图像以原始的DICOM格式保存，它就会一直保有 1024 个明暗度，并且因此它需要用适合的软件和显示器来显示和浏览。另一方面，图像被输出成非DICOM格式之时，比如输出为BMP或JPEG，就只能支持 256 个灰阶了；此外你有可能使用了错误的压缩技术，因此你绝没有可能一直保持着原始的 1024 个灰度级别不变。在广大的世界上，没有软件或者硬件能够用原始的DICOM品质来显示一个已经降质的图像：不知道自己丢了什么（同样的情况也会出现在过分使用有损压缩上）。

这就是为什么任何远程放射诊断项目（通常在简单图像格式或输出格式方面上了别人当）都应该保持原始图像格式，而不要试图通过高价的显示工作站来恢复已经永久失去的图像品质。这同样也是为什么许多官方放射项目方案（包括远程放射诊断）坚持保留原始图像品质

¹ 如我们先前所认可的，图像色彩深度我们认为就是图像明暗度，他们是灰度还是色彩。

而不只是委托给那些图像显示选项来处理的真正原因²（北美放射学院 2002）

所有比这些要点还要超前或更高的要求就实在有点主观了，那只不过是从小个人舒适角度来考虑的了。

首先，对于调整窗宽/窗位功能，几乎任何 PACS 或 DICOM 软件都会提供，你能够动态调节当前可见的灰度范围。比如，如果你正在用一个现成的普通显示设备（只具有 256 个灰度呈现能力）来显示具有 1024 个灰阶明暗度的 CT 图像时，你可以使用你的“骨窗”、“肝脏窗”、“脑窗”和其他窗来浏览它。每个这样的窗会简单地从图像上 1024 个明暗度中提取 [C0, C1] 范围，并将它映射你显示器上可用的 256 个明暗度上。换句话说，这很像改变放大率和移动图像，只不过你“改变放大率”（窗宽）和“进行移动”（窗位）的是灰度范围。如果原始图像品质得到保留，那么不管是在能够显示 5 兆像素的放射专用怪物上还是你的手机上进行显示，你都将能够在显示器上看到所有图像明暗度。只不过，在能够显示 1024 个明暗度的放射专用显示器上你可以一次性看到所有明暗度，而在 256 个明暗度的普通显示器上你只能在各个明暗度之间来回切换。

之前所述内容中，还剩下一个问题，即特定显示器的属性以及它们对放射学产生的影响。尤其是最重要的属性，即显示器亮度：显示器的灰阶可以有多明亮：自然地，显示器越亮，你就越容易察觉不同明暗度之间的区别。这里需要提及一个“恰好可见的不同（just noticeable difference (JND)）”的观念——在给定的环境下，平常的人类观察者所能察觉的灰阶中最小的不同。

理论上，JND 由显示器亮度决定，你能在 DICOM PS3.14 中找到一个令人印象深刻的以实验为依据的原则，即图 29 中的曲线。比如，如果你选择了一个非常好的具有 500cd/m² 亮度的显示器，那么你的眼睛也许可以看到约 700 个不同的灰色明暗度，前提是显示器具有同时提供至少 700 个明暗度的能力（见 DICOM PS3.14 附件 B，来获取更完整的 JND 值表格）。除了这些平常的和理论上的用途外，JND 函数还用在一个非常实用的目的：校准放射专用显示器。校准的目的是在不同级别上调节显示器的亮度，在这样的方式下，JND 将会被均匀一致地分配。你在分辨各个较暗明暗度上的能力和你分辨各个较亮明暗度上的能力是一致的。这就很容易得出一个道理，不要因为过于强调某一个强度而损失了其他的强度；比如 CT，相较于其他密度范围，在某个密度范围中，如果不用亨氏单位（Hounsfield unit）显示得更明显。

但是即使在实际应用中，完全基于 JND 的 DICOM 显示器校准也不能保证完美的图像显示。人类的视觉系统非常复杂。比如，你分辨特定明暗度的能力会受到图像噪声、分布、形状以及你正在观察的对象位置的影响。你的感觉也会受到环境光线的影响，当然还有一大堆别的因素，可能包括你椅子的高度甚至你的饮食质量。在图像读取方面纯粹的“舒适”，意味着应更重视高质量图像的呈现方式，而不是最前沿的高科技显示方案。比如，一次性看到最大灰阶数量能够让图像浏览更方便（Kimpe 和 Tuytschaever 2006），但是它同样会转化（依据图 29 中我们的 JND 曲线）为显示器亮度的提升，依据报告，如果高于 600cd/m²（相当于约 730 JND，这完全低于我们显示 1024 个 CT 明暗度的要求）会造成人们注意力分散。因此，当前放射专用显示器通常提供 300–400 cd/m² 这种比较舒适的亮度，但是根本无法接近我们一次能显示 1024 个 CT 灰阶明暗度的要求。

我用一个全新的、现成的显示器来录入这本书，它有 500 cd/m² 的亮度，1000:1 的对比度，以及 1200×1900 的分辨率，这几乎与当前 PACS 显示器一样的好，甚至更好。真正的区别是 PACS 显示器比我的显示器贵 10 倍（Hirschorn 和 Dreyer 2007）。

² 唯一的例外是在乳腺摄影显示方面：它们需要 5 兆像素分辨率（通常为 2048×2560 像素），并且还要达到高达 750cd/m² 的亮度以及 4096 个灰阶。

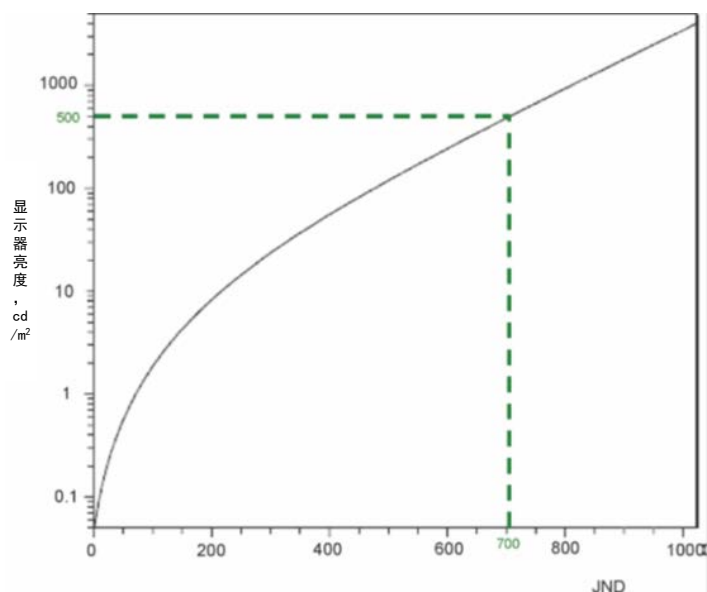


Fig. 29 “恰好可见的不同”（Just Noticeable Difference (JND)）曲线

你能说出不同点吗？

关于显示器使用方面的“理想对决实用”，我在几年前遇到过一个最有趣的例子。那是在一所非常有名的国际医院中使用的一套非常有名的PACS厂商的产品。由于PACS安装了高分辨率、双屏放射专用显示设备（来自一个非常有名的显示设备厂商），每个人都对其具有 10 比特/像素深信不疑，因此他应该具有 $2^{10}=1024$ 个灰阶。此外，PACS公司确认他们的PACS软件也具有 10 比特的图像显示能力。

两年之后，一个完全偶然的的机会，我们发现这个显示器只有 8 比特/像素；换句话说，只提供 $2^8=256$ 个灰阶，任何普通显示器都能够达到这个水平！这表明虽然显示器和PACS软件是 10 比特，但是PACS工作站上的显卡只支持 8 比特，因此会把任何东西转化为只有 8 比特的显示模式！因此，医院的放射影像专家实际上一直在用非常一般的显示器进行顶尖的诊断工作，他们一直这样用是因为他们相信这个显示器会比其他的更好。一个多么有趣的例子啊，同时体现了显示的主观性以及PACS厂商的不负责任。

6.3.4

波形

波形不是图像，但是在很多方面很相近。DICOM 是这么解释的：

“波形信息实体（IE）表示一个多通道、基于时间的数字化波形。波形是由在连续的时间间隔采样得到的一些物理要素（比如，电压、压力、气体浓度、或声音）的测量值组成的。这些可测量的要素可能来自以下源头：

1. 病人的解剖信息；
2. 治疗设备（比如，心脏起搏信号或一个射频消融信号）；
3. 同步诊断设备（比如，不同设备间的时钟信号或定时信号）；
4. 医生的声音（比如，口授报告）。 ”

数字声音更是最常见的一种波形，甚至在我们进入数字医学行业之前可能就已经知道了这一点。ECG 是另一个常见的例子。任何一个例子中，波形获取过程都是时间的演化过程，并且能够以许多信号通道进行展现（比如，立体声）。在这方面，波形很像图像。但是图像有两个空间坐标（x 和 y）作为独有的特征，波形则具有在时间 t 上变化的一维振幅。

所以，就像图像一样，模拟波形必须被转化为数字的。这意味着一个模拟信号被打散为记录着他们振幅值的离散抽样，从而形成了一个数字化的波形（图 30）。一旦被数字化，这个序列就变成了一个抽样（振幅值）的数字序列，它能够存储在 DICOM 中，就像存储图像像素一样。比特分派和存储的术语也适用于波形的表达。

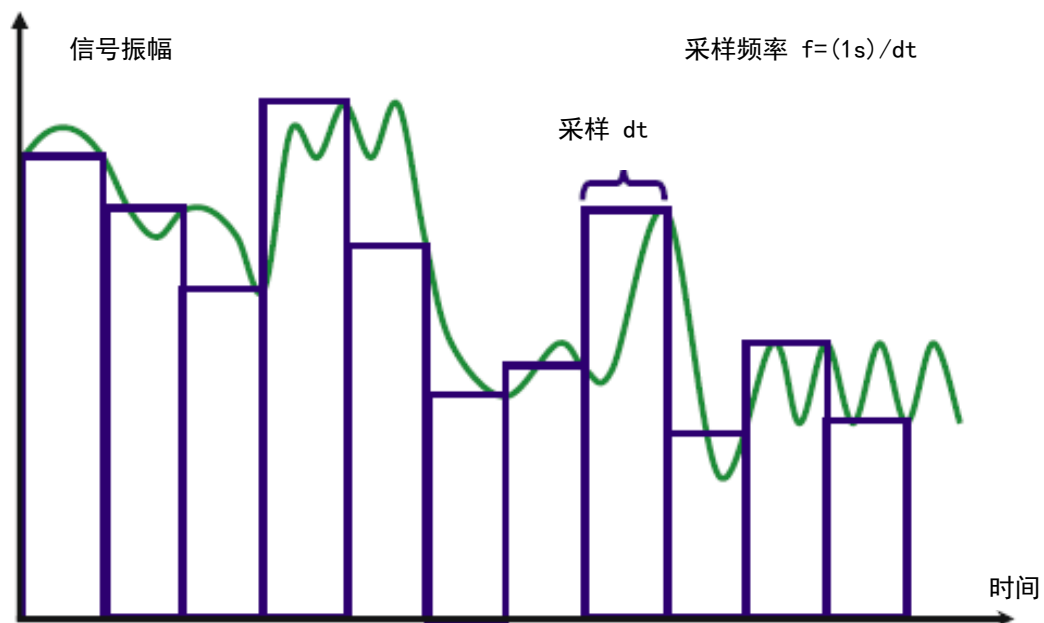


Fig. 30 将连续的（模拟）信号数字化为离散的（数字）值的一个序列。

实际上，DICOM 波形很少用来和数字图像进行比较；然而，无论我们想要表示什么，波形表示法都是一个存储一维的、多通道序列的好方法。由于 DICOM 继续支持数字多媒体，因此波形的作用只会继续增长。