论 文

ISAF 重构算法密度函数快速计算模型

王功明^{①②}, 张法^{①*}, 樊莉亚^③, 孙飞^④, 刘志勇^{①*}^① 中国科学院计算技术研究所, 北京 100190^② 中国科学院研究生院, 北京 100190^③ IBM 中国研究院, 北京 100094^④ 中国科学院生物物理研究所, 北京 100101

* 通信作者. E-mail: zhangfa@ict.ac.cn, zyliu@ict.ac.cn

收稿日期: 2012-01-06; 接受日期: 2012-05-27

中国科学院知识创新工程重大项目 (批准号: KGCX1-YW-13) 和国家自然科学基金项目 (批准号: 61232001, 61070129) 资助

摘要 球坐标系下的 ISAF 算法是一种新的 20 面体分子三维重构方法, 该方法精度优于传统柱坐标系下的 Fourier-Bessel 算法, 但其执行速度远低于 Fourier-Bessel 算法, 严重制约了 ISAF 算法的实际应用. 分析发现, 在 ISAF 算法中密度函数计算是影响重构速度的主要瓶颈之一. 针对上述问题, 文中提出一种密度函数快速计算模型, 该模型包括三个组成部分: 球坐标系网格点密度函数快速计算方法、“球坐标系—直角坐标系”网格点密度函数转换方法、基于两阶段映射法的快速对称映射方法. 该模型可以将密度函数计算阶段的时间复杂度由 $O[(L_M)^8]$ 降低到 $O[(L_M)^7]$. 采用 Psv-F 病毒数据进行实验, 结果表明, 在保证精度的前提下, 该模型可以将密度函数的计算速度提高 2 个数量级, 将三维重构整体速度提高 30 倍左右, 并且随着数据规模的增大、重构精度的提高, 该模型带来的加速比将进一步增大.

关键词 ISAF 三维重构 密度函数 球坐标系 四元数插值

1 引言

20 世纪 60 年代末, Crowther 和 Klug 等人提出了基于电镜图片重现蛋白质分子三维结构的技术^[1], 是结构生物学上一个划时代里程碑. 经过多年的发展, 该技术已经衍化为如下三种方法: 电子晶体学^[1]、电子断层技术^[2]和单颗粒技术^[3]. 其中, 单颗粒技术由于具有不需要结晶、对病毒大小没有限制等优点^[4,5], 已成为研究生物大分子结构的强有力手段.

目前常用的单颗粒三维重构算法是 Crowther 在 20 世纪 70 年代提出的 Fourier-Bessel 算法^[6], 该算法执行速度快, 但精度不能满足高分辨重构要求. 为此, Crowther 等人提出在球坐标系下采用球谐函数进行三维重构的思想^[7], 以提高重构精度. 2003 年, Navaza 整合了球谐函数, 提出了 20 面体对称匹配函数 (icosahedral symmetry-adapted functions, ISAF) 方法, 有效提高了重构精度, 重现了更多的分子细节特征^[8]. 2007 年, Liu 进一步完善了 ISAF 方法, 重构精度比 Fourier-Bessel 算法提高 0.5 Å 左右^[9].

引用格式: 王功明, 张法, 樊莉亚, 等. ISAF 重构算法密度函数快速计算模型. 中国科学: 信息科学, 2013, 43: 584–598

ISAF 重构算法的速度很慢. 主要原因有: 基函数构造复杂、三维空间变换复杂、密度函数计算复杂等. 其中, 计算密度函数涉及数以百万计、乃至千万计点的密度函数值计算, 而每点的计算又涉及球贝塞尔函数、ISAF 值 $S_{l,u}(\theta, \varphi)$ 等复杂操作, 该阶段耗时占重构时间的 90% 以上, 是该类算法的计算瓶颈.

经过分析密度函数的计算过程, 我们发现若干特性, 有利于提高密度函数计算速度. 首先, 沿径向方向, 各个球壳上对应采样点的角度值 (θ, φ) 相同, 所以这些点的 $S_{l,u}(\theta, \varphi)$ 只需计算一次; 此外, 根据 l 的奇偶性不同, $S_{l,u}(\theta, \varphi)$ 关于 y 轴有奇偶对称性, 所以只需要计算一半的 $S_{l,u}(\theta, \varphi)$; 再之, 密度函数计算公式中的积分式 $\int_0^\infty f_{l,u}(R) \cdot j_l(2\pi Rr) \cdot R^2 dR$ 与角度无关, 可预先生成, 由此提出了球坐标系网格点密度函数快速计算方法, 能快速得到球坐标系下网格点密度函数值. 其次, 基于球壳上四元数插值方法和径向线性插值方法, 我们提出了“球坐标系一直角坐标系”网格点密度函数转换方法, 能快速、等效地将球坐标系网格点的密度函数值转换为直角坐标系网格点的密度函数值. 再次, 通过分析空间网格点在象限中的对应关系, 提出了基于两阶段映射法的快速对称映射方法, 将原来需要计算 8 个象限网格点的对称映射操作缩减为 2 个象限网格点的对称映射操作, 使计算量减少了 4 倍. 最后, 综合上述三种方法, 我们提出了密度函数快速计算模型, 使得该阶段的时间复杂度降低了一个幂次, 有效提高了三维重构的计算速度.

为了验证该模型的性能, 我们采用葡枝青霉病毒 F 的模拟数据进行重构实验, 并且与原始 ISAF 算法、recISAFs 算法^[9]、传统 Fourier-Bessel 算法进行对比分析. 结果表明, 在保证精度的前提下, 我们的计算模型能有效提高三维重构的计算速度, 与原始 ISAF 算法相比, 密度函数计算部分的速度可提高 2 个数量级, 整体重构速度可提高 30 倍, 随着数据规模的增大和重构精度的提高, 该加速比将进一步增大.

2 ISAF 重构算法简介及密度函数复杂性分析

2.1 ISAF 重构算法简介

ISAF 三维重构算法主要包括四个步骤^[10], 分别是: 分子图像映射、构造并求解拟和方程组、计算密度函数、写入 MRC 文件. 这四个步骤的简介如下:

(1) 分子图像映射.

该操作主要是把分子图像中的结构信息 (空间位置, 结构因子值) 提取出来, 并映射到非对称三角形 (ISAF 三维重构的基本对称单元, 即 $\{(r, \theta, \varphi) | r \in [0, r_m], \theta \in [69.09, 90.0], \varphi \in [-31.72, 31.72]\}$), 其中, (r, θ, φ) 为球坐标, r_m 为分子半径, 如图 1 阴影所示^[11].

(2) 构造拟合方程组并求解.

该操作主要是根据非对称三角形中的结构信息构造拟和方程组, 通过求解该方程组得到拟和系数 (将空间每点结构因子值都表示为基函数线性组合时所需要的组合系数).

(3) 计算密度函数.

该操作主要是计算出非对称三角形中每个网格点的密度函数值 $\rho(x, y, z)$, 然后根据 20 面体的 60 对称性, 计算出非对称三角形外其他区域网格点的密度函数值 (详细分析见 2.2 小节). 该操作是 ISAF 重构算法的核心, 且其运行时间占据整个 ISAF 重构算法运行时间的 90% 以上, 是 ISAF 算法的瓶颈.

(4) 写入 MRC 文件.

该操作主要是把计算出来的空间网格点密度函数值存入 MRC 文件.

图 1 二次轴坐标系下的非对称三角形

Figure 1 The asymmetric unit in digonal axis coordinate system

2.2 密度函数时间复杂性分析

在 ISAF 重构算法中, 一般采用 (1) 式计算点 (r, θ, φ) 处的密度函数值 $\rho(r, \theta, \varphi)$.

$$\begin{aligned} \rho(r, \theta, \varphi) &= \int F(R, \Theta, \Phi) \exp(-2\pi i \mathbf{R} \cdot \mathbf{r}) R^2 \sin \Theta dR d\Theta d\Phi \\ &= 4\pi \left[\sum_{l=0 \bmod 2}^{L_M} (-i)^l \sum_{u=1}^{u(l)} \left(\int_0^\infty f_{l,u}(R) j_l(2\pi Rr) R^2 dR \right) S_{l,u}(\theta, \varphi) \right. \\ &\quad \left. + \sum_{l=1 \bmod 2}^{L_M} (-i)^l \sum_{u=1}^{u(l)} \left(\int_0^\infty f_{l,u}(R) j_l(2\pi Rr) R^2 dR \right) S_{l,u}(\theta, \varphi) \right], \end{aligned} \quad (1)$$

其中, $S_{l,u}(\theta, \varphi)$ 是 ISAF 基函数, l 是 $S_{l,u}(\theta, \varphi)$ 的阶数, $0 \leq l \leq L_M$, u 是 $S_{l,u}(\theta, \varphi)$ 的重数, $1 \leq u \leq u(l)$, $f_{l,u}(R)$ 是拟合系数, $j_l(2\pi Rr)$ 是球贝赛尔函数, R 是倒易空间的傅里叶半径. 可以看出, 计算每一点的密度函数值, 都需要二重求和、积分、计算 $j_l(2\pi Rr)$ 和 $S_{l,u}(\theta, \varphi)$ 等复杂操作, 运算量很大. 其中, 球贝赛尔函数和 ISAF 基函数是密度函数计算的两个基本元素, 其具体计算方法如下:

(1) 球贝赛尔函数 $j_l(2\pi Rr)$

球贝赛尔函数 $j_l(z)$ 定义如 (2) 式所示^[12]:

$$j_l(z) = \frac{z^l}{1 \cdot 3 \cdot 5 \cdots (2l+1)} \left\{ 1 - \frac{\frac{1}{2}z^2}{1!(2l+3)} + \frac{(\frac{1}{2}z^2)^2}{2!(2l+3)(2l+5)} - \cdots \right\} = (-1)^l z^l \left(\frac{d}{z dz} \right)^l \frac{\sin z}{z}. \quad (2)$$

一般采用 Miller 方法递推计算 $j_l(z)$, 时间复杂度是 $O(l^2)$.

(2) ISAF 基函数 $S_{l,u}(\theta, \varphi)$

$S_{l,u}(\theta, \varphi)$ 是球谐函数的线性组合^[13], 在 (θ, φ) 处第 l 阶、第 u 重 $S_{l,u}(\theta, \varphi)$ 如 (3) 式所示.

$$S_{l,u}(\theta, \varphi) = \sum_{m=-l}^l A_{l,m}^u \cdot Y_{l,m}(\theta, \varphi), \quad (3)$$

其中, m 的取值范围是 $m \in [-l, l]$, $A_{l,m}^u$ 是组合系数, 一般采用裂项法^[14]求解 $A_{l,m}^u$. $Y_{l,m}(\theta, \varphi)$ 是球谐函数, 定义如 (4) 式所示^[15]:

$$Y_{l,m}(\theta, \varphi) = \sqrt{\frac{(2l+1)}{4\pi} \cdot \frac{(l-|m|)!}{(l+|m|)!}} P_l^m(\cos \theta) \cdot e^{im\varphi}, \quad (4)$$

其中, $P_l^m(\cos \theta)$ 是缔合勒让德多项式, 一般采用动态规划方法计算, 时间复杂度是 $O(l^2)$. 结合 (3)(4) 式可知, $S_{l,u}(\theta, \varphi)$ 的时间复杂度是 $O(l^3)$.

在 (1) 式中, R 表示倒易空间的傅里叶半径, $R \in [0, \infty)$, 存在 R_M , 当 $R > R_M$ 时, $j_l(2\pi Rr) \rightarrow 0$, (1) 式中的无穷积分可以转换为定积分, 通常采用累加方式计算该定积分 [7], 由此可将 (1) 式表示如下:

$$\rho(r, \theta, \varphi) = 4\pi \left[\sum_{l=0 \bmod 2}^{L_M} (-i)^l \sum_{u=1}^{u(l)} \left(\sum_{R=0}^{R_M} f_{l,u}(R) j_l(2\pi Rr) R^2 \right) S_{l,u}(\theta, \varphi) + \sum_{l=1 \bmod 2}^{L_M} (-i)^l \sum_{u=1}^{u(l)} \left(\sum_{R=0}^{R_M} f_{l,u}(R) j_l(2\pi Rr) R^2 \right) S_{l,u}(\theta, \varphi) \right]. \quad (5)$$

根据 $j_l(2\pi Rr)$ 和 $S_{l,u}(\theta, \varphi)$ 的时间复杂度, 可以看出, 计算任一点的 $\rho(r, \theta, \varphi)$, 需要的基本操作个数为

$$u(0)(R_M \cdot 1^2 + 1^3) + u(1)(R_M \cdot 1^2 + 1^3) + u(2)(R_M \cdot 2^2 + 2^3) + \cdots + u(L_M)(R_M \cdot (L_M)^2 + (L_M)^3),$$

其上界为

$$u(L_M)[R_M \cdot (L_M)^2 + (L_M)^3] + \cdots + u(L_M)[R_M \cdot (L_M)^2 + (L_M)^3] \\ = (L_M + 1) \cdot u(L_M)[R_M \cdot (L_M)^2 + (L_M)^3].$$

由于 $u(L_M) = \left\lfloor \frac{L_M}{30} \right\rfloor + u(L_M \bmod 30) = \left\lfloor \frac{L_M}{30} \right\rfloor + \{0, 1\}$, $R_M = L_M \cdot \frac{D}{2\pi r_m}$, D 是分子图片的边长 [9], 所以 $\rho(r, \theta, \varphi)$ 的时间复杂度是

$$O\{(L_M + 1) \cdot u(L_M) [R_M \cdot (L_M)^2 + (L_M)^3]\} \\ = O\left\{L_M \cdot \left\{\left\lfloor \frac{L_M}{30} \right\rfloor + \{0, 1\}\right\} \cdot \left[L_M \cdot \frac{D}{2\pi r_m} \cdot (L_M)^2 + (L_M)^3\right]\right\} \\ = O[L_M \cdot L_M \cdot (L_M)^3] = O[(L_M)^5].$$

非对称三角形中的网格点个数约为 $N_1 = \frac{1}{60} \cdot \frac{4}{3} \pi \cdot r_m^3 = \frac{\pi}{45} \cdot r_m^3$, 所以重构过程中密度函数时间复杂度为 $O[(L_M)^5 \cdot \frac{\pi}{45} \cdot r_m^3] = O[(L_M)^5 \cdot r_m^3]$, 又因为 $r_m = L_M \cdot \frac{D}{2\pi R_M}$, $D \geq 2R_M$, 所以 $O[(L_M)^5 \cdot (r_m)^3] = O[(L_M)^5 \cdot (L_M \cdot \frac{D}{2\pi R_M})^3] = O[(L_M)^8]$. 由此可知, 密度函数的时间复杂度相当高, 严重影响了 ISAF 算法的整体运行速度. 所以, 降低密度函数时间复杂度, 是提高 ISAF 算法运行速度的关键所在.

3 密度函数快速计算模型

针对上述问题, 我们提出密度函数快速计算模型, 将该阶段的时间复杂度由 $O[(L_M)^8]$ 降到 $O[(L_M)^7]$. 该模型包含三个组成部分: (1) 球坐标系网格点密度函数快速计算方法, (2) “球坐标系—直角坐标系” 网格点密度函数转换方法, (3) 基于两阶段映射法的快速对称映射方法.

3.1 球坐标系网格点密度函数快速计算方法

该方法涉及下述三个特性:

图 2 非对称三角形的网格划分

Figure 2 The mesh segmentation in asymmetric unit. (a) The mesh segmentation in a given shell; (b) the mesh segmentation in the different shells

特性 1: 沿径向方向, 不同球壳上对应网格点的极角 (θ, φ) 相同.

在非对称三角形区域, 按照固定的角度间隔 $\Delta\theta$ 、 $\Delta\varphi$ 逐个划分各个球壳, 可以得到不同球壳上的网格点, 图 2(a) 表示一个球壳上的网格划分情况. 网格划分完成之后, 从原点出发, 作一条经过图 2(a) 球壳上网格点 (θ, φ) 的径向射线, 由径向角度相等的性质可知, 该射线与其他球壳的交点也一定是划分好的网格点, 它们的极角也一定是 (θ, φ) , 如图 2(b) 所示. 所以这些网格点的 $S_{l,u}(\theta, \varphi)$ 也相同, 因此, 只要计算出一个球壳上网格点的 $S_{l,u}(\theta, \varphi)$ 即可, 其他球壳上的 $S_{l,u}(\theta, \varphi)$ 与此完全相同.

特性 2: 根据 l 的奇偶性不同, $S_{l,u}(\theta, \varphi)$ 关于 y 轴有奇偶对称性, 如 (6) 式所示.

$$S_{l,u}(\theta, \varphi) = \begin{cases} S_{l,u}(\theta, -\varphi), & l \text{ 为偶数,} \\ -S_{l,u}(\theta, -\varphi), & l \text{ 为奇数.} \end{cases} \quad (6)$$

对 $S_{l,u}(\theta, \varphi)$ 而言, 系数 $A_{l,m}^u$ 和 $Y_{l,m}(\theta, \varphi)$ 具有如下性质:

$$A_{l,-m}^u = (-1)^{m-l} A_{l,m}^u, \quad (7)$$

$$Y_{l,-m}(\theta, \varphi) = (-1)^{-m} Y_{l,m}^*(\theta, \varphi), \quad (8)$$

由 (7) 式可知, 当 l 为奇数时, $A_{l,0}^u = 0$, 由此可知

$$\begin{aligned} S_{l,u}(\theta, \varphi) &= A_{l,0}^u Y_{l,0}(\theta, \varphi) + \sum_{m=5}^l [A_{l,m}^u Y_{l,m}(\theta, \varphi) + A_{l,-m}^u Y_{l,-m}(\theta, \varphi)] \\ &= \begin{cases} A_{l,0}^u \sqrt{\frac{2l+1}{4\pi}} P_l^0(\cos\theta) + \sum_{m=5}^l 2A_{l,m}^u P_l^m(\cos\theta) \cos m\varphi, & l \text{ 为偶数,} \\ \sum_{m=5}^l 2A_{l,m}^u P_l^m(\cos\theta) \sin m\varphi, & l \text{ 为奇数.} \end{cases} \end{aligned}$$

由上式可知 (6) 式成立.

所以, 计算 $S_{l,u}(\theta, \varphi)$ 时只需要计算一半 ($\varphi \geq 0$) 即可, 另一半可根据 (6) 式生成.

特性 3: (1) 式中积分式 $\int_0^\infty f_{l,u}(R) j_l(2\pi Rr) R^2 dR$ 与角度 (θ, φ) 无关, 可预先生成.

综合上述三个特性, 可以得到球坐标系网格点密度函数快速计算方法, 流程如下:

(1) 计算积分式 $\int_0^\infty f_{l,u}(R) j_l(2\pi Rr) R^2 dR$, 其中 $l \in [0, L_M]$ 、 $u \in [1, u(l)]$ 、 $r \in [1, r_m]$, 并按照先 r 后 (l, u) 的顺序储存在二维数组中;

(2) 在非对称三角形区域中, 令 $\Delta\theta = \Delta\varphi = \frac{1}{r_m}$, 以 $\Delta\theta$ 、 $\Delta\varphi$ 为角度间隔逐个划分各个球壳, 得到不同球壳上的网格点;

(3) 根据球壳上所有网格点处的极角 (θ, φ) , 生成对应的所有阶、所有重 $S_{l,u}(\theta, \varphi)$;

图 3 “球坐标系 — 直角坐标系” 网格点密度函数转换

Figure 3 The conversion of density function in mesh points from spherical coordinate to Cartesian coordinate

图 4 球面上四元数插值

Figure 4 The quaternion interpolation in a certain shell

(4) 把 (1) 中的 $\int_0^\infty f_{l,u}(R) j_l(2\pi Rr) R^2 dR$, (2) 中的 $S_{l,u}(\theta, \varphi)$ 带入 (1) 式, 求出所有球壳上网格点处的 $\rho(r, \theta, \varphi)$;

上述方法利用不同球壳上径向网格点极角相等、 $S_{l,u}(\theta, \varphi)$ 关于 Y 轴奇偶对称、积分式 $\int_0^\infty f_{l,u}(R) j_l(2\pi Rr) R^2 dR$ 与角度 (θ, φ) 无关等特性, 可以快速计算出球坐标系下所有网格点处的密度函数值。

3.2 “球坐标系 — 直角坐标系” 网格点密度函数转换方法

MRC 文件保存每个整数直角网格点 (X, Y, Z) 处的密度函数值. 然而, 在 3.1 小节中, 得到的只是球坐标系网格点的密度函数值, 所以还需要转换为直角坐标系网格点的密度函数值。

该问题可抽象为:

如图 3 所示, A_1, B_1, C_1, D_1 是某一球壳上的四个网格点, A_2, B_2, C_2, D_2 是相邻球壳上的四个对应网格点, E 是直角坐标系中的某一网格点, 如何由已知点 $A_1, B_1, C_1, D_1, A_2, B_2, C_2, D_2$ 的密度函数值, 求出未知点 E 的密度函数值?

四元数插值是适用于球面的插值方法, 比常见的线性插值方法精度更高. 如图 4 所示, A, B, C, D 是球面网格上四个点, 构成球面四边形 $ABCD$, 其边长分别是 $\Delta\theta, \Delta\varphi$, A, B, C, D 四点的密度函数值分别为 $\rho(r, \theta, \varphi), \rho(r, \theta + \Delta\theta, \varphi), \rho(r, \theta + \Delta\theta, \varphi + \Delta\varphi), \rho(r, \theta, \varphi + \Delta\varphi)$, 则球面四边形 $ABCD$ 内部点 $O'(\theta', \varphi')$ 处的密度函数值 $\rho(r, \theta', \varphi')$ 如 (9) 式所示, 其中, $t_\theta = \frac{\theta' - \theta}{\Delta\theta}, t_\varphi = \frac{\varphi' - \varphi}{\Delta\varphi}, \omega = \frac{\pi}{180}$.

$$\begin{aligned} \rho(r, \theta', \varphi') = & \frac{\sin(1 - t_\theta)\omega}{\sin\omega} \cdot \frac{\sin(1 - t_\varphi)\omega}{\sin\omega} \rho(r, \theta, \varphi) + \frac{\sin t_\theta \omega}{\sin\omega} \cdot \frac{\sin(1 - t_\varphi)\omega}{\sin\omega} \rho(r, \theta + \Delta\theta, \varphi) \\ & + \frac{\sin(1 - t_\theta)\omega}{\sin\omega} \cdot \frac{\sin t_\varphi \omega}{\sin\omega} \rho(r, \theta, \varphi + \Delta\varphi) + \frac{\sin t_\theta \omega}{\sin\omega} \cdot \frac{\sin t_\varphi \omega}{\sin\omega} \rho(r, \theta + \Delta\theta, \varphi + \Delta\varphi). \end{aligned} \quad (9)$$

经过分析, 采用球壳上四元数插值方法和径向线性插值方法, 可以快速、准确地计算非对称三角形中任意直角网格点的密度函数值, 流程如下:

- (1) 从球心 O 出发, 作一条经过 E 的射线, 该射线与点 E 上下相邻的两个球壳分别相交于点 E_1, E_2 ;
- (2) 在球面四边形 $A_1B_1C_1D_1$ 上, 采用 (9) 式进行球面插值, 得到点 E_1 的密度函数值;
- (3) 同理, 在球面四边形 $A_2B_2C_2D_2$ 上, 得到点 E_2 的密度函数值;
- (4) 在线段 E_1EE_2 上, 采用沿径向的线性插值, 就得到点 E 的密度函数值。

图 5 密度函数快速计算模型

Figure 5 The fast calculation model of density function

3.3 基于两阶段映射法的快速对称映射方法

为了计算非对称三角形外其他区域的密度函数值, 首先, 需要根据 20 面体的 60 对称性, 将其他区域的网格点 (x, y, z) 绕 2、3、5 次轴对称旋转到如图 1 所示的非对称三角形中, 得到网格点 (x', y', z') . 然后, 根据非对称三角形中已知网格点的密度函数值, 采用三线性插值得到点 (x', y', z') 的密度函数值, 根据 60 对称性, 可知 $\rho(x, y, z) = \rho(x', y', z')$. 反复利用该方法, 就可以得到所有点的密度函数值.

对称映射操作涉及大量区域判断和矩阵向量乘法, 极其耗时. 因为 X 、 Y 、 Z 轴都是 2 次轴, 改变坐标符号就可把第 1、4 象限外的网格点映射到第 1、4 象限. 由此, 提出下述两阶段映射法来减少计算量.

第一阶段: 根据 20 面体的 60 对称性, 把第 1、4 象限中的网格点映射到非对称三角形中, 采用插值计算出这些点的密度函数值;

第二阶段: 改变坐标符号, 把第 1、4 象限外的网格点映射到第 1、4 象限中, 然后根据第一阶段的计算结果, 直接赋值即可得到第 1、4 象限外网格点的密度函数值.

在第一阶段, 该算法和原始算法的时间开销相同; 但在第二阶段, 第 1、4 象限外的点通过改变其坐标符号就可映射到第 1、4 象限, 时间开销相对原始算法可忽略不计. 所以, 该算法将原来在 8 个象限内进行的区域判断和矩阵向量乘法转化为在 2 个象限内进行, 使计算量减少了 4 倍.

3.4 密度函数快速计算模型

结合前面的内容, 可以得到一个密度函数快速计算模型, 如图 5 所示. 首先, 采用球坐标系网格点密度函数快速计算方法得到球坐标系下非对称三角形网格点处的密度函数值; 其次, 采用“球坐标系

— 直角坐标系” 网格点密度函数转换方法得到直角坐标系下非对称三角形网格处的密度函数值; 最后, 采用基于两阶段映射法的快速对称映射方法, 就可以得到直角坐标系下整个空间的密度函数值。

采用该模型计算密度函数时各阶段时间复杂度分析如下:

(1) 球坐标系网格点密度函数时间复杂度分析

在球坐标系网格点密度函数快速计算方法中, 四个阶段时间复杂度如下:

① $\int_0^\infty f_{l,u}(R) j_l(2\pi Rr) R^2 dR$ 的时间复杂度上界为 $O[(L_M)^2 R_M] = O[(L_M)^3]$, 需要计算 $r_m \sum_{l=0}^{L_M} u(l)$ 次, 所以该阶段时间复杂度为 $O[(L_M)^3 r_m \cdot \sum_{l=0}^{L_M} u(l)] = O[(L_M)^3 r_m \cdot L_M \cdot u(L_M)] = O[(L_M)^3 L_M \cdot L_M \cdot L_M] = O[(L_M)^6]$;

② 该阶段需要计算非对称三角形球面上所有网格点的极坐标值, 网格点个数为 $N_2 = \frac{1}{60} \cdot 4\pi r_m^2 = \frac{\pi}{15} r_m^2$, 则该阶段时间复杂度为 $O[r_m^2] = O[(L_M)^2]$;

③ 该阶段需要计算非对称三角形球面上所有网格点所有阶、所有重的 $S_{l,u}(\theta, \varphi)$, 网格点个数为 $N_2 = \frac{\pi}{15} r_m^2$, $S_{l,u}(\theta, \varphi)$ 的时间复杂度上界为 $O[(L_M)^3]$, 每个网格点需要计算的 $S_{l,u}(\theta, \varphi)$ 个数为 $\sum_{l=0}^{L_M} u(l)$, 所以该阶段时间复杂度为 $O[(L_M)^3 r_m^2 \cdot \sum_{l=0}^{L_M} u(l)] = O[(L_M)^3 r_m^2 \cdot L_M \cdot u(L_M)] = O[(L_M)^3 \cdot (L_M)^2 \cdot L_M \cdot L_M] = O[(L_M)^7]$ 。

④ 该阶段根据 ① 或 ② 结果计算非对称三角形球面上所有网格点的 $\rho(r, \theta, \varphi)$, 网格点个数为 $N_3 = N_2 \cdot r_m = \frac{\pi}{15} r_m^3$, 每个点的时间复杂度为 $O[\sum_{l=0}^{L_M} \sum_{u=1}^u 1] = O[(L_M)^2]$, 所以该阶段时间复杂度为 $O[r_m^3 \cdot (L_M)^2] = O[(L_M)^3 \cdot (L_M)^2] = O[(L_M)^5]$ 。

由此可知, 计算球坐标系网格点密度函数阶段的时间复杂度为 $O[(L_M)^7]$ 。

(2) “球坐标系—直角坐标系” 网格点密度函数转换方法时间复杂度分析

该阶段对非对称三角形中的每个网格点都执行四个操作, 这四个操作的时间复杂度都是 $O(1)$, 网格点总数为 $N_1 = \frac{\pi}{45} \cdot r_m^3$, 所以该阶段时间复杂度为 $O[r_m^3] = O[(L_M)^3]$ 。

(3) 基于两阶段映射法的快速对称映射方法时间复杂度分析

该阶段的运算主要是矩阵向量乘和坐标符号变换, 其中矩阵大小为 3×3 , 向量大小为 3×1 , 需要进行矩阵向量乘的网格点个数为 $N_4 = \frac{1}{4}(2r_m)^3 = 2r_m^3$, 需要进行坐标符号变换的网格点个数为 $N_5 = \frac{3}{4}(2r_m)^3 = 6r_m^3$. 两种运算均可在常数时间 C 内完成, 所以该阶段时间复杂度为 $O[C \cdot 2r_m^3 + C \cdot 6r_m^3] = O[r_m^3] = O[(L_M)^3]$ 。

综上所述, 采用该模型计算密度函数的时间复杂度为 $O[(L_M)^7]$, 较原始方法时间复杂度 $O[(L_M)^8]$ 降低了一个幂次。

4 实际验证

为了验证快速密度函数计算模型的性能, 我们采用模拟数据进行重构试验. 首先, 在密度函数计算阶段, 与原始密度函数计算方法进行精度和速度对比分析; 然后, 在整个三维重构阶段, 分别与原始 ISAF 算法、recISAFs 算法^[9]、传统 Fourier-Bessel 算法进行对比分析. 下述实验在一台 AMD 2.6GHz 双核 CPU, 2G 内存机器上进行。

4.1 数据介绍

我们从 PDB 数据库中选择葡枝青霉病毒 F(Penicillium stoloniferum virus F, Psv-F) 作为实验源. 该病毒具有二十面体对称结构, 其结构由 X 光晶体衍射得到, 分辨率为 $3.3\text{\AA}$, PDB 编号为 3ES5^[16].

图 6 加噪声前后的投影图片

Figure 6 Simulated projection images of Psv-F. (a) Without noise; (b) with a SNR of 0.1

图 7 不同 L_M 下的重构结果, 每幅图左边是原始方法的结果, 右边是新方法的结果

Figure 7 The reconstructed maps at the different L_M , the left in each picture is the reconstructed map with the original method, the right in each picture is the reconstructed map with the fast calculation model. (a) $L_M = 100$; (b) $L_M = 150$; (c) $L_M = 200$; (d) $L_M = 250$

为了生成模拟数据, 首先采用 Eman 的 pdb2mrc^[17] 将其转换为 MRC 文件, 参数为 apix=1.3, res=3.8, 生成结构大小为 $304 \times 304 \times 304$. 然后采用 IMIRS 的 ProjectIcosFile^[18] 生成 2000 个随机投影, 最后加上信噪比为 0.1 的噪声, 如图 6 所示.

4.2 密度函数计算部分性能比较

我们分别采用原始方法和我们的模型计算了不同 L_M 下的密度函数值. 两种方法得到的重构结果 (完整结构) 如图 7 所示, 从表面看不出明显差异.

为了进一步验证新模型重构结果的精度, 分别采用 (10)(11) 式计算原始方法和新方法重构结果 ($L_M = 250$) 在实空间、倒空间的相似性. 在 (10) 式中, N 表示结构中点的个数, ρ_A^i 、 ρ_B^i 分别表示分子 A 和分子 B 中第 i 个点的密度函数值, 该公式可以计算分子 A 和分子 B 在某一分辨率下的实空

图 8 重构结果在实空间和倒空间中相似性和分辨率的关系

Figure 8 The similarities between reconstructed maps from the two methods at the different resolutions in the real space and the reciprocal space. (a) The similarities in the real space; (b) The similarities in the reciprocal space

表 1 当 L_M 不同时, 原始算法和新算法计算密度函数值消耗的时间**Table 1** The running time of calculating density function with the two methods at the different L_M

	Index 100	Index 150	Index 200	Index 250
Target resolution(Å)	12.309375	8.20625	6.1546875	4.92375
The running time with the original method(s)	188.533	547.144	1181.406	2153.379
The running time with the new method(s)	3.891	5.592	8.767	12.779
Speedup	48.45	97.84	134.76	168.51

间相似性^[19], 对不同分辨率下的结果应用该公式, 就可依次得到不同分辨率下对应的相似性. 在 (11) 式中, $F_A(r)$ 、 $F_B(r)$ 分别表示分子 A 和分子 B 在傅里叶球壳 $r = r_i$ 上任意点的结构因子值, 该公式称为 FSC^[20](Fourier shell correlation), 可以计算分子 A 和分子 B 在倒空间不同傅里叶球壳上的相似性.

$$S_{A,B} = \frac{\sum_{i=1}^N \rho_A^i \cdot \rho_B^i}{\sqrt{\sum_{i=1}^N (\rho_A^i)^2 \sum_{i=1}^N (\rho_B^i)^2}}, \quad (10)$$

$$\text{FSC}_{A,B}(r_i) = \frac{\sum_{r \in r_i} F_A(r) \cdot F_B(r)^*}{\sqrt{\sum_{r \in r_i} F_A^2(r) \cdot \sum_{r \in r_i} F_B^2(r)}}. \quad (11)$$

(10)(11) 式的结果如图 8 所示, 可以看出, 两种方法重构结果在实空间的相似性在 99.3% 以上, 在倒空间的相似性在 98.8% 以上, 所以新方法不会影响重构结果的精度.

这两种方法的计算时间如表 1 所示. 可知, 我们提出的快速计算模型在保证精度的前提下, 可以提高密度函数的计算速度, 加速比最高可以达到 170 倍左右, 而且随着分辨率的增加而进一步增加.

4.3 三维重构整体计算性能比较

为了验证新密度函数计算模型对三维重构整体性能的提升, 我们分别对比了采用新模型 ISAF 算法、原始 ISAF 算法、recISAFs 算法和 Fourier-Bessel 算法. 我们进行了四组不同分辨率的实验, 目标分辨率分别是 32.825Å、13.13Å、6.565Å、3.9Å. 为了便于比较, 我们将葡枝青霉病毒 F 结构在上述分

图 9 采用不同算法, 在不同目标分辨率下的重构结果

Figure 9 The reconstructed maps with the different algorithms at the different target resolutions

分辨率下进行滤波, 分别得到了四种分辨率下的标准三维结构。

上述实验的重构结果如图 9 所示, 第一列是标准结构, 第二列是采用新模型 ISAF 算法的重构结果, 第三列是原始 ISAF 算法的重构结果, 第四列是 recISAFs 算法的重构结果, 第五列是 Fourier-Bessel 算法的重构结果。从上向下, 是不同分辨率下的重构结果, 对应的目标分辨率依次是: 32.825 Å、13.13 Å、6.565 Å、3.9 Å。可以看出, 采用新模型 ISAF 算法的重构结果与其他方法得到的结果没有明显差异。

我们分别计算目标分辨率为 3.9 Å 时这四种方法重构结果的 FSC 曲线^[20], 如图 10 所示。采用新模型 ISAF 算法、原始 ISAF 算法、recISAFs 算法和 Fourier-Bessel 算法的分辨率分别是 3.9 Å、3.94 Å、3.98 Å 和 4.19 Å, 这表明采用新模型获得的重构结果与标准结构最匹配。

为了分析该计算模型的速度优势, 首先以 3.9 Å 为目标分辨率, 统计当投影图片数量不同时, 这四种方法的耗时, 结果如表 2 所示。然后设置投影图片数量为 2000, 统计不同目标分辨率下, 这四种方法的耗时, 结果如表 3 所示。

可以看出, 在保证精度的前提下, 采用新模型的 ISAF 算法可以大幅度降低密度函数计算的运行时间, 进而提高了整个三维重构阶段的速度, 其速度一般是原始 ISAF 算法的 30 倍左右。而且, 随着图片数量的增加与目标分辨率的提高, 采用新模型的 ISAF 算法的优势越明显, 其加速比将进一步提高。

图 10 目标分辨率为 3.9 埃时的 FSC 曲线

Figure 10 The FSC curves of the reconstructed maps with four method at the target resolution of 3.9 Å

表 2 投影图片数量不同时四种重构方法耗时 (单位: s)

Table 2 The running times with different number of projections at the target resolution of 3.94 Å (unit: s)

	Accelerated ISAF algorithm	Original ISAF algorithm	recISAFs algorithm	Fourier-Bessel algorithm
250 projections	241.486	7798.473	696.769	551.573
500 projections	246.871	7824.061	702.855	695.259
1000 projections	252.104	7863.921	718.489	956.190
2000 projections	259.673	7946.828	768.853	1583.489

表 3 不同目标分辨率下四种重构方法耗时 (单位: s)

Table 3 The running times with 2000 projections at the different target resolutions (unit: s)

	Accelerated ISAF algorithm	Original ISAF algorithm	recISAFs algorithm	Fourier-Bessel algorithm
32.825 Å	15.454	81.442	151.821	36.386
13.13 Å	25.441	344.938	174.327	92.055
6.565 Å	52.352	1913.249	255.106	383.569
3.9 Å	259.673	7946.828	768.853	1583.489

5 结论与展望

计算密度函数在 ISAF 三维重构算法中极其耗时, 已经成为该算法瓶颈之一. 通过分析密度函数的计算过程, 我们提出一种密度函数快速计算模型, 可以将密度函数计算阶段的时间复杂度由 $O[L_M^8]$ 降低到 $O[L_M^7]$. 该模型包括下述三个组成部分:

- (1) 球坐标系网格点密度函数快速计算方法: 可以快速得到球坐标系下所有网格点的密度函数值;
- (2) “球坐标系一直角坐标系” 网格点密度函数转换方法: 实现球坐标系网格点密度函数值与直角坐标系网格点密度函数值的等效转换;

(3) 基于两阶段映射法的快速对称映射方法: 将原始算法中网格点对称映射操作的计算量降低到原来的 $1/4$ 。

我们采用葡枝青霉病毒 F 的模拟数据进行重构实验, 结果表明, 在保证精度的前提下, 我们的方法与原始 ISAF 算法相比, 密度函数计算部分的速度可以提高 2 个数量级, 三维重构整体速度可以提高 30 倍, 并且随着数据规模的增大和重构精度的提高, 该模型带来的加速比进一步增大。

我们的方法虽然将密度函数的计算复杂性降低了一个幂次, 但该阶段依然是 ISAF 重构算法的计算瓶颈, 如何进一步降低该阶段的计算复杂性, 继续提高 ISAF 重构算法的性能, 将是下一步的研究重点。

参考文献

- 1 De Rosier D J, Klug A. Reconstruction of three-dimensional structures from electron micrographs. *Nature*, 1968, 217: 130–134
- 2 Colsher J G. Iterative three-dimensional image reconstruction from tomographic projections. *Comput Graph Image Process*, 1977, 6: 513–537
- 3 Frank J, Goldfarb W, Eisenberg D, et al. Reconstruction of glutamine synthetase using computer averaging. *Ultramicroscopy*, 1978, 3: 283–290
- 4 Dubochet J, Adrian M, Chang J J, et al. Cryo-electron microscopy of vitrified specimens. *Q Rev Biophys*, 1988, 21: 129–228
- 5 Chiu W. Electron microscopy of frozen, hydrated biological specimens. *Annu Rev Biophys Biophys Chem*, 1986, 15: 237–257
- 6 Crowther R A. Procedures for three-dimensional reconstruction of spherical viruses by Fourier synthesis from electron micrographs. *Philos Trans Roy Soc B*, 1971, 261: 221–230
- 7 Crowther R A, DeRosier D J, Klug A. The reconstruction of a three-dimensional structure from projections and its application to electron microscopy. *P Roy Soc Lond A Mat*, 1970, 317: 319–340
- 8 Navaza J. On the three-dimensional reconstruction of icosahedral particles. *J Struct Biol*, 2003, 144: 13–23
- 9 Liu H R, Cheng L P, Zeng S J, et al. Symmetry-adapted spherical harmonics method for high-resolution 3D single-particle reconstructions. *J Struct Biol*, 2008, 161: 64–73
- 10 Liu H R. The new method for high-resolution 3D reconstruction of virus. Ph. D. Thesis. Xiangtan: Xiangtan University. 2008. 2[刘红荣. 高分辨病毒三维重构的新方法. 博士论文. 湘潭: 湘潭大学, 2008. 2]
- 11 Baker T S, Olson N H, Fuller S D. Adding the third dimension to virus life cycles: three dimensional reconstruction of icosahedral viruses from cryo-electron micrographs. *Microbiol Mol Biol R*, 1999, 63: 862–922
- 12 Abramowitz M, Stegun I A. *Handbook of Mathematical Functions with Formulas, Graphs and Mathematical Tables*. New York: Dover Press, 1965
- 13 Bradley C J, Cracknell A P. *The Mathematical Theory of Symmetry in Solids*. New York: Oxford University Press, 1972
- 14 Prandl W, Schiebel P, Wulf K. A recursive algorithm for the generation of symmetry-adapted functions: principles and applications to the icosahedral group. *Acta Crystallogr A*, 1996, 52: 171–175
- 15 Jackson J D. *Classical Electrodynamics*. 3rd ed. New York: Wiley Press, 1999
- 16 Pan J, Dong L, Lin L, et al. Atomic structure reveals the unique capsid organization of a dsRNA virus. *P Natl Acad Sci USA*, 2009, 106: 4225–4230
- 17 Ludtke S J, Baldwin P R, Chiu W. EMAN: semi-automated software for high-resolution single-particle reconstructions. *J Struct Biol*, 1999, 128: 82–97
- 18 Liang Y R, Ke E Y, Zhou Z H. IMIRS: a high-resolution 3D reconstruction package integrated with a relational image database. *J Struct Biol*, 2002, 137: 292–304
- 19 Willett P. Chemical similarity searching. *J Chem Inf Comp Science*. 1998, 38: 983–996
- 20 Harauz G, van Heel M. Exact filters for general geometry three dimensional reconstruction. *Optik*, 1986, 73: 146–156

A fast calculation strategy of density function in ISAF reconstruction algorithm

WANG GongMing^{1,2}, ZHANG Fa^{1*}, FAN LiYa³, SUN Fei⁴ & LIU ZhiYong^{1*}

¹ *Institute of Computing Technology, Chinese Academy of Sciences, Beijing 100190, China;*

² *Graduate University of Chinese Academy of Sciences, Beijing 100190, China;*

³ *IBM China Research Lab, Beijing 100094, China;*

⁴ *Institute of Biophysics, Chinese Academy of Sciences, Beijing 100101, China*

*E-mail: zhangfa@ict.ac.cn, zyliu@ict.ac.cn

Abstract The ISAF reconstruction algorithm is a new method for reconstructing icosahedral molecules from their projections. This algorithm works in spherical coordinate system and can achieve higher resolution than the traditional Fourier-Bessel algorithm in cylindrical coordinate system; however this method needs huge computations, which limits its application in reality. The main bottleneck lies in the calculation of density function as it occupies 90% running time of the whole algorithm. A fast calculation strategy of density function is proposed to solve this problem. This strategy is composed of three components: the fast calculation method of density function of mesh point in spherical coordinate system, the transformation method of density function of mesh point from spherical coordinate system to Cartesian coordinate system and the fast two-phase mapping method. The time complexity of calculating density function is decreased from $O[(L_M)^8]$ to $O[(L_M)^7]$ in our strategy. The experimental results on Psv-F simulated data indicate that the speed of calculating density function is increased almost two orders of magnitude and the speedup of the whole algorithm could reach 30 times. In addition, the speedup could go up with the increase in the number of images and the requirement of accuracy.

Keywords ISAF, 3D reconstruction, density function, spherical coordinate system, quaternion interpolation

WANG GongMing was born in 1981. He received the Ph.D. degree in Institute of Computing Technology from the Chinese Academy of Sciences, Beijing, in 2012. Currently, he is a Research Assistant Professor at Tsinghua University. His research interests include digital image processing, computer graphics, bioinformatics, and visualization.

FAN LiYa was born in 1982. He received the Ph. D. degree in computer science from the Institute of Computing Technology, Chinese Academy of Sciences, Beijing, in 2010. Currently, he is a staff researcher at IBM China Laboratory. His research interests include distributed computing, data mining, and optimization theory.

ZHANG Fa was born in 1974. He is an Associate Professor of Institute of Computing Technology, Chinese Academy of Sciences. His research interests include high performance algorithms and bioinformatics.

LIU ZhiYong was born in 1946. He is a Professor of Institute of Computing Technology, Chinese Academy of Sciences, and a senior member of China Computer Federation. His research interests include high performance algorithm and architecture, and parallel processing.