

# 基于 NURBS 曲线插值的实时大气折射误差修正

赵晓飞<sup>1,2</sup> 韦明智<sup>1</sup> 何俊华<sup>1</sup> 陈良益<sup>1</sup>

(1 中国科学院西安光学精密机械研究所, 西安 710068)

(2 中国科学院研究生院, 北京 100039)

**摘要** 提出了一种采用传统的逐点描述法和 NURBS 曲线插值方法相结合的技术, 对多种可见光测量设备进行高准确度的实时大气折射误差修正方法. 该方法克服了逐点描述法计算量大、无法达到实时计算要求等缺点, 尤其适用于全自动胶片判读仪或光电经纬仪中实时数据的大气折射误差修正.

**关键词** 大气折射误差修正; NURBS 曲线; 插值; 逐点描述法; 测角误差

**中图分类号** TN011.2 **文献标识码** A

## 0 引言

随着武器命中准确度要求的日益提高, 对武器弹道测量系统准确度的要求也不断提高. 武器弹道测量准确度主要取决于外测系统本身的测量准确度, 而外测系统的准确度则由各个误差源所决定. 随着外测系统中设备的改进和数据处理方法的进一步优化, 大气折射误差<sup>[1]</sup>修正准确度已成为外测系统实现高准确度的主要限制因素之一. 小步长的逐点描述法<sup>[2]</sup>是得到大气折射误差高准确度修正的最好方法, 然而这种方法的计算量很大, 不适合实时和大量数据修正的场合.

NURBS(Non-Uniform Rational B-Spline, 非均匀有理 B 样条)曲线插值方法<sup>[3]</sup>是当前自由曲线和曲面描述领域中应用最广泛的技术, 它具有良好的几何性质和计算性质, 可以精确地表示工程实用中最常用的圆、椭圆等二次曲线和球、椭球、锥台等二次曲面, 并可通过调节权因子来十分方便地控制曲线和曲面的形状, 使它已成为目前计算机图形学中最常用的造型方法之一. 与传统的插值方法相比, 它具有计算简单、可靠有效等优点.

为了实时、精确地修正大气折射误差, 我们提出了一种结合逐点描述法和 NURBS 曲线插值方法的大气折射误差修正方法. 它综合了逐点描述法的精确性和 NURBS 曲线插值方法计算量小、可靠等优点, 可以快速得到高准确度的大气折射误差修正数据.

## 1 逐点描述法修正大气折射误差

假设目标  $M$  在测量系统的实际高低角为  $\beta_1$ , 海拔高为  $h_M$ , 如图 1. 由于大气折射率随海拔高度的变化在不断的变化, 光线在传输过程中不断的发生折射, 使得在地面观测点  $C_0$  点观测到的高低角为

$\beta_2$ , 由此产生的测角误差  $\Delta\beta = \beta_1 - \beta_2$ .

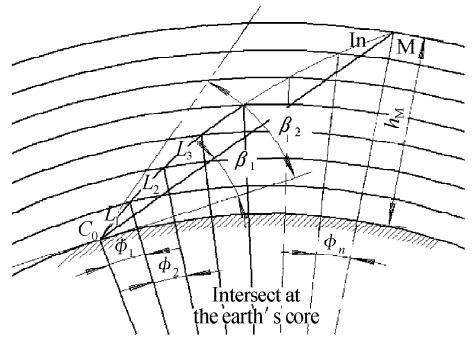


图 1 大气折射  
Fig. 1 Atmosphere refraction

为计算测角误差  $\Delta\beta$ , 可以参照图 1, 将光线实际的传播路径弧  $C_0 M$  平均分为  $n$  个小段, 当  $n$  足够大时, 可以认为每一段小的弧均为一条直线. 如图 2, 近似的认为弧  $C_0 C_1$  为直线.

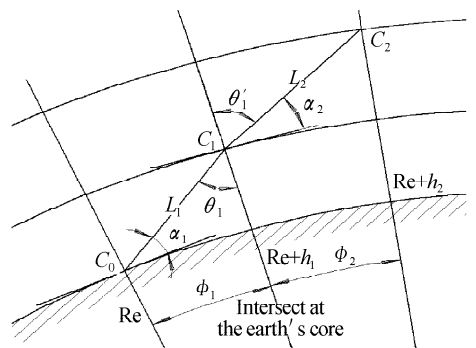


图 2 角度误差修正计算  
Fig. 2 Calculation of angle error correction

在  $\Delta C_0 O C_1$  中, 已知:  $C_0 D = R_{\text{earth}}$  (地球半径),  $C_0 C_1 = L_1$ ,  $\alpha_1 = \beta_2$ ,  $\angle C_1 C_0 O = \alpha_1 + 90^\circ$ , 地表折射率为  $n_1$ ; 根据余弦定理可以求得

$$C_1 O = \sqrt{R_{\text{earth}}^2 + L_1^2 - 2R_{\text{earth}} L_1 \cos(\alpha_1 + 90^\circ)} \quad (1)$$

再利用正弦定理, 可以求得

$$\angle C_0 C_1 O = \arcsin\left(\frac{R_{\text{earth}} \sin(\alpha_1 + 90^\circ)}{R_{\text{earth}} + h_1}\right) = \theta_1 \quad (2)$$

$$\angle C_0 O C_1 = \arcsin\left(\frac{L_1 \sin(\alpha_1 + 90^\circ)}{R_{\text{earth}} + h_1}\right) = \phi_1 \quad (3)$$

$C_1$  点的海拔高度为  $h_1 = C_1O - R_{\text{earth}}$ ; 根据折射率计算公式<sup>[2]</sup>可以求得  $C_1$  点处的大气折射率  $n_2$ , 当  $L_1$  足够小时,  $h_1$  也非常小, 因此可以近似认为在  $h_1$  范围内的大气折射率是恒定的.

光线在  $C_1$  点发生折射时, 已知入射角  $\theta_1$ 、 $C_1$  点下层大气折射率  $n_1$  和  $C_1$  点上层大气折射率  $n_2$ , 利用折射定理可以求得折射角

$$\theta'_1 = \arcsin\left(\frac{n_1 \sin \theta_1}{n_2}\right) \quad (4)$$

对于  $C_1$  点来说, 光线的高低角  $\alpha_2 = 90^\circ - \theta'_1$ ,  $C_1O = R_{\text{earth}} + h_1$ ,  $L_2 = L_1$ ,  $C_1$  点上层大气折射率  $n_2$ ; 根据这些条件, 可以利用上述的式(1)~(4)来计算光线在  $C_2$  点的折射情况.

依此类推, 光线经过  $n$  层大气后, 最终可求得  $M$  的实际海拔高度  $h_M$ ,  $n$  次折射中地球曲率修正角  $\phi = \sum_{i=1}^n \phi_i$ . 如图 1, 解  $\triangle C_0OM$ , 得  $\angle OC_0M$

$$\angle OC_0M = \arcsin\left(\frac{L \sin \phi}{R_{\text{e}} + h_M}\right)$$

式中

$$L = C_0M =$$

$$\sqrt{R_{\text{earth}}^2 + (R_{\text{earth}} + h_M)^2 - 2R_{\text{earth}}(R_{\text{earth}} + h_M)\cos \phi}$$

目标  $M$  在测量系统的实际的高低角为  $\beta_1 = \angle OC_0M$ , 测角误差  $\Delta\beta = \beta_1 - \beta_2$ .

由上述原理和计算公式可知, 逐点描述法仅有两个误差源: 一是分层时直线代替分段曲线引入的误差, 二是假设每层内大气折射率不变引入的误差. 当步长减小时, 层数增加, 上述的两个误差都将减小, 修正准确度提高; 但是相应的计算量也会成倍增加. 表 1 为在 CPU: P-IV 2.4 GHz、内存: 256M DDR 的计算环境下, 不同步长逐点描述法计算时间的比较. 从表 1 可知, 逐点描述法步长越小, 所需的计算时间就越长.

表 1 逐点描述法在不同步长下计算时间的比较

角度 时间/s	$\alpha_1$ ( $^\circ$ )	$\alpha_2$ ( $^\circ$ )	$\alpha_3$ ( $^\circ$ )	$\alpha_4$ ( $^\circ$ )	$\alpha_5$ ( $^\circ$ )	$\alpha_6$ ( $^\circ$ )	$\alpha_7$ ( $^\circ$ )	平均值 (s)
步长 1	0.030	0.030	0.030	0.030	0.030	0.030	0.030	0.030
步长 2	0.331	0.330	0.330	0.336	0.333	0.330	0.331	0.332
步长 3	1.162	1.202	1.172	1.072	1.272	1.172	1.282	1.191
步长 4	9.374	9.524	9.253	10.787	11.046	10.996	11.678	10.380
步长 5	45.976	47.399	43.813	47.899	47.654	43.442	51.474	46.808

注: 步长 1:  $d/200$ ; 步长 2:  $d/2000$ ; 步长 3:  $d/4000$ ; 步长 4:  $d/10000$ ; 步长 5:  $d/20000$ .  $d$  为从目标  $M$  到测站的实测距离.  $\alpha_i$  为高低角,  $i=1, 2, \dots, 7$ .

## 2 NURBS 曲线插值原理及算法

NURBS 曲线是用分段有理 B 样条多项式定义的, 其形式如下

$$p(u) = \frac{\sum_{i=0}^n \omega_i d_i N_{i,k}(u)}{\sum_{i=0}^n \omega_i N_{i,k}(u)} \quad (5)$$

式中  $d_i$  是控制点矢量,  $\omega_i$  是控制点相应的权因子.  $N_{i,k}(u)$  是  $k$  次 B 样条基函数, 是由节点矢量  $U = [u_0, u_1, \dots, u_{n+k+1}]$  按照以下的递推公式定义的

$$\begin{cases} N_{i,0}(u) = \begin{cases} 1 & (\text{若 } u_i \leq u < u_{i+1}) \\ 0 & (\text{其它}) \end{cases} \\ N_{i,k}(u) = \frac{u - u_i}{u_{i+k} - u_i} N_{i,k-1}(u) + \frac{u_{i+k+1} - u}{u_{i+k+1} - u_{i+1}} N_{i+1,k-1}(u) \\ \text{规定 } \frac{0}{0} = 0 \end{cases} \quad (6)$$

$N_{i,k}(u)$  的双下标  $k$  表示次数, 第一下标  $i$  表示序号. 该递推公式表明, 欲确定第  $i$  个  $k$  次 B 样条  $N_{i,k}(u)$ , 需要用到  $u_i, u_{i+1}, \dots, u_{i+k+1}$  共  $k+2$  个节点.

称区间  $[u_i, u_{i+k+1}]$  为  $N_{i,k}(u)$  的支承区间.  $N_{i,k}(u)$  的第一下标等于支承区间左端节点的下标, 即表示该 B 样条在参数  $u$  轴上的位置. 曲线方程中相应  $n+1$  个控制顶点  $d_i, i=0, 1, \dots, n$ , 要用到  $n+1$  个  $k$  次 B 样条基函数  $N_{i,k}(u), i=0, 1, \dots, n$ . 它们每个都是  $k$  次 B 样条. 它们的支承区间所含节点的并集就是定义这一组 B 样条基的节点矢量  $U = [u_0, u_1, \dots, u_{n+k+1}]$ .

给定控制顶点  $d_i (i=0, 1, \dots, n)$ 、次数  $k$  及节点矢量  $U = [u_0, u_1, \dots, u_{n+k+1}]$  后, 求该 B 样条曲线上对应参数  $u \in [u_i, u_{i+1}] \subset [u_k, u_{n+1}]$  的点  $q(u)$ . 虽然可以采用将给定参数  $u$  直接带入曲线定义式的方法求出, 但是要反复叠代来求出各基函数的值. 因此用软件实现上述算法计算时间较长, 不利于大气折射误差修正系统的实时性. 为此采用更加快捷省时的德布尔算法<sup>[3]</sup>. 德布尔算法求对应点  $q(u)$  的递推公式如下

$$q(u) = \sum_{j=0}^i d_j N_{j,k}(u) = \sum_{j=i-k}^i d_j N_{j,k}(u) = \sum_{j=i-k}^{i-1} d_j^l N_{j,k-l}(u) = \dots = d_i^k$$

$$u \in [u_i, u_{i+1}] \subset [u_k, u_{n+1}]$$

$$d_j^l = \begin{cases} d_j, & l=0 \\ (1-a_j^l)d_j^{l-1} + a_j^l d_{j+1}^{l-1}, & j=i-k, i-k+1, \dots, i-l; l=1, 2, \dots, k \end{cases}$$

$$a_j^l = \frac{u - u_{j+l}}{u_{j+k+1} - u_{j+l}} \quad \text{规定 } \frac{0}{0} = 0$$

可以利用上述算法计算权因子  $\omega_i$  为常数的 NURBS 曲线, 并可以利用该算法计算不同距离、不同高低角时大气模型的高低角修正值。

### 3 修正大气折射误差的新方法

逐点描述法是一种修正大气折射产生误差的方法, 当步长  $L$  取值非常小时, 这种方法得到的修正结果是非常精确的。但是, 这种方法的计算量较大, 步长  $L$  越小、折射次数越多。这种算法的计算量将成倍增加, 在对大量数据进行修正时, 无法实现实时的误差修正。因此, 逐点描述法不能直接应用于实时误差修正。

NURBS 曲线不仅能够精确地表示各类曲线, 而且其形状控制能力也十分强大、灵活。因此, 可以采用 NURBS 插值与逐点描述相结合的方法来修正由于大气折射所产生的误差。具体步骤如下。

首先, 利用小步长的逐点描述法计算一些修正准确度非常高的顶点  $d_i$ , 并将这些点存储起来, 作

为 NURBS 曲线插值的控制顶点(这些控制顶点采用小步长的逐点描述法计算, 计算量较大, 但可以事先计算好后存储起来, 以备在后续的计算中使用。因此, 可以不将控制顶点计算时间记入运算时间)。

其次, 利用控制顶点计算 NURBS 曲线插值算法中需要的节点矢量  $U$ , 并将待测点数据带入节点矢量参数值反演程序, 得到待计算数据对应的节点矢量参数值。

最后, 利用 NURBS 曲线插值算法和上一步得到的节点矢量参数值来进行插值计算, 得到待测点的大气误差修正结果。

### 4 新方法的验证及结果分析

对上述方法进行验证。计算环境: CPU: P-IV 2.4GHz、内存: 256M DDR、操作系统: Windows2000 Professional、计算工具: Matlab 6.1。

1) 同一距离下, 不同角度利用 3 次 NURBS 曲线插值算法和逐点描述法得到的修正结果和计算时间比较。地表环境为  $P=1013.3 \text{ pPa}$ ,  $T=15^\circ\text{C}$ ,  $e=10.2 \text{ hPa}$ , 实测距离为  $d=20 \text{ km}$ , 实测高低角分别为  $\alpha_i, i=1, 2, \dots, 7$  时, 采用不同算法得到的结果比较如表 2。

表 2 同距离不同角度各种方法修正结果

方法 \ 角度	$\alpha_1$	$\alpha_2$	$\alpha_3$	$\alpha_4$	$\alpha_5$	$\alpha_6$	$\alpha_7$	
	( $^\circ$ )	( $^\circ$ )	( $^\circ$ )	( $^\circ$ )	( $^\circ$ )	( $^\circ$ )	( $^\circ$ )	
1	修正结果( $''$ )	72.2036	61.0549	49.0166	36.9655	26.0938	19.5657	8.3632
	计算时间(s)	0.030	0.030	0.030	0.030	0.030	0.030	0.030
2	修正结果( $''$ )	72.5251	61.3036	49.2041	37.0997	26.1846	19.6324	8.3911
	计算时间(s)	0.331	0.330	0.330	0.336	0.333	0.330	0.331
3	修正结果( $''$ )	72.5430	61.3174	49.2145	37.1072	26.1896	19.6361	8.3926
	计算时间(s)	1.162	1.202	1.172	1.072	1.272	1.172	1.282
4	修正结果( $''$ )	72.5606	61.3305	49.2242	37.1139	26.1941	19.6393	8.3939
	计算时间(s)	9.374	9.524	9.253	10.787	11.046	10.996	11.678
5	修正结果( $''$ )	72.5571	61.3286	49.2245	37.1139	26.1944	19.6396	8.3940
	计算时间(s)	0.020	0.020	0.020	0.020	0.020	0.020	0.020
6	修正结果( $''$ )	72.5573	61.3285	49.2229	37.1132	26.1937	19.6390	8.3938
	计算时间(s)	45.976	47.399	43.813	47.899	47.654	43.442	51.474

注: 方法 1: 步长为  $(d/200)$  米的逐点描述法; 方法 2: 步长为  $(d/2000)$  米的逐点描述法; 方法 3: 步长为  $(d/4000)$  米的逐点描述法; 方法 4: 步长为  $(d/10000)$  米的逐点描述法; 方法 5: 81 个控制顶点的 3 次 NURBS 曲线插值算法(利用步长为  $(d/20000)$  米的逐点描述法获得控制顶点); 方法 6: 步长为  $(d/20000)$  米的逐点描述法;

2) 同一角度下, 不同距离利用 3 次 NURBS 曲线插值算法和逐点描述法得到的修正结果和计算时间比较。地表环境为  $P=1013.3 \text{ pPa}$ ,  $T=15^\circ\text{C}$ ,  $e=10.2 \text{ hPa}$ , 实测高低角为  $\alpha=5^\circ$ , 实测距离分别为  $d_i, i=1, 2, \dots, 7$  时, 采用不同算法得到的结果比较如表 3。

3) NURBS 插值与逐点描述结合算法分析。由

表 2 和表 3 可以看出, 获得高准确度修正数据的逐点描述法的计算量非常大。以  $(d/20000)$  作为步长的逐点描述法需要计算 20000 个循环, 每计算一次数据所需的时间大约在 50 s 左右; 以  $(d/10000)$  作为

表 3 同角度不同距离各种方法修正结果

方法	距离	$d_1$	$d_2$	$d_3$	$d_4$	$d_5$	$d_6$	$d_7$
	/km	/km	/km	/km	/km	/km	/km	/km
1	修正结果(")	8.9272	19.3056	28.4212	39.1615	49.2819	62.0158	69.8717
	计算时间(s)	0.020	0.020	0.020	0.020	0.020	0.020	0.020
2	修正结果(")	8.9327	19.3926	28.4217	39.1650	49.2830	62.3272	69.8467
	计算时间(s)	0.320	0.381	0.380	0.391	0.385	0.340	0.389
3	修正结果(")	8.9290	19.3981	28.4360	39.1699	49.2955	62.3122	69.8642
	计算时间(s)	1.222	1.362	1.332	1.551	1.792	1.513	1.732
4	修正结果(")	8.9303	19.4023	28.4328	39.1756	49.3079	62.3278	69.8748
	计算时间(s)	9.373	10.203	10.546	12.0480	12.929	12.1270	11.696
5	修正结果(")	8.9304	19.4017	28.4353	39.1777	49.3028	62.3248	69.8750
	计算时间(s)	0.010	0.010	0.010	0.010	0.010	0.010	0.010
6	修正结果(")	8.9308	19.4019	28.4354	39.1778	49.3054	62.3247	69.8752
	计算时间(s)	49.431	46.367	49.582	50.5830	51.093	50.482	51.314

注:方法 1:步长为( $d_i/200$ )米的逐点描述法;方法 2:步长为( $d_i/2000$ )米的逐点描述法;方法 3:步长为( $d_i/4000$ )米的逐点描述法;方法 4:步长为( $d_i/10000$ )米的逐点描述法;方法 5:21 个控制顶点的 3 次 NURBS 曲线插值算法(利用步长为( $d_i/20000$ )米的逐点描述法获得控制顶点);方法 6:步长为( $d_i/20000$ )米的逐点描述法。

步长的逐点描述法需要计算 10000 个循环,每计算一次数据所需的时间大约在 10 s 左右;以( $d/4000$ )作为步长的逐点描述法需要计算 4000 个循环,每计算一次数据所需的时间大约在 1.2 s 左右;假设实测地点的地表环境为  $P=1013.3$  pPa,  $T=15^\circ\text{C}$ ,  $e=10.2$  hPa;实测距离为  $d=20$  km,实测高低角从  $5^\circ$  到  $85^\circ$  时,分别采用步长为  $d/2000$ 、 $d/4000$ 、 $d/10000$  和  $d/20000$  逐点描述法( $1^\circ$  间隔)得到的高低角修正曲线,并将步长为  $d/20000$  的逐点描述法得到的结果作为 81 个控制顶点的 3 次 NURBS 方法的控制顶点,得到高低角修正曲线,如图 3、4(图 4 为图 3 的局部放大图)所示。

4000 作为步长的逐点描述法得到的结果;然而采用 3 次 NURBS 插值算法计算一次修正仅需要 0.01~0.02 s 的时间,以作为步长的逐点描述法计算一次修正需要 10 s 左右,两者的计算量相差 500 倍以上。

从图 3、图 4 结合表 2、表 3 可以看出,利用 3 次 NURBS 插值算法(步长为  $d/20000$  的逐点描述法得到控制顶点)计算的结果大多数与  $d/10000$  作为步长的逐点描述法得到的结果近似,远优于以  $d/$

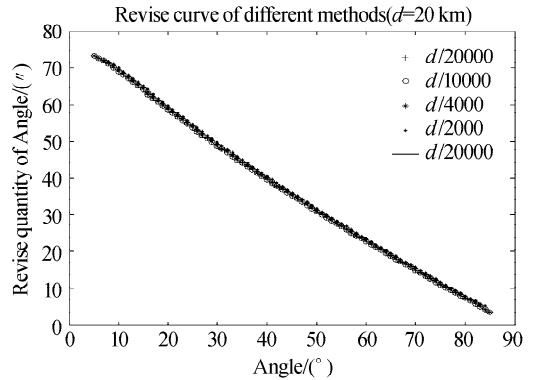
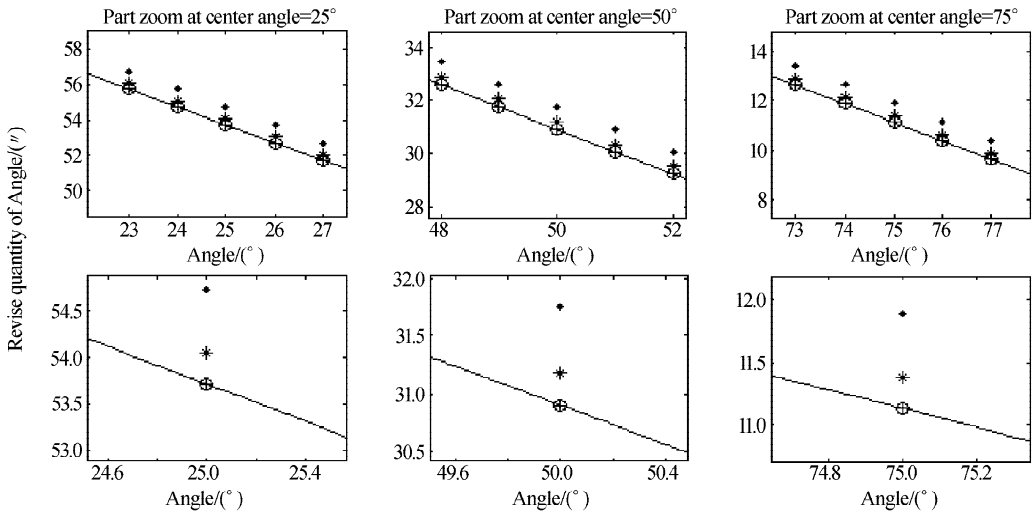


图 3 NURBS 曲线算法拟合大气修正曲线  
Fig. 3 NURBS curve algorithm fitting atmosphere correction curve



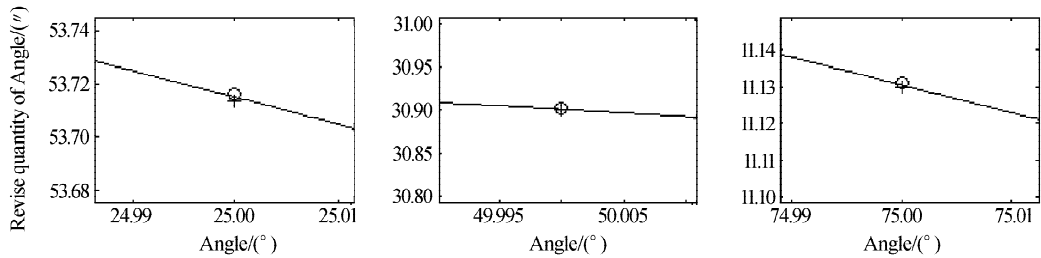


图 4 NURBS 曲线算法拟合大气修正曲线局部放大图

Fig. 4 Partial magnified figure of NURBS curve algorithm fitting atmosphere correction curve

## 5 结论

由上述计算和实际测试可知, NURBS 插值与逐点描述相结合的方法来修正由于大气折射所产生的误差, 虽然不能得到与同准确度逐点描述法相同的结果, 但在相同的准确度下可以大大提高计算效率, 将运算速度提高 500~1000 倍。

鉴于 NURBS 插值与逐点描述相结合的方法具有高准确度、简单、快速等特点, 完全可以应用于全自动胶片判读仪或光电经纬仪实时数据判读的大气折射误差修正。

## 参考文献

- 1 Thayer G D. A modified equation for radio refractivity of air. *Radio Science*, 1974, **9**(10): 803~807
- 2 黄捷. 电波大气折射误差修正. 北京: 国防工业出版社, 1999  
Huang J. Correction for Atmosphere Refractive Error of Radio Wave. Beijing: National Defence Industry Press, 1999
- 3 施法中. 计算机辅助几何设计与非均匀有理 B 样条 (CAGD&NURBS). 北京: 北京航空航天大学出版社, 1994  
Shi F Z. Computer Aid Geometry Design and Non-Uniform Rational B-Spline. Beijing: Beijing University of Aeronautics and Astronautics Press, 1994

# Real Time Correction of Atmospheric Refractive Error Based on NURBS Curve Interpolation

Zhao Xiaofei<sup>1,2</sup>, Wei Mingzhi<sup>1</sup>, He Junhua<sup>1</sup>, Chen Liangyi<sup>1</sup>

1 Xi'an Institute of Optics and Precision Mechanics, Chinese Academy of Sciences, Xi'an 710068

2 The graduate School of Chinese Academy of Sciences, Beijing 100039

Received date: 2004-09-08

**Abstract** A technology, which integrated the traditional algorithm of tracing by every vertex with NURBS curve interpolation method, was put forward. This technology was applied in high precision and real time correction of atmospheric refractive error field of many visible light measure equipments. Some shortcomings such as large quantity calculation, nonreal time calculate of algorithm of tracing by every vertex could be overcome by this method. The method was particularly fit for correcting the atmospheric refractive error in the real time data interpretation of full-automatic film interpretoscopes or optic-electronic theodolite.

**Keywords** Correction for atmospheric refractive error; NURBS Curve; Interpolation; Algorithm of tracing by every vertex; Angle measure error



**Zhao Xiaofei** was born on July 14th, 1978, in Shaanxi. He earned the Bachelor' Degree in optical engineering from Zhejiang University, Hangzhou, China, in 1999. Now he is a Master candidate in Xi'an Institute of Optics and Precision Mechanics, Chinese Academy of Sciences. His research mainly focuses on software development and image process.