FAST EXTRACTION OF PLANES NORMAL TO THE CENTRELINE FROM CT COLONOGRAPHY DATASETS

N. Sezille, R.J.T. Sadleir and P.F. Whelan

Vision Systems Laboratory, Dublin City University, Ireland

ABSTRACT

CT Colonography (CTC) is an emerging imaging technique for colon screening. It is possible to extend this technique to automatically detect colonic anomalies. In order to improve the detection rate of the potential polyps, it is desirable to analyse plane sections orthogonal to the colon centreline rather than the axial CT slices. This paper describes a fast artefact free method for orthogonal plane extraction. The centreline in its initial form lacks the smoothness required to provide accurate local orientation of the colon. An approach based on down-sampling and B-spline interpolation generates a suitable smooth centreline. The cubic B-spline form gives low cost access to accurate local orientation and curvature of the colon. The local orientation is used as a normal vector for the orthogonal plane. The plane extraction from the anisotropic CT dataset is divided into two stages. First, a projection of the orthogonal plane on one of the dataset faces is created. A bilinear interpolation on the projected plane extracts the actual viewing window. This plane extraction is fast: extracting a plane using trilinear interpolation requires $\sim 7*N^2$ linear interpolations, our method achieves the same image quality and requires only $\sim 3*N^2$ linear interpolations. This technique can be use to analyse cross-sections of the colon without significant processing overhead.

INTRODUCTION

Colorectal cancer is a major cause of cancer related death in Ireland (10% of cancer related death), this is second only to lung cancer (19.5 %), NCRI (1). It is also the second most common cancer with 9% of cases. This cancer can be prevented by detecting and resecting precursor polyps early in their course.

Computed Tomography Colonography (CTC) is an emerging technique for mass screening. A particularly interesting aspect of this technique is its suitability for generating virtual reality models of the colon that radiologist can examine, Vining et al. (2). This can be extended to automated polyp detection by dedicated computer programs.

Traditionally both the medical specialists and the detection software have been working on the axial, sagittal and coronal images from the CTC datasets. This is partly due to the fact that the CT scanners are unable to provide oblique slices. It also gives strong references to the medical population for identifying anomalies. However the analysis of oblique planes, more particularly planes orthogonal to the colon centreline, is very likely to improve the performance of some automated polyp detection methods. As CT scanners only provide axial images, the oblique planes must be generated by computer reconstruction. The reconstruction of these planes must be fast and artefact free to be suitable for analysis. The plane extraction method we present in this paper satisfies these criteria.

Previous Work

Due to the inability of the 3D scanning devices to generate oblique slices, researchers have proposed a number of different approaches to the problem. One of the first one, introduced by Rhodes et al. (3), was very computationally intensive: the entire dataset was reformatted so that the slices would have the required orientation; all voxels outside the desired slice were discarded. This method is not suitable for our application since planes orthogonal to the colon centreline are seldomly parallel. A common approach is to directly compute the pixels for the plane using a three dimensional floating point interpolation scheme like trilinear or tricubic B-spline interpolations. Once again, as three dimensional interpolations are costly (one trilinear interpolation requires seven linear interpolations) this method is not suitable when a great number of planes are to be extracted for analysis. Kramer et al. (4) proposed an original approach using the Fourier-shift theorem but the need to go back and forth between the image and Fourier domains limits the utilisation of this approach to cases where a great number of parallel slices are to be extracted.

Finally, following the development of the discrete geometry theory, Figueiredo (5) proposed a fast and robust method developed using mathematical theory and based on integer arithmetic. Our method is a modified version of this approach, incorporating floating point calculation in order to improve the resulting data quality for analysis.

METHODS

Centreline Processing

The aim of the centreline processing stage is to generate a smoother version of the extracted centreline in a form that gives access to a reliable positioning inside the colon. It also provides reliable first derivative which is used as the local colon orientation and an approximation of the second derivative. The form chosen is a cubic B-spline interpolation of the original centreline.

In its initial form, the centreline is a relatively high frequency signal representing the central path trough the colon, generated with the method described by Sadleir and Whelan (6). At first a simple second order low pass filter is applied to remove the high frequency components. The filtered centreline is then down-sampled (typically of a factor seven). Three cubic B-spline interpolators are constructed for the resulting set of points (one for each of the coordinates). The interpolator provides exact fit. In order to improve the performance the B-spline coefficients are calculated recursively and tri-diagonal system is solved using an adapted LU decomposition method proposed by Thorson (7). The final centreline is obtained by interpolating, with the B-spline interpolators, the points discarded when down-sampled. Points between the initial points, and the derivatives at these points, are calculated by linear interpolation on the bounding centreline points.

Oblique Plane Extraction

Theory. A digital plane is a subset of the three dimensional integer space described by equation (1). The digital equivalent of the Euclidian plane is called *naïve digital plane,* it is a subset of the digital planes for which $\rho = \max(|a|,|b|,|c|)$. These planes are 18-connected and have no hole for 6-connectivity. The naïve digital plane described by equation (2) is the digitalisation by truncation of the Euclidean plane described by equation (3).

$$P(a,b,c,\gamma,\rho) = \{(x,y,z) \in Z^3 \,/\, \gamma \le ax+by+cz < \gamma + \rho\} \quad (1)$$

$$P(a,b,c,\gamma) = \left\{ \begin{array}{l} (x,y,z) \in Z^3 \,/ \\ \gamma \le ax+by+cz < \gamma + \max(|a|,|b|,|c|) \end{array} \right\} \quad (2)$$

$$ax + by + cz = \gamma \quad (3)$$

Without loss of generality, we can suppose that the main direction of the normal ***N****(a,b,c)* to *P* is *c* (i.e. *z* is function of *x* and *y*, *c=max(|a|,|b|,|c|)*). We suppose as well that *a*, *b* and *c* are positives. From the inequality in equation (2) follows the equation (4) which implements the integer division.

$$z = -\left\lceil \frac{ax + by - \gamma}{c} \right\rceil \quad (4)$$

Based on Figueiredo (5) it can be shown that we can evaluate *z(x+1,y)* in function of *z(x,y)* as follow.

$$r(x+1,y) = r(x,y) - \frac{a}{c} \quad (5)$$

if *r(x+1,y) < 0*:

$$z(x+1,y) = z(x,y) - 1 \quad (6)$$

else:

$$z(x+1,y) = z(x,y) \quad (7)$$

Implementation. The plane extraction algorithm is divided in two steps. First the plane is located and projection onto one face of the volumetric dataset is created (Fig. 1). Then, the actual viewing window is extracted from the projection using bilinear interpolation.

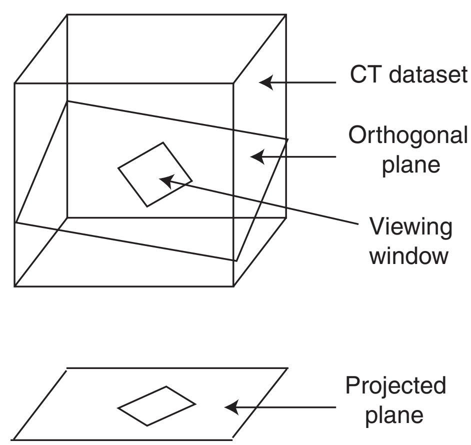


Fig. 1: Projection of the oblique plane

The plane to extract is defined by a point in the dataset and a normal vector. The point *C(ox,oy,oz)* used is a point from the processed colon centreline; it is the local centre of the colon and will be the centre of the viewing window. The normal vector ***N****(a,b,c)* used is the first derivative of the centreline at the selected point; it represents the local orientation of the colon. The parameters *ox*, *oy*, *oz*, *a*, *b* and *c* are used to calculate γ and to determine which face to project the plane on. If *a=max(|a|,|b|,|c|)* the plane is projected on the *(O,**y**,**z**)* face, if *b=max(|a|,|b|,|c|)* it is projected on the *(O,**z**,**x**)* and if *b=max(|a|,|b|,|c|)* it is projected on the *(O,**x**,**y**)* face. The oblique plane is not entirely projected; in order to save calculation time only a portion, of the projected plane, big enough to retrieve the viewing window is generated. The projected plane is centred on the projection of C, its bound are calculated for its width and height to be equal to the diagonal of the viewing window. The elevations (*z* if *c= max(|a|,|b|,|c|)*) of points to project are calculated in a very fast double loop where the only operations are one floating point division and subtraction and unity increments and decrements. This double loop is described in the following pseudo-code in the case *c=max(|a|,|b|,|c|)*. The voxel value projected is interpolated on the ***z*** axis at the elevation *z+r* where *z* is an integer and *r* a floating point value between -1 and 1. The initial value of *z* is calculated using equation (4), the initial value of *r* is the fractional part of *z* when calculated with a floating point division. The projected plane is stored in *projection*.

```
float z_double = -(a*x+b*y-γ );
int zy = floor(z_double);
float ry = z_double – zy;

for(y=miny;y<=maxy;y++)
{
    int z = zy;
    float r = ry;
    for(x=minx;x<=maxx;x++)
    {
        projection[x-minx][y-miny] =
            getVoxelValueInterpolatedOnZ(x,y,z,r);

        r -= a/c;
        if(r < 0) {
            r++;
            z--;
        }
        else if(r >= 1) {
            r--;
            z++;
        }
    }
    ry -= b/c;
    if(ry < 0) {
        ry++;
        zy--;
    }
    else if(ry >= 1) {
        ry--;
        zy++;
}
```

The second part of the oblique plane extraction is the viewing window interpolation from the projected plane. The viewing window is defined by its centre which is the projection of the point *C*, its dimensions and a unity vector ***D*** which represents the horizontal in the viewing plane. As some coherence is needed between successive extracted slices orthogonal to the centreline, the vector ***D*** is calculated using its first and second derivatives at selected point. The second vector of the viewing window, ***R***, represents the vertical in the viewing plane. It is calculated as the cross-product of ***N*** and ***D***. The vector ***D*** and ***R*** are projected on the projected plane, the resulting vectors $\mathbf{D_p}$ and $\mathbf{R_p}$ constitute the basis the referential of the viewing window pixels onto the projected plane. These vectors can be multiplied by a factor, hence allowing low cost scaling of the viewing window. The viewing window pixels are calculated one by one using a bilinear interpolation on the projected plane at the coordinates (x_p, y_p) generated the equation (8) where *d* and *r* are respectively the coordinates of the pixel in the viewing image.

$$\begin{bmatrix} x_p \\ y_p \end{bmatrix} = d * \vec{D}_p + r * \vec{R}_p \qquad (8)$$

RESULTS

The centreline processing is efficient, our implementation runs the full processing, including the evaluation of N and D for all the interpolated points in about 90 ms for a centreline of 1700 points on a standard desktop computer (java, Pentium-IV 1.5GHz/512Mb). The processed centreline is very accurate; the average distance between a processed point its corresponding point in the original centreline (6) is 0.76 voxel with a standard deviation of 0.54 (see Fig. 2).

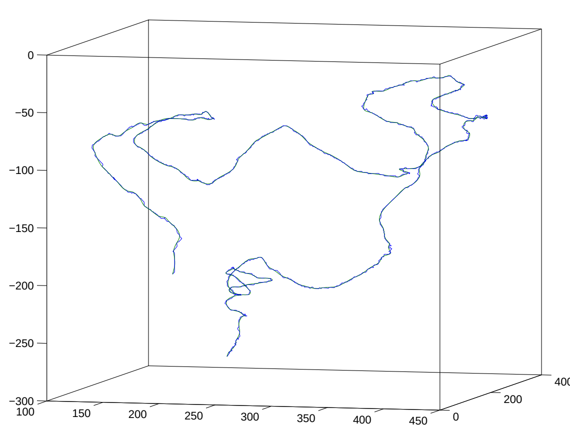
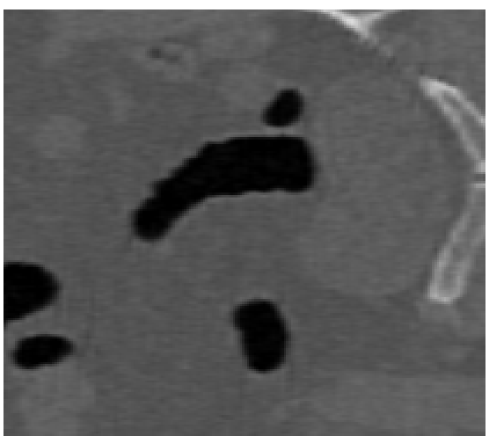
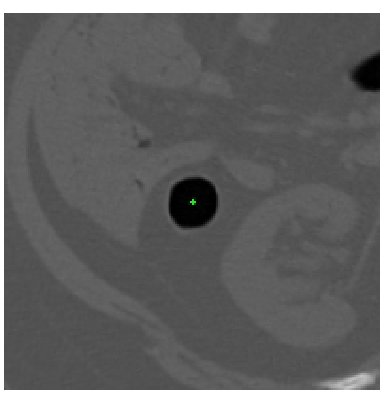


Fig. 2: Global view of the original and the processed centreline

The oblique plane extraction algorithm realises both objectives of a fast and high quality method. Depending on the orientation of the oblique plane the quality achieved is similar or better than for a coronal or sagittal image (the dataset is anisotropic), see Fig. 3a and Fig. 3b. These images are not related, they are however on the same scale. A visual inspection shows that the quality of the image in the coronal plane is much poorer, due to the lack of resolution in one of the dimensions than an arbitrary chosen oblique plane. Obviously if the orientation of the oblique plane tends toward the coronal or the sagittal planes the image quality will decrease accordingly. The algorithm is fast enough for our purpose (analysis), our implementation (java, Pentium-IV 1.5GHz, 512Mb) take an average of 55 ms (depending of how much of the viewing window is out of bounds of the dataset) to extract an 200×200 16 bits image from a 512×512×286 dataset. The time required to perform the analysis of such an image is significantly higher. This algorithm achieves the same quality than as trilinear based method whilst using only $\sim 3*N^2$ linear interpolations compared with $\sim 7*N^2$ for the trilinear interpolation to extract a N×N image.



a



b

Fig. 3: Standard interpolated coronal image (a) and custom extracted image using the documented technique (b) at the same resolution.

CONCLUSION

This paper presents a method for efficiently producing high quality images orthogonal to the centreline in a CT Colonography dataset. Both the centreline processing to make it suitable to provide reliable information on the colon orientation and the oblique plane extraction used to generate these images are discussed.

The oblique plane extraction algorithm relies on the digital geometry theory and achieves very good speed without quality loss of quality.

ACKNOWLEDGEMENTS

This project is founded by the Health Research Board (HRB). The authors would also like to thank members of the Department of Radiology and Gastrointestinal Unit at the Mater Misercordiae Hospital in Dublin.

REFERENCES

1. NCRI, 2000, “Cancer in Ireland, 1997: Incidence and Mortality”, Healy & Associates
2. Vining D.J., Gelfand D.W., Bechtold R.E., Scharling E.S., Grishaw E.K. and Shifrin R.Y., 1994, “Technical Feasibility of Colon Imaging with Helical CT and Virtual Reality”, Am. J. Roentgenology, 162, suppl:104
3. Rhodes L.M., Glenn W.V. and Azzawi Y.M., 1980, “Extracting oblique planes from serial CT sections”, J. Comp. Ass. Tomog., 4, 679-657
4. Kramer D.M., Kaufman L., Guzman R.J. and Hawryszko C., 1990, “A General Algorithm for Oblique Image Reconstruction”, IEEE C.G. & A., 10, 62-65
5. Figueiredo O., 1999, “Advances in Discrete Geometry Applied to the Extraction of Planes and Surfaces from 3D Volumes”, thesis, EPFL, Lausanne, CH
6. Sadleir R.J.T. and Whelan P.F., 2002, "Colon Centreline Calculation for CT Colonography using Optimised 3D Topological Thinning", First Int Symp. 3D Data Proc. Vis. & Trans.
7. Thorson J, 1979 “Gaussian Elimination on a Banded Matrix”, SEP, 20