3D localization of crackles

1. Identify crackles on a single channel
2. Identify sister-crackles on all channels
3. Crosscorrelate the mother-crackle with sister crackles in order to calculate the delay between the mother- and sister-crackles. See explanations of crosscorrelation in this paper: DOI: 10.1378/chest.128.3.1468 Transmission of Crackles in Patients With Interstitial Pulmonary Fibrosis, Congestive Heart Failure, and Pneumonia
4. Use this function to calculate XYZ coordinates in the chest.

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Localization\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

void CSTG1View::LocInitialize\_xyzvm(UINT16 nM)

{

UINT16 x,y,z,v,m;

// UINT16 scalar=nSinms;//points/ms

double R;

for(x=0; x<nX; x++){

for(y=0; y<nY; y++){

for(z=0; z<nZ; z++){

for(v=0; v<nV; v++){

for(m=0; m<nM; m++){

R = sqrt( pow((rX\*x+minX-Mx[m]),2.0)+pow((rY\*y+minY-My[m]),2.0)+pow((rZ\*z+minZ-Mz[m]),2.0) );// in mm

xyzvm[x][y][z][v][m] = UINT16( RoundDouble( R \* nSinms , (rV\*v+minV) ) ); //time to reach mic m from point xyz with speed V (in number of points);

}

}

}

}

}

}

void CSTG1View::LocInitialize\_xyzvm\_constV(UINT16 nM)

{

UINT16 x,y,z,m;

// UINT16 scalar=nSinms;//points/ms

double R;

for(x=0; x<nX; x++){

for(y=0; y<nY; y++){

for(z=0; z<nZ; z++){

for(m=0; m<nM; m++){

R = sqrt( pow((rX\*x+minX-Mx[m]),2.0)+pow((rY\*y+minY-My[m]),2.0)+pow((rZ\*z+minZ-Mz[m]),2.0) );// in mm

xyzvm[x][y][z][0][m] = UINT16( R \* nSinms );// mm\*point/ms->later devided by mm/ms will produce points

}

}

}

}

}

void CSTG1View::LocLocalize(short side, UINT16 MotherCh, int excludech, short \*delay, int \*sistermax, int \*xyzv)

{

UINT16 x, y, z, v, ch, xx, yy, zz, vv;

unsigned int err, minerr=UINT\_MAX;

// float err, minerr=FLT\_MAX;//AV 10/10/07, since some err are negative

short diff=0;

short ch1, ch2;

ch1ch2(&ch1, &ch2, side);

UINT16 V = xyzv[3];

if(GetDocument()->ParamDlg.bconstV){//constant V obtained from Freq Eq., use the same matrix, v index =0

for(x=0; x<nX; x++){//later: start near mother cr to locate min faster

for(y=0; y<nY; y++){

for(z=0; z<nZ; z++){

for(ch=ch1, err=0; ch<ch2; ch++){

if(ch != MotherCh && ch != excludech && (ch!=7 || b4x4) ){//&& CrSisterQuality[ch+8\*side][index]

diff= (xyzvm[x][y][z][0][ch - 8\*side] - xyzvm[x][y][z][0][MotherCh - 8\*side])/V - delay[ch];//in points, not in ms

err += sistermax[ch] \* diff \* diff;

}

}

if(err < minerr){

minerr = err;

xx=x;

yy=y;

zz=z;

}

}

}

}

xyzv[0]=xx\*rX + minX;//1. positive on the right , negative on the left; 2. convert to mm

xyzv[1]=yy\*rY + minY;

xyzv[2]=zz\*rZ + minZ;

xyzv[4]=int(minerr);

}

else{

for(x=0; x<nX; x++){//later: start near mother cr to locate min faster

for(y=0; y<nY; y++){

for(z=0; z<nZ; z++){

for(v=0; v<nV; v++){

for(ch=ch1, err=0; ch<ch2; ch++){

if(ch != MotherCh && ch != excludech && ch != 7){//&& CrSisterQuality[ch+8\*side][index]

diff= xyzvm[x][y][z][v][ch - 8\*side] - xyzvm[x][y][z][v][MotherCh - 8\*side] - delay[ch];

err += sistermax[ch] \* diff \* diff;

}

}

if(err < minerr){

minerr = err;

xx=x;

yy=y;

zz=z;

vv=v;

}

}

}

}

}

xyzv[0]=xx\*rX + minX;//1. positive on the right , negative on the left; 2. convert to mm

xyzv[1]=yy\*rY + minY;

xyzv[2]=zz\*rZ + minZ;

xyzv[3]=vv\*rV + minV;

xyzv[4]=int(minerr);

}

}

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*End of Localization\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*