I2D\*

weightedSample(F2D\* w)

{

I2D \*bin;

F2D \*seed;

int n, i, j;

n = w->height;

seed = randWrapper(n, 1);

bin = iSetArray(n, 1, 0);

for(i=0; i<n; i++)

{

for(j=0; j<n; j++)

{

if(asubsref(seed,j) > 0)

asubsref(bin,j) = asubsref(bin,j) + 1;

}

for(j=0; j<n; j++)

asubsref(seed,j) = asubsref(seed,j) - asubsref(w,i);

}

free(seed);

return bin;

}

void

generateSample(F2D \*w, F2D \*quat, F2D \*vel, F2D \*pos)

{

int rows, cols, i, j, index;

I2D \*sampleXId;

F2D \*retQuat, \*retVel, \*retPos;

sampleXId = weightedSample(w);

rows = sampleXId->height;

cols = sampleXId->width;

if(cols > 1) {

printf("ERROR: Cols more than 1\n");

return ;

}

retQuat = fSetArray(quat->height, quat->width, 0);

retVel = fSetArray(vel->height, vel->width, 0);

retPos = fSetArray(pos->height, pos->width, 0);

for(i=0; i<rows; i++)

{

index = asubsref(sampleXId, i) - 1;

for(j=0; j<quat->width; j++)

{

subsref(retQuat,i,j) = subsref(quat,index,j);

}

}

for(i=0; i<rows; i++)

{

index = asubsref(sampleXId, i) - 1;

for(j=0; j<vel->width; j++)

{

subsref(retVel,i,j) = subsref(vel,index,j);

}

}

for(i=0; i<rows; i++)

{

index = asubsref(sampleXId, i) - 1;

for(j=0; j<pos->width; j++)

{

subsref(retPos,i,j) = subsref(pos,index,j);

}

}

for(i=0; i<quat->height; i++)

{

for(j=0; j<quat->width; j++)

{

subsref(quat,i,j) = subsref(retQuat,i,j);

}

}

for(i=0; i<vel->height; i++)

{

for(j=0; j<vel->width; j++)

{

subsref(vel,i,j) = subsref(retVel,i,j);

}

}

for(i=0; i<pos->height; i++)

{

for(j=0; j<pos->width; j++)

{

subsref(pos,i,j) = subsref(retPos,i,j);

}

}

fFreeHandle(retQuat);

fFreeHandle(retVel);

fFreeHandle(retPos);

iFreeHandle(sampleXId);

return;

}