```cpp
//------------------------------------------------------------------------
//    Purpose:
//
//        Class Angles Implementation Files
//
//    Discussion:
//
//
//    Licensing:
//
//        This code is distributed under the GNU LGPL license.
//
//    Modified:
//
//        2012.05.11
//
//    Author:
//
//        Young Won Lim
//
//    Parameters:
//
//------------------------------------------------------------------------


//------------------------------------------------------------------------
//    Plot Non-uniform Quantization of CORDIC
//------------------------------------------------------------------------
void Angles::plot_quantization ()
{

  vector <double> B, D;
  vector <double> ::iterator first, last;
  double mean, std;
  ofstream myout;


  cout << "* calc_statistics... ";
  if (Leaf) cout << "(LeafAngles)" << " nAngle = " << nAngle << endl;
  else cout << "(AllAngles)" << " nAngle = " << nAngle << endl;


  for (int i=0; i < nAngle; ++i) {
    // cout << "A[" << i << "]=" << setw(12) << setprecision(8) << A[i] << endl;
    // cout << "B[" << i << "]=" << setw(12) << setprecision(8) << B[i] << endl;
  }

  // B : sorted angles array
  for (int i=0; i < nAngle; ++i)
    B.push_back(A[i]);

  sort(B.begin(), B.end());


  // D : difference angle array
  for (int i=0; i < nAngle-1; ++i)
    D.push_back(B[i+1]- B[i]);

  sort(D.begin(), D.end());


  double udelta = (B[B.size()-1] - B[0]) /  nAngle; // computed unifrom delta

  // write histogram data from delta array
  myout.open("angle.dat");
  double pb ;
  for (int i=0; i<nAngle; i++) {
```

```cpp
    myout << fixed << right << setw(12) << setprecision(7) << B[0] + delta*i ;
    myout << fixed << right << setw(12) << setprecision(7) << B[i] ;
    myout << " " << endl;
  }
  myout.close();


  cout << "  + Delta Distribution Plot \n" ;

  // writing gnuplot commands
  myout.open("command.gp");
  myout << "set terminal " << GnuTerm << endl;
  if (Leaf) {
    myout << "set output 'eg05.leaf.delta_dist.emf'" << endl;
    myout << "set title \"Delta Distribution of Leaf Angles\" " << endl;
  } else {
    myout << "set output 'eg05.all.delta_dist.emf'" << endl;
    myout << "set title \"Delta Distribution of All Angles\" " << endl;
  }
  myout << "set xlabel \"Delta (Adjacent Angle Difference)\" " << endl;
  myout << "set ylabel \"probability\" " << endl;
  myout << "set yrange [0:+1]" << endl;

  myout << "set arrow from " << avg_delta << ", 0";
  myout << " to " << avg_delta << ", 0.7" << endl;
  myout << "set label \"avg_delta \" at " << avg_delta;
  myout << ", 0.7 right" << endl;

  myout << "set arrow from " << udelta << ", 0";
  myout << " to " << udelta << ", 0.5" << endl;
  myout << "set label \"uniform delta \" at " << udelta;
  myout << ", 0.5 right" << endl;

  myout << "plot 'angle.dat' with linespoints" << endl;
  myout << "pause mouse keypress" << endl;
  myout.close();


  system("gnuplot command.gp");



  cout << "  + Angle-Delta Plot \n" ;

  // write angle-delta data
  myout.open("angle.dat");
  for (int i=0; i<B.size()-1; i++) {
    myout << B[i] << " " << B[i+1] - B[i] << endl;
  }
  myout.close();


  // writing gnuplot commands
  myout.open("command.gp");
  myout << "set terminal " << GnuTerm << endl;
  if (Leaf) {
    myout << "set output 'eg06.leaf.angle_delta.emf'" << endl;
    myout << "set title \"Angle-Delta Plot of Leaf Angles\" " << endl;
  } else {
    myout << "set output 'eg06.all.angle_delta.emf'" << endl;
    myout << "set title \"Angle-Delta Plot of All Angles\" " << endl;
  }
  myout << "set xlabel \"Angles in radian\" " << endl;
  myout << "set ylabel \"Delta (Adj Angle Diff) \" " << endl;

  myout << "set arrow from " << "-1.0, " << avg_delta;
  myout << " to " << "+1.0, " << avg_delta << endl;
```

```cpp
    myout << "set label \"avg_delta \" at " << "+0.0, ";
    myout << avg_delta << " left" << endl;

    myout << "set arrow from " << "-1.0, " << udelta;
    myout << " to " << "+1.0, " << udelta << endl;
    myout << "set label \"uniform delta \" at " << "+1.0, ";
    myout << udelta << " left" << endl;

    myout << "set arrow from " << "-0.7853, " << min_delta;
    myout << "0 to -0.7853, " << max_delta  << endl;
    myout << "set label \"-pi/4 \" at " << "-0.7853, " ;
    myout << min_delta << " right " << endl;

    myout << "set arrow from " << "+0.7853, " << min_delta;
    myout << "0 to +0.7853, " << max_delta  << endl;
    myout << "set label \"+pi/4 \" at " << "+0.7853, " ;
    myout << min_delta << " left " << endl;

    myout << "plot 'angle.dat' with linespoints" << endl;
    myout << "pause mouse keypress" << endl;
    myout.close();


    system("gnuplot command.gp");


    return;
}
```