

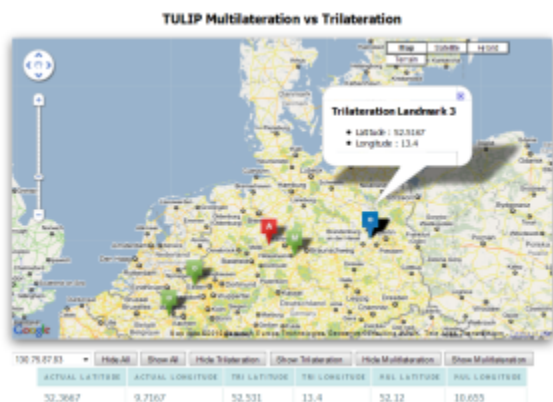
# TULIP Summary for CBG (Visualization Tool)

## Motivation

This tool is developed to assist in comparing results of the [Apollonius](#), [Multilateration](#) and [Trilateration](#) methods for Geolocation of target IP hosts that respond to pings from known landmarks (see <http://www.slac.stanford.edu/comp/net/wan-mon/tulip/>). Initially a dual map approach has been used to pinpoint errors and calculate distances between the estimated results and actual locations. We are also considering a more conventional single map approach as an alternate too. This visualization will help researchers analyze the strengths and weaknesses of both algorithms and will aid in figuring out suitable conditions for each case.


Try it!

## Approach



We have used Google Maps Api for plotting the results. The measurement results are exported by a script into a csv file. The csv file has specific format (listed below). When the page is loaded a javascript function is called to parse csv file into js variables. These variables are then used to show different information all over the page. HTML file has a combo box. This combobox is populated on the fly with ip addresses from the csv file. When a user selects an ip address, google map is refreshed to show Actual Location, Trilateration and Multilateration estimates with their landmarks used to pinpoint the region.

Three different color markers are used to identify each set of variable.

-  Actual Location
-  Trilateration Estimate
-  Trilateration Landmark
-  Multilateration Estimate
-  Multilateration Landmark

## Flow



## Implementation

### TriVsMul.JavaScript

```

var gmap;
var markersArray = []; // Contains All markers

var markersArrayTri = [];
var markersArrayMul = [];

function initialize() {
}

function getFile(pURL,pFunc) {
    if (window.XMLHttpRequest) { // code for Mozilla, Safari, etc
        xmlhttp=new XMLHttpRequest();
        eval('xmlhttp.onreadystatechange='+pFunc+'');
        xmlhttp.open("GET", pURL, true); // leave true for Gecko
        xmlhttp.send(null);
    } else if (window.ActiveXObject) { //IE
        xmlhttp=new ActiveXObject('Microsoft.XMLHTTP');
        if (xmlhttp) {
            eval('xmlhttp.onreadystatechange='+pFunc+'');
            xmlhttp.open('GET', pURL, false);
            xmlhttp.send();
        }
    }
}

function makeTable() {
    if (xmlhttp.readyState==4) {
        if (xmlhttp.status==200) {
            var tmpArr=xmlhttp.responseText.split('\n');
            var out='';
            var tmp;
            var val;
            var ips;

            var ip; ; var act_lat ; var act_long ; var tri_est_lat ; var tri_est_long ;
            var tri_land_lat1 ; var tri_land_rtt1 ; var tri_land_lat2 ; var tri_land_rtt2 ; var tri_land_lat3;

```

```

var tri_land_rtt3; var tri_error; var mul_est_long ; var mul_land_lat1; var mul_land_rtt1; var mul_land_lat2 ;
var mul_land_rtt2; var mul_land_lat3 ; var mul_land_rtt3 ; var mul_land_lat4 ; var mul_land_rtt4; var
mul_land_lat4;
var mul_land_rtt4; var mul_error;

    for (var idx=0;idx<tmpArr.length;idx++) {
        tmp=tmpArr[idx].split(',');

        ip = tmp[0];
        act_lat = tmp[1];
        act_long = tmp[2];

        tri_est_lat = tmp[3];
        tri_est_long = tmp[4];

        tri_land_lat1 = tmp[5];
        tri_land_long1 = tmp[6];
        tri_land_rtt1 = tmp[7];

        tri_land_lat2 = tmp[8];
        tri_land_long2 = tmp[9];
        tri_land_rtt2 = tmp[10];

        tri_land_lat3 = tmp[11];
        tri_land_long3 = tmp[12];
        tri_land_rtt3 = tmp[13];

        tri_error = tmp[14];

        mul_est_lat = tmp[15];
        mul_est_long = tmp[16];

        mul_land_lat1 = tmp[17];
        mul_land_long1 = tmp[18];
        mul_land_rtt1 = tmp[19];

        mul_land_lat2 = tmp[20];
        mul_land_long2 = tmp[21];
        mul_land_rtt2 = tmp[22];

        mul_land_lat3 = tmp[23];
        mul_land_long3 = tmp[24];
        mul_land_rtt3 = tmp[25];

        mul_land_lat4 = tmp[26];
        mul_land_long4 = tmp[27];
        mul_land_rtt4 = tmp[28];

        mul_land_lat5 = tmp[29];
        mul_land_long5 = tmp[30];
        mul_land_rtt5 = tmp[31];

        mul_error = tmp[32];

    out +=

        "<table><tr>" +
        "Ip: " + ip+ "<br>" +
        "act_lat: " + act_lat + "<br>" +
        "act_long: " + act_long + "<br>" +
        "tri_est_lat: " + tri_est_lat + "<br>" +
        "tri_est_long: " + tri_est_long + "<br>" +
        "tri_land_lat1: " + tri_land_lat1 + "<br>" +
        "tri_land_long1: " + tri_land_long1 + "<br>" +
        "tri_land_rtt1: " + tri_land_rtt1 + "<br>" +
        "tri_land_lat2: " + tri_land_lat2 + "<br>" +
        "tri_land_long2: " + tri_land_long2 + "<br>" +
        "tri_land_rtt2: " + tri_land_rtt2 + "<br>" +
        "tri_land_lat3: " + tri_land_lat3 + "<br>" +
        "tri_land_long3: " + tri_land_long3 + "<br>" +
        "tri_land_rtt3: " + tri_land_rtt3 + "<br>" +

        "tri_error: " + tri_error + "<br>" +

```

```

        "mul_est_lat: "+ mul_est_lat + "<br>" +
        "mul_est_long: "+ mul_est_long + "<br>" +

        "mul_land_lat1: "+ mul_land_lat1 + "<br>" +
        "mul_land_long1: "+ mul_land_long1 + "<br>" +
        "mul_land_rtt1: "+ mul_land_rtt1 + "<br>" +

        "mul_land_lat2: "+ mul_land_lat2 + "<br>" +
        "mul_land_long2: "+ mul_land_long2 + "<br>" +
        "mul_land_rtt2: "+ mul_land_rtt2 + "<br>" +

        "mul_land_lat3: "+ mul_land_lat3 + "<br>" +
        "mul_land_long3: "+ mul_land_long3 + "<br>" +
        "mul_land_rtt3: "+ mul_land_rtt3 + "<br>" +

        "mul_land_lat4: "+ mul_land_lat4 + "<br>" +
        "mul_land_long4: "+ mul_land_long4 + "<br>" +
        "mul_land_rtt4: "+ mul_land_rtt4 + "<br>" +

        "mul_land_lat5: "+ mul_land_lat5 + "<br>" +
        "mul_land_long5: "+ mul_land_long5 + "<br>" +
        "mul_land_rtt5: "+ mul_land_rtt5 + "<br>" +

        "mul_error: "+ mul_error

        + '<br><br></tr>';

        ips += "<OPTION value= " + act_lat + "," + act_long + "," + tri_est_lat +
        "," + tri_est_long + "," + tri_land_lat1 + "," + tri_land_long1 + "," + tri_land_lat2 + "," + tri_land_long2 +
        "," + tri_land_lat3 + "," + tri_land_long3 + "," + mul_est_lat + "," + mul_est_long + "," + mul_land_lat1 + ","
        + mul_land_long1 + "," + mul_land_lat2 + "," + mul_land_long2 + "," + mul_land_lat3 + "," + mul_land_long3 +
        "," +
        + mul_land_lat4 + "," + mul_land_long4 + "," + mul_land_lat5 + "," + mul_land_long5 + ">" + ip + "</option> ";

    }
    out += '</table>';
    document.getElementById('Example').innerHTML=out;

    document.getElementById('Node').innerHTML=ips;

}
}

function createMarker(point,html)
{
    var marker = new GMarker(point);
    GEvent.addListener(marker, "click", function() {
        marker.openInfoWindowHtml(html);});
    return marker;
}

function showMarkers() {

    deleteOverlays(markersArray);
    deleteOverlays(markersArrayTri);
    deleteOverlays(markersArrayMul);

    var myLatLng = new google.maps.LatLng(-25.363882,131.044922);
    var myOptions = {
        zoom: 5,
        center: myLatLng,
        mapTypeId: google.maps.MapTypeId.ROADMAP
    }
}

```

```

var map = new google.maps.Map(document.getElementById("map"), myOptions);

var act_lat = parseFloat(document.getElementById("Node").value.split(",")[0]);
var act_long = parseFloat(document.getElementById("Node").value.split(",")[1]);
var tri_est_lat = parseFloat(document.getElementById("Node").value.split(",")[2]);
var tri_est_long = parseFloat(document.getElementById("Node").value.split(",")[3]);

var tri_land_lat1 = parseFloat(document.getElementById("Node").value.split(",")[4]);
var tri_land_long1 = parseFloat(document.getElementById("Node").value.split(",")[5]);
var tri_land_lat2 = parseFloat(document.getElementById("Node").value.split(",")[6]);
var tri_land_long2 = parseFloat(document.getElementById("Node").value.split(",")[7]);
var tri_land_lat3 = parseFloat(document.getElementById("Node").value.split(",")[8]);
var tri_land_long3 = parseFloat(document.getElementById("Node").value.split(",")[9]);

var mul_est_lat = parseFloat(document.getElementById("Node").value.split(",")[10]);
var mul_est_long = parseFloat(document.getElementById("Node").value.split(",")[11]);

var mul_land_lat1 = parseFloat(document.getElementById("Node").value.split(",")[12]);
var mul_land_long1 = parseFloat(document.getElementById("Node").value.split(",")[13]);
var mul_land_lat2 = parseFloat(document.getElementById("Node").value.split(",")[14]);
var mul_land_long2 = parseFloat(document.getElementById("Node").value.split(",")[15]);
var mul_land_lat3 = parseFloat(document.getElementById("Node").value.split(",")[16]);
var mul_land_long3 = parseFloat(document.getElementById("Node").value.split(",")[17]);
var mul_land_lat4 = parseFloat(document.getElementById("Node").value.split(",")[18]);
var mul_land_long4 = parseFloat(document.getElementById("Node").value.split(",")[19]);
var mul_land_lat5 = parseFloat(document.getElementById("Node").value.split(",")[20]);
var mul_land_long5 = parseFloat(document.getElementById("Node").value.split(",")[21]);

//      var zoomaus = parseInt(document.getElementById("Node").value.split(",")[8]);

var myLatlngA = new google.maps.LatLng(act_lat,act_long);
var myLatlngT = new google.maps.LatLng(tri_est_lat,tri_est_long);
var myLatlngM = new google.maps.LatLng(mul_est_lat,mul_est_long);

var myLatlngTL1 = new google.maps.LatLng(tri_land_lat1,tri_land_long1);
var myLatlngTL2 = new google.maps.LatLng(tri_land_lat2,tri_land_long2);
var myLatlngTL3 = new google.maps.LatLng(tri_land_lat3,tri_land_long3);

var myLatlngML1 = new google.maps.LatLng(mul_land_lat1,mul_land_long1);
var myLatlngML2 = new google.maps.LatLng(mul_land_lat2,mul_land_long2);
var myLatlngML3 = new google.maps.LatLng(mul_land_lat3,mul_land_long3);
var myLatlngML4 = new google.maps.LatLng(mul_land_lat4,mul_land_long4);
var myLatlngML5 = new google.maps.LatLng(mul_land_lat5,mul_land_long5);

var infowindowA = setInfoWindow(setPopupText("Actual Location",act_lat,act_long),myLatlngA);
var infowindowT = setInfoWindow(setPopupText("Trilateration Estimate",tri_est_lat,tri_est_long),myLatlngT);
var infowindowM = setInfoWindow(setPopupText("Multilateration Estimate",mul_est_lat,mul_est_long),myLatlngM);

var infowindowTL1 = setInfoWindow(setPopupText("Trilateration Landmark 1",tri_land_lat1,tri_land_long1),
myLatlngTL1);
var infowindowTL2 = setInfoWindow(setPopupText("Trilateration Landmark 2",tri_land_lat2,tri_land_long2),
myLatlngTL2);
var infowindowTL3 = setInfoWindow(setPopupText("Trilateration Landmark 3",tri_land_lat3,tri_land_long3),
myLatlngTL3);

var infowindowML1 = setInfoWindow(setPopupText("Multilateration Landmark 1",mul_land_lat1,mul_land_long1),
myLatlngML1);
var infowindowML2 = setInfoWindow(setPopupText("Multilateration Landmark 2",mul_land_lat2,mul_land_long2),
myLatlngML2);
var infowindowML3 = setInfoWindow(setPopupText("Multilateration Landmark 3",mul_land_lat3,mul_land_long3),
myLatlngML3);
var infowindowML4 = setInfoWindow(setPopupText("Multilateration Landmark 4",mul_land_lat4,mul_land_long4),
myLatlngML4);
var infowindowML5 = setInfoWindow(setPopupText("Multilateration Landmark 5",mul_land_lat5,mul_land_long5),
myLatlngML5);

var table = "<table align='center'><th>Actual Latitude</th><th>Actual Longitude</th><th>Tri Latitude</th>";

```

```

<th>Tri Longitude</th><th>Mul Latitude</th><th>Mul Longitude</th><tr><td>"+ act_lat + "</td><td>"+
act_long + "</td><td>"+ tri_est_lat + "</td>      <td>"+ tri_est_long + "</td><td>"+ mul_est_lat + "</td>
<td>"+ mul_est_long + "</td></tr></table>";

```

```

document.getElementById('test').innerHTML=table;

```

```

    var TriImage = new google.maps.MarkerImage('tri.png',
        new google.maps.Size(50,50),
        new google.maps.Point(0,0),
        new google.maps.Point(50,50)
    );

```

```

    var MulImage = new google.maps.MarkerImage('mul.png',
        new google.maps.Size(50,50),
        new google.maps.Point(0,0),
        new google.maps.Point(50,50)
    );

```

```

    var ActImage = new google.maps.MarkerImage('actual.png',
        new google.maps.Size(50,50),
        new google.maps.Point(0,0),
        new google.maps.Point(50,50)
    );

```

```

    var TriLandImage = new google.maps.MarkerImage('triL.png',
        new google.maps.Size(50,50),
        new google.maps.Point(0,0),
        new google.maps.Point(50,50)
    );

```

```

    var MulLandImage = new google.maps.MarkerImage('mulL.png',
        new google.maps.Size(50,50),
        new google.maps.Point(0,0),
        new google.maps.Point(50,50)
    );

```

```

    var parkingShadow = new google.maps.MarkerImage('parking_shadow.png',
        new google.maps.Size(70,50),
        new google.maps.Point(0,0),
        new google.maps.Point(60, 50)
    );

```

```

    var markerTL1 = createMarker(myLatlngTL1,TriLandImage,parkingShadow,infowindowTL1,map);
    var markerTL2 = createMarker(myLatlngTL2,TriLandImage,parkingShadow,infowindowTL2,map);
    var markerTL3 = createMarker(myLatlngTL3,TriLandImage,parkingShadow,infowindowTL3,map);

```

```

    var markerML1 = createMarker(myLatlngML1,MulLandImage,parkingShadow,infowindowML1,map);
    var markerML2 = createMarker(myLatlngML2,MulLandImage,parkingShadow,infowindowML2,map);
    var markerML3 = createMarker(myLatlngML3,MulLandImage,parkingShadow,infowindowML3,map);
    var markerML4 = createMarker(myLatlngML4,MulLandImage,parkingShadow,infowindowML4,map);
    var markerML5 = createMarker(myLatlngML5,MulLandImage,parkingShadow,infowindowML5,map);

```

```

    var markerT = createMarker(myLatlngT,TriImage,parkingShadow,infowindowT,map);
    var markerM = createMarker(myLatlngM,MulImage,parkingShadow,infowindowM,map);

```

```

    var markerA = createMarker(myLatlngA,ActImage,parkingShadow,infowindowA,map);

```

```

    // Pushing into Marker Array

```

```

    markersArray.push(markerTL1);
    markersArray.push(markerTL2);
    markersArray.push(markerTL3);
    markersArray.push(markerML1);
    markersArray.push(markerML2);
    markersArray.push(markerML3);
    markersArray.push(markerML4);
    markersArray.push(markerML5);

```

```

markersArray.push(markerT);
markersArray.push(markerM);
markersArray.push(markerA);

// Pushing into TriMarker Array

markersArrayTri.push(markerTL1);
markersArrayTri.push(markerTL2);
markersArrayTri.push(markerTL3);
    markersArrayTri.push(markerT);

// Pushing into MulMarker Array

markersArrayMul.push(markerML1);
markersArrayMul.push(markerML2);
markersArrayMul.push(markerML3);
markersArrayMul.push(markerML4);
markersArrayMul.push(markerML5);
markersArrayMul.push(markerM);

map.setCenter(myLatlngA);

    gmap=map;
};

function createMarker(myLatlng,image,shadow, infowindow,map) {

    var marker = new google.maps.Marker({
        position: myLatlng,
        map: map,
        icon: image,
        shadow: shadow
    });

    google.maps.event.addListener(marker, 'click', function() {
        infowindow.open(map,marker);
    });

    return marker;
}

function setInfoWindow(contentString,myLatlng) {

    var infowindow = new google.maps.InfoWindow({
        content: contentString,
        position:myLatlng
    });

    return infowindow;
}

function setPopupText(title,lat,lng) {

    var contentString = '<h3>' + title + '</h3><ul><li>Latitude : ' + lat + '</li>'+
        '<li>Longitude : ' + lng+
        '</li></ul>';
    return contentString;
}

// Removes the overlays from the map, but keeps them in the array
function clearOverlays(markersArray) {
    if (markersArray) {
        for (i in markersArray) {
            markersArray[i].setMap(null);
        }
    }
}

```

```
// Shows any overlays currently in the array
function showOverlays(markersArray) {
  if (markersArray) {
    for (i in markersArray) {
      markersArray[i].setMap(gmap);
    }
  }
}

// Deletes all markers in the array by removing references to them
function deleteOverlays(markersArray) {
  if (markersArray) {
    for (i in markersArray) {
      markersArray[i].setMap(null);
    }
    markersArray.length = 0;
  }
}
```