کارهایی که برای انجام پروژه Osmium انجام دادیم

امیر رضا ویشته ۹۹۵۲۱۳۸۸
محمدحسین رحیمی ۹۹۵۲۱۲۹۸

درس شبکه های تلفن همراه
دانشگاه علم و صنعت ایران

۱۶ تیر ۱۴۰۳

فهرست مطالب

۱.۰ راهکار اول

در راهکار اول ما با استفاده از توان سیگنال‌های دریافتی (RSSI) و اطلاعات مکانی (GPS) برای تخمین مکان سلول‌های شبکه تلفن همراه گام برداشتیم. در این راهکار، برنامه اندروید سیگنال‌های دریافتی را جمع‌آوری می‌کند و با استفاده از الگوریتم‌های مختلف مکان‌یابی، مکان سلول‌های شبکه را تخمین می‌زند. مراحل:

۱. دریافت permission ها: برنامه مجوزهای لازم برای دسترسی به سیگنال‌های تلفن همراه و مکان دستگاه را درخواست می‌کند.

۲. جمع‌آوری اطلاعات سلولی: برنامه اطلاعات سلولی از جمله قدرت سیگنال و اطلاعات شناسایی سلول را جمع‌آوری می‌کند.

- دریافت اطلاعات سیگنال از سلول‌های اطراف از جمله قدرت سیگنال .(dBm) - جمع‌آوری شناسه‌های سلول (Cell IDs) و سایر اطلاعات مرتبط مانند Area (Location LAC Code) و Id Cell

- GPS - دریافت اطلاعات مکانی شامل طول جغرافیایی (Longitude) و عرض جغرافیایی (Latitude).

۳. محاسبه قدرت سیگنال: سیگنال‌های دریافتی از سلول‌های مختلف تجزیه و تحلیل می‌شوند تا قدرت سیگنال هر سلول مشخص شود. - قدرت سیگنال دریافتی با فاصله از منبع سیگنال رابطه معکوس دارد. - از مدل‌های انتشار سیگنال (مثل مدل log-distance) استفاده می‌شود تا فاصله تقریبی از هر سلول را بر اساس RSSI تخمین بزنیم.

۴. ارسال پیامک و دریافت پیامک: برنامه پیامک ارسال و دریافت می‌کند.

```
class MainActivity : AppCompatActivity() {
    private lateinit var telephonyManager: TelephonyManager
    @RequiresApi(Build.VERSION_CODES.P)
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_main)
        telephonyManager = getSystemService(TELEPHONY_SERVICE) as TelephonyManager
        if(ActivityCompat.checkSelfPermission(this, Manifest.permission.RECEIVE_SMS) != PackageManager.PERMISSION_GRANTED) {
            ActivityCompat.requestPermissions(this, arrayOf(Manifest.permission.RECEIVE_SMS, Manifest.permission.SEND_SMS), 111)
        }
        else{
            recieve_msg()
        }
        if (ContextCompat.checkSelfPermission(this, Manifest.permission.ACCESS_FINE_LOCATION) != PackageManager.PERMISSION_GRANTED ||
            ContextCompat.checkSelfPermission(this, Manifest.permission.READ_PHONE_STATE) != PackageManager.PERMISSION_GRANTED) {
            ActivityCompat.requestPermissions(this, arrayOf(Manifest.permission.ACCESS_FINE_LOCATION, Manifest.permission.READ_PHONE_STATE), 1)
        } else {
            getServingCellInfo()
        }

        button.setOnClickListener {
            var sms = SmsManager.getDefault()
            sms.sendTextMessage(editTextPhone.text.toString(), "ME", editTextTextMultiLine.text.toString(), null, null)

        }
        Log.d("number1", "hey")
        button2.setOnClickListener {
            Log.d("n", "hey")
            if (ContextCompat.checkSelfPermission(this, Manifest.permission.READ_PHONE_STATE) == PackageManager.PERMISSION_GRANTED) {
                Log.d("n", "here")
                val telephonyManager = getSystemService(TELEPHONY_SERVICE) as TelephonyManager
```

```
private fun getServingCellInfo() {
    // Check permission before accessing telephony services
    if (ContextCompat.checkSelfPermission(this, Manifest.permission.ACCESS_FINE_LOCATION) == PackageManager.PERMISSION_GRANTED &&
        ContextCompat.checkSelfPermission(this, Manifest.permission.READ_PHONE_STATE) == PackageManager.PERMISSION_GRANTED) {

        // Get all cell info
        val cellInfoList = telephonyManager.allCellInfo

        if (cellInfoList != null && cellInfoList.isNotEmpty()) {
            // Iterate through the cell info list and display the serving cell details
            for (cellInfo in cellInfoList) {
                if (cellInfo.isRegistered) {
                    val cellInfoDetails = when (cellInfo) {
                        is CellInfoGsm -> {
                            val cellIdentity = cellInfo.cellIdentity
                            val cellSignalStrength = cellInfo.cellSignalStrength
                            "GSM Cell\n" +
                                    "CID: ${cellIdentity.cid}\n" +
                                    "LAC: ${cellIdentity.lac}\n" +
                                    "Signal Strength: ${cellSignalStrength.dbm} dBm\n" +
                                    "ASU: ${cellSignalStrength.asuLevel}\n"
                        }
                        is CellInfoLte -> {
                            val cellIdentity = cellInfo.cellIdentity
                            val cellSignalStrength = cellInfo.cellSignalStrength
                            "LTE Cell\n" +
                                    "CID: ${cellIdentity.ci}\n" +
                                    "PCI: ${cellIdentity.pci}\n" +
                                    "TAC: ${cellIdentity.tac}\n" +
                                    "Signal Strength: ${cellSignalStrength.dbm} dBm\n"
                                      "RSRP: ${cellSignalStrength.rsrp} dBm\n" +
                                      "RSRQ: ${cellSignalStrength.rsrq} dB\n"
                        }
                        is CellInfoWcdma -> {
                            val cellIdentity = cellInfo.cellIdentity
                            val cellSignalStrength = cellInfo.cellSignalStrength
                            "WCDMA Cell\n" +
                                    "CID: ${cellIdentity.cid}\n" +
                                    "LAC: ${cellIdentity.lac}\n" +
                                    "Signal Strength: ${cellSignalStrength.dbm} dBm\n" +
                                    "ASU: ${cellSignalStrength.asuLevel}\n"
                        }
                        else -> "Unknown Cell Type\n"
                    }
                    editTextTextMultiLine2.setText(cellInfoDetails)
                    break // Only show the serving cell information
                }
            }
        } else {
            editTextTextMultiLine2.setText("No cell information available")
        }
    } else {
        editTextTextMultiLine2.setText("Permission not granted")
    }
}


override fun onRequestPermissionsResult(
    requestCode: Int,
    permissions: Array<out String>,
    grantResults: IntArray
) {
    super.onRequestPermissionsResult(requestCode, permissions, grantResults)
    if(requestCode == 111 && grantResults[0] == PackageManager.PERMISSION_GRANTED)
    {
        recieve_msg()
    }
```

```
        if(requestCode == 111 && grantResults[0] == PackageManager.PERMISSION_GRANTED)
        {
            recieve_msg()
        }
        if (requestCode == 1) {
            if (grantResults.isNotEmpty() && grantResults[0] == PackageManager.PERMISSION_GRANTED && grantResults[1] == PackageManager.PERMISSION_GRANTED)
                getServingCellInfo()
            } else {
                textView.text = "Permission not granted"
            }
        }
//        if(requestCode == 112 && grantResults[0] == PackageManager.PERMISSION_GRANTED)
//        {
//            registerSignalStrengthListener()
//        }
    }
    private fun recieve_msg(){
        var br = object: BroadcastReceiver(){
            override fun onReceive(p0: Context?, p1: Intent?) {
                if(Build.VERSION.SDK_INT >= Build.VERSION_CODES.KITKAT) {
                    for(sms in Telephony.Sms.Intents.getMessagesFromIntent(p1)) {
                        editTextPhone.setText(sms.originatingAddress)
                        editTextTextMultiLine.setText(sms.displayMessageBody)
                    }
                }
            }
        }
        registerReceiver(br, IntentFilter("android.provider.Telephony.SMS_RECEIVED"))
        val deliveryIntentFilter = IntentFilter("android.provider.Telephony.SMS_DELIVERED")
        val deliveryReceiver = SmsDeliveryReceiver()
        registerReceiver(deliveryReceiver, deliveryIntentFilter)
    }
}
```

۲.۰ راهکار دوم

راهکار دوم ما شامل استفاده از الگوریتم پهلوبندی دایره‌ای (Circular Lateration) برای تخمین موقعیت هدف است. این روش با استفاده از موقعیت‌ها و فاصله‌های اندازه‌گیری شده بین نقاط مختلف، مکان دقیق هدف را محاسبه می‌کند. مراحل:
۱. ایجاد ماتریس‌ها: ماتریس‌های لازم برای انجام محاسبات پهلوبندی دایره‌ای را ایجاد می‌کنیم

- از ماتریس‌های زیر برای حل معادلات استفاده می‌شود:

$$A = \begin{bmatrix} x_1 & y_1 & -0.5 \\ x_2 & y_2 & -0.5 \\ \vdots & \vdots & \vdots \\ x_n & y_n & -0.5 \end{bmatrix}$$

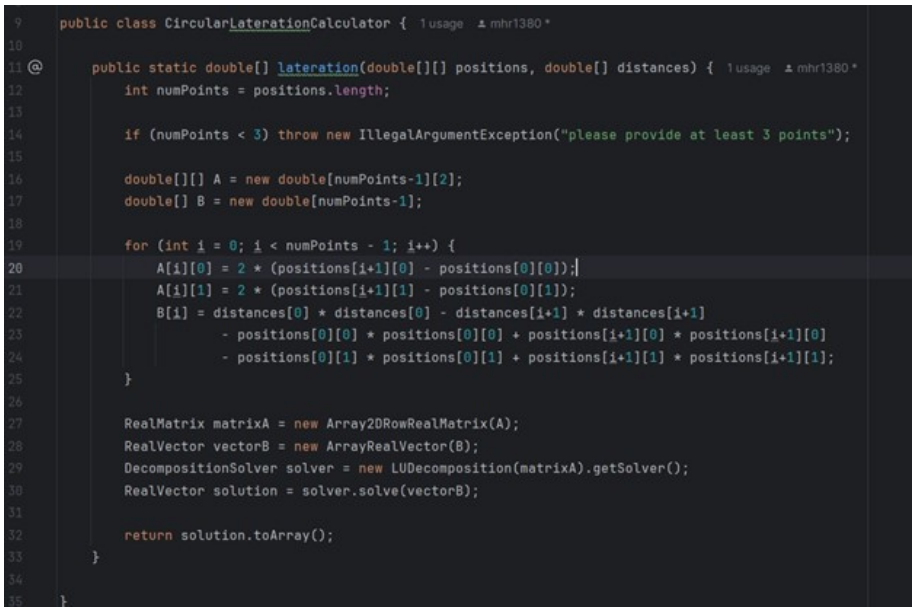
$$b = \begin{bmatrix} d_1^2 - x_1^2 - y_1^2 \\ d_2^2 - x_2^2 - y_2^2 \\ \vdots \\ d_n^2 - x_n^2 - y_n^2 \end{bmatrix}$$

۲. محاسبه موقعیت: با استفاده از ماتریس‌ها و حل معادلات خطی، موقعیت دقیق هدف محاسبه می‌کنیم. - با استفاده از روش‌های مختلف حل معادلات خطی (مانند روش حداقل مربعات)، موقعیت تخمین زده شده را پیدا می‌کنیم:

$$A = \begin{bmatrix} x_1 & y_1 & -0.5 \\ x_2 & y_2 & -0.5 \\ \vdots & \vdots & \vdots \\ x_n & y_n & -0.5 \end{bmatrix}$$

$$b = \begin{bmatrix} d_1^2 - x_1^2 - y_1^2 \\ d_2^2 - x_2^2 - y_2^2 \\ \vdots \\ d_n^2 - x_n^2 - y_n^2 \end{bmatrix}$$

$$\begin{bmatrix} x \\ y \\ c \end{bmatrix} = (A^T A)^{-1} A^T b$$



```
public class CircularLaterationCalculator {  1 usage   mhr1380 *

    public static double[] lateration(double[][] positions, double[] distances) {  1 usage   mhr1380 *
        int numPoints = positions.length;

        if (numPoints < 3) throw new IllegalArgumentException("please provide at least 3 points");

        double[][] A = new double[numPoints-1][2];
        double[] B = new double[numPoints-1];

        for (int i = 0; i < numPoints - 1; i++) {
            A[i][0] = 2 * (positions[i+1][0] - positions[0][0]);
            A[i][1] = 2 * (positions[i+1][1] - positions[0][1]);
            B[i] = distances[0] * distances[0] - distances[i+1] * distances[i+1]
                    - positions[0][0] * positions[0][0] + positions[i+1][0] * positions[i+1][0]
                    - positions[0][1] * positions[0][1] + positions[i+1][1] * positions[i+1][1];
        }

        RealMatrix matrixA = new Array2DRowRealMatrix(A);
        RealVector vectorB = new ArrayRealVector(B);
        DecompositionSolver solver = new LUDecomposition(matrixA).getSolver();
        RealVector solution = solver.solve(vectorB);

        return solution.toArray();
    }

}
```

تابع lateration به منظور تخمین موقعیت یک نقطه هدف در یک سیستم مختصات با استفاده از موقعیت‌ها و فاصله‌های اندازه‌گیری شده از چندین گره راهنما طراحی شده است.

ورودی‌ها: ۱. positions: - یک آرایه دو بعدی از نوع double که مختصات گره‌های راهنما را نشان می‌دهد. - هر عنصر در این آرایه یک آرایه دو عنصری است که مختصات x و y یک گره راهنما را مشخص می‌کند.

۲. distances: - یک آرایه از نوع double که فاصله‌های گره‌های راهنما تا گره هدف را نشان می‌دهد. - هر عنصر در این آرایه یک فاصله است که مربوط به گره راهنمای متناظر در آرایه ‘positions’ است. - مثلاً distances[i] فاصله‌ی گره راهنمای i تا گره هدف است.

خروجی: - یک آرایه از نوع double که شامل مختصات تخمینی گره هدف است. - این آرایه دو عنصر دارد که عنصر اول مختصات ‘x’ و عنصر دوم مختصات ‘y’ گره هدف را نشان می‌دهد.

عملکرد:

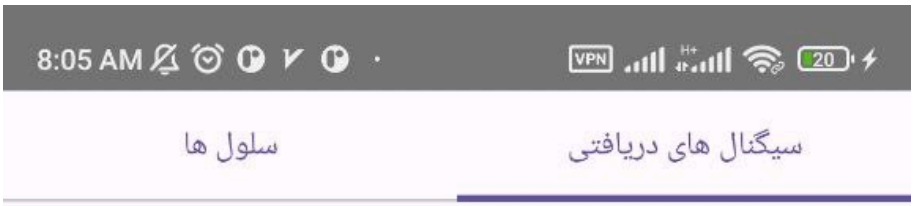
۱. ساخت ماتریس‌ها: - یک ماتریس A و یک بردار B ساخته می‌شوند که برای حل سیستم معادلات خطی لازم هستند.(قبلا به فرم انها اشاره شد.) - ماتریس A با استفاده از تفاوت مختصات گره‌ها و بردار B با استفاده از فاصله‌ها و مختصات محاسبه می‌شود.

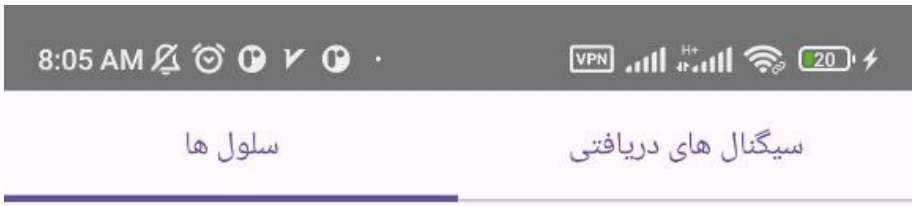
۲. حل سیستم معادلات: - با استفاده از ماتریس A و بردار B، سیستم معادلات خطی Ax = B حل می‌شود تا مختصات گره هدف تخمین زده شود.

۳.۰ باقی بخش ها

توابعی که در بالا توضیح داده شدند در پوشه utils قرار دارند. کد اصلی برنامه در Main Activity توسعه داده شده است. ما ابتدا دسترسی های لازم را از کاربر میگیریم. سپس در ادامه برای دسترسی به اطلاعات سلول های اطراف از TelephonyManager استفاده کردیم و با استفاده از آن توان سیگنال و دیگر اطلاعات را بدست آوردیم و با استفاده از توابع توسعه

داده شده و با استفاده از پهلو بندی دایره ای و دیگر محاسبات اطلاعات مورد نیاز را بدست آوردیم. این کار را به طور مداوم انجام میدهیم تا اطلاعات را به صورت Real Time داشته باشیم. برنامه ما دو تب دارد که در هر کدام اطلاعات مربوطه نمایش داده میشوند.


8:05 AM
VPN
20
سیگنال های دریافتی
سلول ها


8:05 AM
VPN
20
سیگنال های دریافتی
سلول ها

در صفحه سیگنال های دریافتی، با توجه به اطلاعات دریافت شده از سلول ها، آیدی هر سلول و توان آن را نمایش میدهیم. در هنگام نوشتن این گزارش، گوشی من به باگی در پرمیشن ها خورد (مطمئن نیستم ولی اینطور فکر میکنم چون لاگ های دریافتی خالی بودند)و دسترسی ها به درستی داده نمیشد و نتوانستم عکسی از اطلاعاتی که نمایش میدهیم بگذارم اما در اولین فرصت براتون ارسال میکنم.
در صفحه سلول ها، ما موقعیت سلول هایی که محاسبات لازم را برای آن ها انجام دادیم نمایش میدهیم.