

基于Qt的小目标探测声纳显控软件设计

何睿¹, 王彦鹏^{1*}, 冯其瑞¹, 王怀锋¹, 颜波²

¹中海油安全技术服务有限公司, 天津

²中海油田服务股份有限公司, 北京

Email: herui@cnooc.com.cn, *wangyp40@cnooc.com.cn

收稿日期: 2020年11月27日; 录用日期: 2020年12月22日; 发布日期: 2020年12月31日

摘要

声纳显控软件是声纳系统重要的组成部分。本文介绍了一种以Qt组件为基础的小目标探测声纳显控软件的设计与实现。对软件的构架、数据传输、图像显示及处理方法等进行了阐述, 并采用静态目标抑制方法有效降低虚警率, 能较好地满足用户需求。为后续小目标探测声纳的显示控制系统提供一种有效的参考。

关键词

小目标探测声纳, Qt, 图像处理, 静态目标抑制

Display and Control Software Design of Small Target Detection Sonar Based on Qt

Rui He¹, Yanpeng Wang^{1*}, Qirui Feng¹, Huifeng Wang¹, Bo Yan²

¹CNOOC Safety Technology Service Co., Ltd., Tianjin

²CNOOC Oilfield Services Co., Ltd., Beijing

Email: herui@cnooc.com.cn, *wangyp40@cnooc.com.cn

Received: Nov. 27th, 2020; accepted: Dec. 22nd, 2020; published: Dec. 31st, 2020

Abstract

Sonar display and control software is an important part of sonar system. This paper introduces the design and implementation of the small target detection sonar display and control software

*通讯作者。

based on Qt. The software architecture, data transmission, image display and processing methods are described, and static target suppression method is used to effectively reduce the false alarm rate, which can better meet the needs of users. It provides an effective reference for subsequent display control system of small target detection sonar.

Keywords

Small Target Detection Sonar, Qt, Image Processing, Static Target Suppression

Copyright © 2020 by author(s) and Hans Publishers Inc.

This work is licensed under the Creative Commons Attribution International License (CC BY 4.0).

<http://creativecommons.org/licenses/by/4.0/>



Open Access

1. 引言

随着科学技术的发展, 小型目标对于高价值海洋设施的威胁越来越大。小目标探测声纳是水下预警体系重要的一环。一款人机界面友好、人机交互方便、信息显示直观、功能定制方便的小目标探测声纳显控软件[1]可以有效改善小目标探测系统的操作性。

本文介绍一种基于 Qt 的小目标探测声纳显控软件设计实现方法。满足跨平台移植需求, 同时采用静目标抑制方法, 有效降低虚警率。

2. 基于 Qt 的显控软件构架

Qt 是一种面向对象、先进的、跨平台图形库软件[2], 具有广阔的应用前景。Qt 支持 VxWorks、Windows、Linux 等操作系统, 只需通过交叉编译后就可使源代码在不同平台上运行。而且 Qt 模块化程度高并同时支持 OpenGL、OpenCV 等[3], 对于开发人员而言十分便捷且友好, 因此选择 Qt 作为本文显控软件的开发平台。

本文提出的声纳显控软件分为参数显示、画面显示与控制菜单三部分。架构框图如图 1 所示。实际显控界面如图 2 所示。

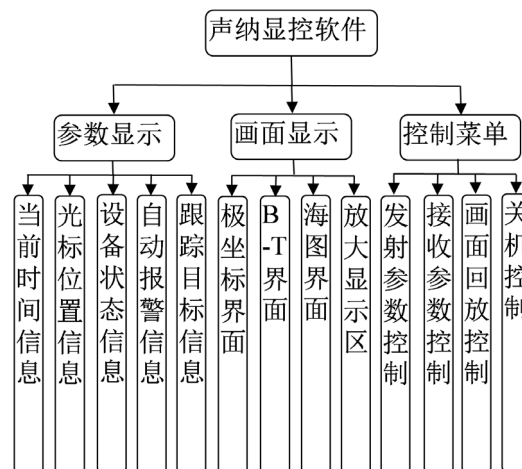


Figure 1. Display and control software architecture

图 1. 显控软件架构图

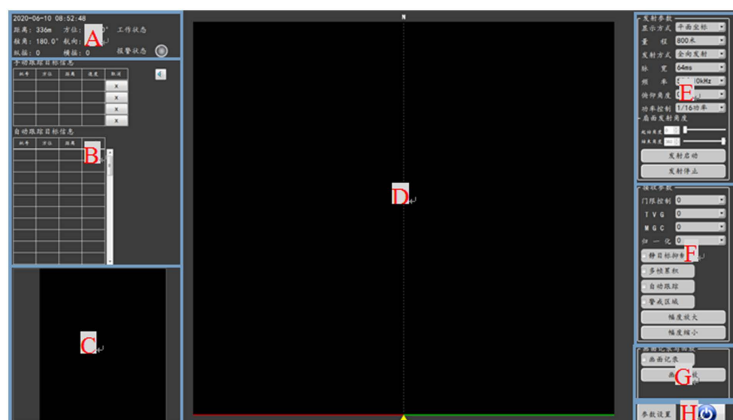


Figure 2. Display and control software interface
图 2. 显控软件界面

A 信息显示区：包含当前时间、光标位置信息、航速、航向、纵横摇信息、工作状态及报警状态显示。

B 跟踪目标信息显示区：显示当前手动或自动跟踪目标的批号、距离、方位、速度信息。

C 放大区：以光标为中心将回波放大显示。

D 回波显示区：显示回波信息数据。

E 发射参数设置区：包含显示方式、量程、发射方式、脉宽、频率、俯仰角度、功率控制、扇面发射角度设置、发射启动、发射停止控制。

F 接收参数设置区：包含门限、TVG、MGC、归一化、静目标抑制、多帧累积、自动跟踪、警戒区域设置、幅度放大、幅度缩小控制。

G 画面记录及回放区：包含画面记录及画面回放显示。

H 参数设置及关机区：参数设置及关机按钮。

2.1. 参数显示

参数显示区域位于界面左侧中上部，主要包含当前时间、光标位置信息、设备状态信息、自动报警信息、跟踪目标信息等。参数显示部分主要采用 QLabel 及 QTableWidgetItem 两种控件进行显示。

QLabel 是 Qt 中最基本的小部件类。一般用于显示当前日期时间、光标位置信息等标签文本，Qt 默认坐标系原点位于左上角。X 轴从左向右增长，Y 轴从上到下增长。通过 mousemoveevent 事件，获得光标当前位置的坐标信息。

QTableWidgetItem 是一种常见的 Qt 程序显示数据表空间，类似于 vc, c# 中的 datagrid，通常用于大容量文本跟踪目标信息等。实现跟踪目标信息实时更新显示。

2.2. 画面显示

画面显示区位于界面中部及左侧下部，界面中部为回波显示区域，可根据选择显示 B-T 界面、极坐标界面、海图界面等；左侧下部为放大区，以光标位置为中心放大显示。画面显示部分采用 PaintEvent 实现。paintEvent(QPaintEvent*) 函数是 QWidget 类中的虚函数，用于 ui 的绘制。接收完一个周期的数据后执行 update()，当绘制事件发生时，更新区域通常被擦除。如果需要在上次绘制的基础上进行绘制，用一个临时变量存储上次绘制后的图，然后在这个图上进行绘制，最后再直接显示这个图。只要窗口部件需要被重绘就被调用。每个要显示输出的窗口部件必须实现它。这个事件处理器可以在子类中被重新实

现来接收绘制事件。

由于每个周期的数据量非常大,绘图可能需要几秒钟甚至更长的时间,而且有时还会出现闪烁现象,为了解决这些问题,可采用双缓冲[4][5]技术来绘图。

首先,创建一个缓冲区(pix)和一个临时缓冲区(tempPix),将每个周期的数据分批写入缓冲区。当所有数据都写入缓冲区时,将数据复制到临时缓冲区。主体部分绘制临时缓冲区的内容,判断绘制是否完成。如果绘制完成,更新启动缓冲区以接收下一个周期的数据,并清除临时缓冲区影响区。

整个图像数据的完全刷新占用了太多的资源,而双缓冲区绘图可以实现局部刷新,只更新两个接收数据之间的区域变化,大大降低了资源消耗。if(isDrawing)

```
{
    tempPix = pix;
    QPainter pp(&tempPix);
    pp.drawRect(x,y,w,h);
    painter.drawPixmap(0,0,tempPix);
}
else
{
    QPainter pp(&pix);
    pp.drawRect(x,y,w,h);
    painter.drawPixmap(0,0,pix);
}
```

2.3. 控制菜单

控制菜单区域位于界面右侧,主要包含声纳发射参数控制(量程、频率、功率、脉宽、发射启动及停止等)、接收参数控制(门限控制、TVG、MGC、幅度放大及缩小等)、回放控制等。控制菜单部分采用 QComboBox 及 QPushButton 进行实现。

QComboBox 提供了一种以占用最少屏幕空间的方式向用户显示选项列表的方法。组合框是一个显示当前项目的选择小部件,可以弹出可选项目列表。组合框可以是可编辑的,允许用户修改列表中的每个项目。采用 currentIndexChanged(int)及 activated(int)实现量程、频率、发射方式等复数选择框。

QPushButton 控件是一个非常常用的按钮控件,它继承自 QAbstractButton。采用 clicked()实现发射控制、幅度控制。

3. UDP 数据传输

数据发送方通过源端口发送 udp 数据报文[4]。Udp 协议使用标头中的校验和来确保数据安全。校验和是由数据发送者的一个特殊算法计算出来的,如果一批数据在传输过程中损坏,发送者和接收者的计算值将不匹配,并将检查传输是否有错误。

```
Receiver= new QUdpSocket(this);
Receiver->bind();//设置端口号
int i=30*1024*1024;
::setsockopt(receiver->socketDescriptor(),SOL_
SOCKET,SO_RCVBUF,(char*)&i,sizeof(i));//
```

设置缓冲区大小为 30 M，由于数据传输速度过快且数据量较大，使用默认缓冲区大小将导致大量数据丢包

```

Connection(receiver,SIGNAL(readyRead()),this,SLOT(processPendingDatagram);
void MainWindow::processPendingDatagram()
{
    QByteArray datagram;
    Datagram.resize(receiver->pendingDatagramSize());
    If((datagram[0]==//数据头及标志位)
    {
        Pdata1=datagram[];//波束批次数据低位
        Pdata2=datagram[];//波束批次数据高位
        a2=Pdata1+Pdata2*256;
        for(a1=0; a1<//数据长度; a1++
        {
            H[a1][a2]= datagram[]//数据低位+ datagram[]//数据高位
        }
    }

```

4. 嵌入海图

为了更好地显示威胁目标的态势，可以通过嵌入预下载的高清图片实现。采用 QImage 生成二维纹理的方法。将图片导入 Qt 的 QImage 类的一个对象中，当然也可以通过 QPainter 直接绘制到一 QImage 对象中。然后在 OpenGL 中将 QImage 的对象转化为 texture，通过 OpenGL 中贴纹理的方式，将包含图片显示在 OpenGL 中。

```

glTexImage2D(GL_TEXTURE_2D, //指定纹理目标
0, //指定多级渐远纹理的级别
GL_RGB, //纹理储存为 RGB 值
width, height, //图像高度及宽度
0,
GL_RGB, //源图格式
GL_UNSIGNED_BYTE, //源图数据类型
data); //源图数据
glGenerateMipmap(GL_TEXTURE_2D);

```

根据实际地理坐标嵌入海图画面如图 3 所示。

5. 静态目标抑制

针对小目标等动态目标，采用静态目标抑制技术剔除静态目标可以有效降低虚警率。

背景差分法是静态场景运动目标识别的常用方法。该算法将当前帧数据与背景帧数据进行区分，得到目标运动轨迹的灰度图，并采用阈值化方法提取运动目标，为了避免环境杂波的影响，同时根据当前采集到的数据对背景帧数据进行更新。

在背景建模时，采用混合高斯模型[6]，对背景像素区进行更新。利用像素在较长时间内的概率密度表示背景，然后使用统计差分进行目标像素判断。由于处理 R、G、B 三色通道数据运算量较大，故采用 1 维幅值数据进行处理，之后通过色棒转化为伪彩色图像显示。



Figure 3. Chart interface display
图 3. 海图界面显示

在混合高斯背景模型中，不同像素点间的处理都是相互独立的，单个采样点 X_t 服从的混合高斯分布概率密度函数为

$$P(X_t) = \sum_{i=1}^K \omega_{i,t} * \eta(X_t, \mu_{i,t}, \Sigma_{i,t})$$

其中， K 是高斯函数的个数，本文取 3， X_t 是像素点 t 时刻的值， $\omega_{i,t}$ 为 t 时刻第 i 个高斯分布的权重， $\eta(X_t, \mu_{i,t}, \Sigma_{i,t})$ 为 t 时刻第 i 个高斯分布的概率密度函数。

$$\eta(X_t, \mu_{i,t}, \Sigma_{i,t}) = \frac{1}{(2\pi)^{\frac{n}{2}} |\Sigma|^{\frac{1}{2}}} e^{-\frac{1}{2}(X-\mu)^T \Sigma^{-1}(X-\mu)}$$

对每个新像素值 X_t ，与当前 K 个高斯模型进行比较，若该像素值与背景模型的均值偏差在 2.5σ 内，即 $(X_t - \sigma_k)^2 \leq 2.5^2 * \Sigma_k$ ，则该像素点属于背景点，否则属于前景点。

各个模式权重重新归一化

$$\omega_{k,t} = (1-\alpha)\omega_{k,t-1} + \alpha(M_{k,t})$$

对于匹配模型 M_k ， $t = 1$ ，否则 M_k ， $t = 0$ ；

对于未匹配的模型，均值 μ 与标准差 σ_i 保持不变。对匹配模型的参数进行更新。

$$\rho = \alpha * \eta(X_t | \mu_k, \sigma_k)$$

若 t 时刻像素 X_t 与第 i 个高斯模型匹配，则该高斯模型被更新。

$$\mu_t = (1-\rho) * \mu_{t-1} + \rho * X_t$$

$$\sigma_t^2 = (1-\rho) * \sigma_{t-1}^2 + \rho * (X_t - \mu_t)^T (X_t - \mu_t)$$

各高斯分布按照 $\frac{\omega_i}{\sigma_i^2}$ 降序排列，权重大，标准差小的模式靠前。

选取前 B 个高斯分布作为背景的背景分布， B 有下式确定：

$$B = \arg \left(\min \left(\sum_{k=1}^b w_k > T \right) \right)$$

其中 T 为预设的阈值，本文取 0.75。原始图像如图 4 所示，建立的混合高斯背景模型如图 5 所示，静目标抑制后图像如图 6 所示。

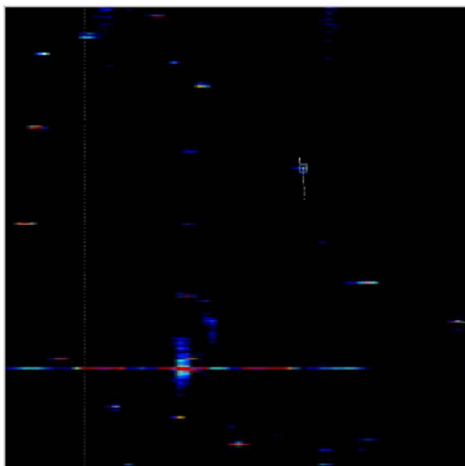


Figure 4. Original image
图 4. 原始图像

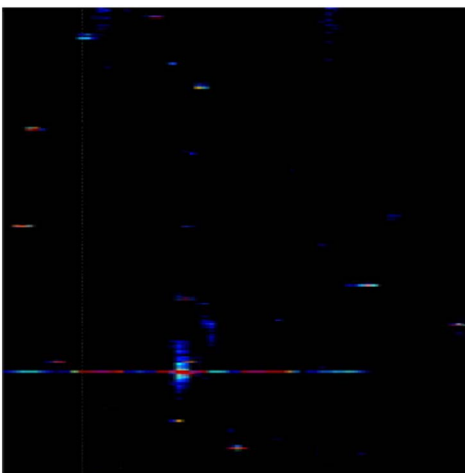


Figure 5. Gaussian mixture background model
图 5. 混合高斯背景模型

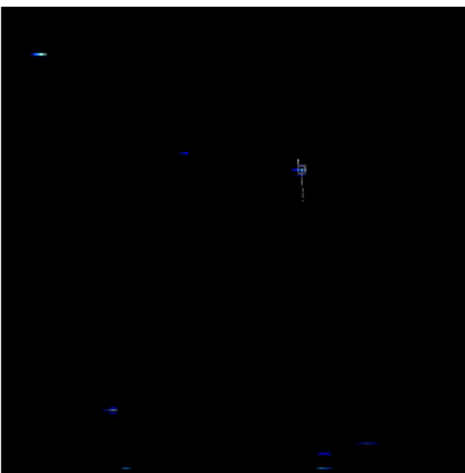


Figure 6. Image after static target suppression
图 6. 静目标抑制后图像

6. 结束语

本文提出了一种基于 Qt 图形开发工具和面向对象技术的小目标探测声纳显示控制接口软件的设计方法。设计了一种人机界面友好、人机交互方便、信息显示直观、功能定制方便的声纳显示控制界面。

参考文献

- [1] 刘娜, 肖雅静. 三维成像声纳显控软件系统设计[J]. 电子工业专用设备, 2017, 46(2): 52-54.
- [2] 邓子强, 王玉玫, 邓红艳, 张宇. 基于 Qt 绘图系统的图形应用优化研究与实现[J]. 软件, 2016, 37(12): 59-62.
- [3] 孙裕超, 张学磊, 田甜, 王文琮. 多波束成像声纳显控软件设计与实现[J]. 电声技术, 2017, 41(3): 82-87.
- [4] 冯贝. 基于 Qt 的分布式雷达生命探测显控软件设计[D]: [硕士学位论文]. 成都: 成都理工大学, 2019.
- [5] 陈艳静. 探鱼声纳系统综合显控软件[D]: [硕士学位论文]. 哈尔滨: 哈尔滨工程大学, 2014.
- [6] 范超男, 李士心, 张海, 等. 基于改进的混合高斯模型运动目标检测算法研究[J]. 天津职业技术师范大学学报, 2019(4): 25-31.