基于SCADE Display工具实现视频叠加功能的开发实例

王文恺[1]，翟志豪[2]

（1.苏州长风航空电子有限公司，江苏 苏州 215151；
2.93145部队，江苏 苏州 215151）

摘　要： 传统座舱显示系统的视频叠加功能通常依赖硬件逻辑器件实现，而SCADE Display软件环境缺少直接视频处理能力。对此，文章提出一种基于SCADE Display与OpenCV的视频叠加实现方法。首先，利用OpenCV完成视频帧解码，获取待显示的视频图像数据；其次，结合SCADE Display纹理贴图机制，将视频帧数据转换为可显示的纹理资源，并通过动态替换纹理实现视频叠加显示；最后，基于SCADE Display图形建模流程和纹理处理机制，完成视频叠加功能的软件化实现。该方法无需额外硬件支持，可应用于座舱显示系统仿真与显示控制软件开发，减少实际系统与仿真环境之间的开发差异。

关键词： SCADE Display；OpenCV；视频叠加；座舱显示系统；纹理贴图

中图分类号： TP311　　**文献标识码：** A

Video overlay function implementation based on SCADE Display tool

WANG Wenkai[1], ZHAI Zhihao[2]

(1.Suzhou Changfeng Aviation Electronics Co., Ltd., Suzhou, Jiangsu 215151, China;
2.93145 Army, Suzhou, Jiangsu 215151, China)

Abstract: Traditional video overlay functions in cockpit display systems are usually implemented by hardware logic devices, while the SCADE Display software environment lacks direct video processing capability. To address this issue, a video overlay implementation method based on SCADE Display and OpenCV is proposed. First, OpenCV is used to decode video frames and obtain video image data for display. Second, based on the texture mapping mechanism of SCADE Display, the video frame data is converted into displayable texture resources, and dynamic texture replacement is applied to achieve video overlay display. Finally, the video overlay function is implemented through the graphical modeling process and texture processing mechanism of SCADE Display. The proposed method requires no additional hardware support and can be applied to cockpit display system simulation and display control software development, reducing the development differences between actual systems and simulation environments.

Key words: SCADE Display, OpenCV, video overlay, cockpit display system, texture mapping

1　引言

随着计算机技术和数据总线技术的发展，航空电子系统逐渐向数字化方向演进，座舱显示系统也由传统仪表显示方式发展为以多功能显示器为核心的综合显示模式。多功能显示器能够集成飞行状态、导航信息以及任务数据等内容，为飞行员提供直观的信息显示与交互支持。

SCADE是一种面向安全关键嵌入式软件开发的工具体系，而SCADE Display是该体系下的图形显示开发工具，可用于航空电子系统中座舱显示软件的设计与开发[1]。目前，座舱显示系统中的视频叠加功能通常依赖FPGA等硬件逻辑器件实现。然而，在基于SCADE Display的软件开发环境中引入额外硬件支持会增加系统集成复杂度，且仿真环境与实际系统在功能实现方式上存在一定差异。针对上述问题，本文结合OpenCV视频处理能力与SCADE Display纹理贴图机制，提出一种基于软件的视频叠加实现方法。

2　SCADE及SCADE Display开发环境

SCADE采用模型驱动开发方式，以“正确构建”（Correct by Construction）为设计理念，支持需求建模、图形化设计、模型检查、仿真验证及代码生成等软件开发环节。通过建立形式化的软件模型，SCADE能够在设计阶段对系统功能进行分析和验证，并保证模型、仿真结果与生成代码之间的一致性[2]。同时，SCADE支持生成符合工

程应用需求的ANSI C或Ada代码，可减少传统手工编码过程中的重复性工作，适用于航空电子等对安全性和可靠性要求较高的嵌入式系统开发。

SCADE Display支持图形对象创建、属性配置、变量绑定与显示逻辑控制等功能，可实现显示元素的构建和动态更新。在SCADE Display中，图形对象可通过纹理资源和变量映射机制完成显示内容控制，为后续视频帧数据转换与纹理替换奠定基础[3]。

3 SCADE Display图形化建模技术

以SCADE代码生成器（KCG）为例，其在DO-178B软件开发流程中的应用情况如表1所列，部分验证工作可通过工具支持完成，从而减轻开发过程中的验证负担[4]。

表1 KCG的使用及验证

验证	需求	设计	编码	集成	验证	合计
DO-178B规定的操作项	7	13	6	6	8	40
KCG可完全省略项	4	8	6	0	3	21
KCG可部分省略项	3	4	0	6	0	13

基于SCADE Display，开发人员可通过图形对象、变量以及逻辑关系建立显示模型，实现显示元素创建、数据绑定和动态更新。实际开发过程中，显示模型需要根据座舱显示系统的功能需求，对输入数据、处理逻辑以及显示内容进行统一设计。

3.1 顶层设计需求分析

座舱显示系统或仿真软件中的显示模块需根据外部输入数据完成信息显示、状态监控以及交互控制等功能。在SCADE Display建模过程中，应根据显示需求对输入数据、显示逻辑以及输出内容进行分析，明确各功能模块之间的数据关系。

（1）数据采集功能：接收来自不同子系统的数据，并根据显示需求提取相关参数，将其转换为显示对象所需的数据输入，为后续显示更新提供数据来源。

（2）数据处理与状态监控功能：对输入数据进行解析和状态判断，根据数据有效性更新显示内容。在未接收到有效数据时，可在对应显示区域显示“-”或“/”等提示字符；当检测到子系统处于异常或离线状态时，可通过颜色或提示信息标识当前状态。

（3）数据交互功能：根据用户操作向相关子系统发送控制信息，并接收反馈数据，实现显示界面与外部设备之间的信息交互。

3.2 图形化建模

SCADE Display采用图形化建模方式完成显示界面的设计与开发。开发过程中，显示界面由图形对象、变量以及逻辑关系共同构成，开发人员可通过图形元素配置和数据关联实现显示内容的创建与动态更新。图形化建模过程主要包括图形对象设计、显示逻辑控制以及数据传递等环节。

3.2.1 图符绘制

以速度表盘图符（见图1）为例，该图形对象与同一画面中的其他图形对象位于同一图形设计文件的同一层上，绘制内容如下：（1）刻度带和刻度线采用环形图元和线段图元进行绘制，并通过颜色填充实现刻度显示；（2）背景图形采用扇形图元完成绘制；（3）文字标识和实时数值利用文本图元进行显示，用于呈现速度信息及实时数据；（4）指针采用自定义多边形图元绘制，并通过关联角度参数控制其旋转，实现速度变化时的动态显示。此外，SCADE Display支持对颜色、线宽、线型、字体以及纹理等显示资源进行统一管理，便于不同图形对象保持一致的显示风格。

图1 速度表盘

3.2.2 图形对象逻辑控制

在SCADE Display中，图形对象的动态显示主要通过变量绑定机制实现。需要实时变化的图形属性可与变量（在SCADE Display中称为“plugs”）建立关联，并将变量映射至图形对象的位置、颜色、角度等属性。当变量值发生变化时，关联属性同步更新，从而实现图形对象的动态显示。

对于简单控制逻辑，SCADE Display可通过表达式完成变量计算，包括算术运算、逻辑运算以及类型转换等操作。对于复杂的数据处理和逻辑判断，可在外部工程代码中完成计算，并将计算结果传递至对应plugs，实现外部程序与SCADE Display之间的数据交互。

此外，SCADE Display在显示更新过程中会根据设定周期刷新图形对象，将动态数据实时反映至显示界面中。通过变量绑定和外部逻辑扩展，图形对象能够根据输入数据变化实现相应的状态更新。

3.2.3 数据传递

SCADE Display中的显示数据主要通过变量进行传递，变量可根据数据来源和使用方式设置为输入变量、输出变量、局部变量或常量等类型，以管理外部输入数据和内部计算数据。同时，显示对象中的动态属性可与变量建立关联，通过变量值变化控制图形元素的位置、颜色、角度等显示参数。当外部数据更新时，相关变量

同步变化，并驱动图形对象完成显示更新。对于周期性输入数据，Memory 属性可保存变量上一周期的数值，使显示对象在当前周期未接收到新数据时保持已有状态，从而保证显示连续性。

3.3　模型仿真验证

SCADE Display 提供的仿真环境可用于验证图形对象、变量绑定及显示逻辑的正确性。开发人员能够通过仿真过程观察输入数据、内部变量和输出结果的变化情况，从而检查显示对象能否根据数据变化实现正确更新。

在软件部署前，模型仿真验证有助于发现显示逻辑和数据传递过程中存在的问题，确保图形对象、控制逻辑及外部输入数据之间的关联关系满足设计要求，为后续视频叠加功能的实现提供条件。

3.4　嵌入式代码生成

SCADE 支持根据经过验证的设计模型自动生成嵌入式代码，实现从图形化模型到软件实现的转换。生成代码主要面向工程应用，可支持 ANSI C 或 Ada 等编程语言，并通过规范化的数据结构和代码组织方式保证模型设计与程序实现之间的一致性。与传统手工编码方式相比，基于模型的代码生成方式能够减少重复性编码工作，降低人工实现过程中可能引入的错误风险。完成模型设计、逻辑验证以及仿真分析后，开发人员可利用 SCADE 代码生成工具将显示对象、变量关系和控制逻辑转换为软件模块，为后续座舱显示系统集成提供支撑。

4　基于SCADE Display与OpenCV的视频叠加开发实例

SCADE Display 不具备视频解码与实时视频叠加能力，传统座舱显示系统中的视频叠加通常依赖 FPGA 等硬件逻辑器件实现。为减少对额外硬件的依赖，本文将 OpenCV 的视频解码能力与 SCADE Display 的纹理贴图机制相结合，通过软件方式完成视频帧读取、纹理更新及显示对象刷新，实现座舱显示系统中的视频叠加功能。该实现方式既可用于实际座舱显示软件，也可用于相应仿真环境，从而减少两类开发环境在实现方式上的差异。

4.1　OpenCV 视频帧解码

SCADE Display 和 OpenGL 主要用于图形显示与渲染，本身不承担视频文件的解码处理，因此需借助 OpenCV，FFmpeg 等视频处理库完成视频帧读取。本文采用 OpenCV 实现视频解码。OpenCV 支持图像与视频读取、存储和处理，可在 C，C++，Python 等编程环境中使用。

本文以 Visual Studio 工程为例，将 OpenCV 相关库集成至座舱显示或仿真开发工程中。通过调用 OpenCV 中的 VideoCapture 类接口，利用 open，read 和 release 函数，分别完成 MP4 视频文件打开、视频帧读取及资源释放。视频叠加运行期间，程序按照设定周期持续读取后续视频帧，并将解码后的图像数据作为后续纹理更新的数据源。

4.2　SCADE Display 纹理贴图处理

OpenCV 解码得到的视频帧数据需要转换为 SCADE Display 可识别的纹理资源，才能在显示界面中进行渲染。为此，在 SCADE Display 纹理表中创建初始纹理资源，其分辨率与待显示视频的分辨率保持一致。本文采用纯白色 JPG 图片作为初始纹理，由于图像内容较为简单，可减少纹理资源占用。在 SCADE Display 的 Specification 或 Reference Object 中创建用于视频显示的矩形区域，并将上述纹理资源绑定至该图形对象。完成纹理绑定后，通过 UV 映射功能调整纹理坐标，使纹理内容按照设定方向和范围显示。具体操作分别如图 2 和图 3 所示。

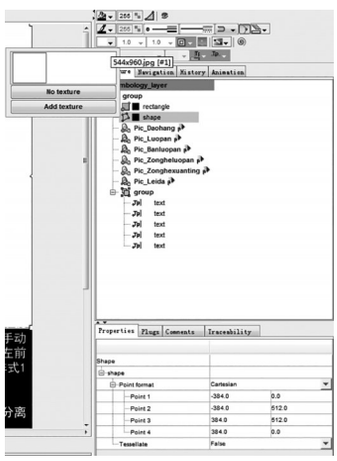

图 2　纹理插入

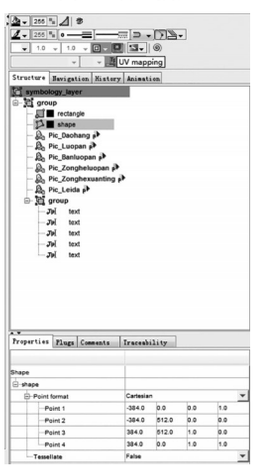

图 3　UV 映射操作

4.3　视频纹理动态替换

视频叠加的核心是将 OpenCV 解码后的视频帧更新至 SCADE Display 纹理资源。程序按照设定周期读取视频帧，并将图像数据转换为新的纹理内容，通过调用 SCADE 提供的 sglBindTexture 函数重新绑定纹理资源，以替换显示对象中的初始纹理。

随着视频帧持续更新，显示区域中的纹理内容同步变化，从而形成连续的视频显示效果。基于 OpenCV 视频解码和 SCADE Display 纹理动态替换机制，可在无需

（下转第 203 页）