TES 3D Surround View – Software Engine
Product Brief

V 1.4 Mar 29th 2022

Overview

TES 360° 3D Surround View (**TSV**) is the latest technology of driver assistance systems with vehicle cameras. Designed to increase safety and comfort through improved situation awareness it supports the principle of accident avoidance. TSV meets the rising demand for a simple and robust system for automotive, industrial and safety markets.

The **TSV** technology is offered as software component (Library) for various embedded platforms and operating systems.
By enabling the use of modern standard 3D acceleration GPUs it provides a high level of performance and portability while minimizing system resources usage, in particular CPU load. It utilizes the generic industry proven OpenGL (ES) API for main rendering tasks and adapts easily to a range of hardware platforms.

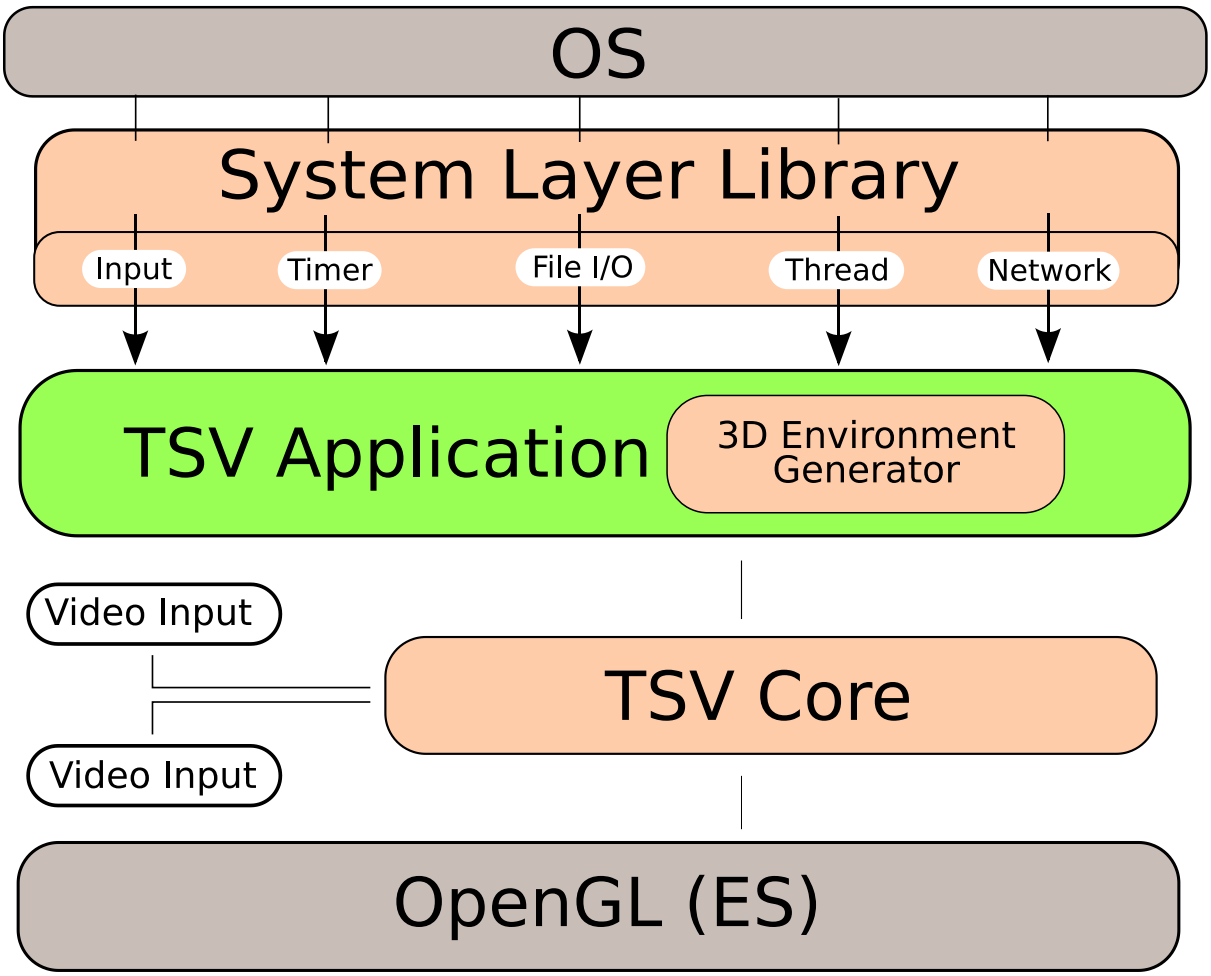
Besides the well-known Top View or Bird View, a single viewing perspective from exactly above the vehicle, **TSV** supports any viewer position and viewing angle from outside of the vehicle, i.e. full **360° Surround View**. The viewing position can be changed on-the-fly or by selecting one of several predefined viewing positions.

TSV builds on years of experience in surround view technology and offers a very high level of flexibility and customizability to cope with specific application and system requirements often found in special vehicle applications such as construction machines.
Designed for integration in modern embedded real-time video processing systems and providing well defined interfaces it integrates easily within the customers software system, e.g. from video input and application layer point of view. The multi-platform architecture allows porting to any operating system.

TSV is highly efficient and supports any kind of video input options, from analog (CVBS) through digital and network connected cameras.

Technology Details

High Level Architecture



Deliverables

- Binary
  - TSV Core : shared/static library with C header files
    - Implements 3D rendering
    - API for image input as raw buffers
    - API for observer position and camera data setting
  - System Layer Libraries: shared/static libraries with C++ header files
    - Support library providing OS abstraction layer for standard OS features
    - Support library providing abstraction for all established GL, Display and Window Systems (Wayland, X11, GLX, direct use of framebuffer or integrated in Qt GUI)
  - 3D Environment mesh generator : dynamically creates bowl projection surface
- Source
  - TSV Application : C++ reference application implementation
  - Video Input : Frame grabber examples
    - gstreamer

- MJPEG Software Decoder
- Decode various static images based on OpenCV

Features

- Camera system agnostic, no limit on camera type or count (within platform resource constraints)
- Cameras can be mounted anywhere on the vehicle
- Auto-Calibration for setup of camera on vehicle within minutes
  - Simple “Standard” Mode for “standard” camera setups
  - “Strict Mode” allowing complex camera positioning with several overlapping areas
- Animated and freely adjustable observer positions (“Views”) in the 3D space, any number possible
- Dynamic environment model, adjustable to size of area of interest per “View” and also adjustable on-the-fly
- 360° Surround View as well as Sub-360° Surround View supported for setups where the cameras do not cover the complete vehicle surroundings
- Freely exchangeable 3D vehicle models with on-the-fly adjustable transparency
- Multiple 3D objects allowing
  - 3D vehicles with movable parts and
  - to mark objects or obstacles in the 3D scene by any kind of 3D object (frames, transparent boxes or 3D icons)
- Virtual rear-mirror mode: Mirroring of single camera input or of complete output scene
- Three stitching modes (Border-, Sector- or Seam-Blending)
- Line drawing: API to draw lines and polygones into the 3D scence, e.g. to indicate for distances, highlight objects or indicate the drive path
  - Option to “raytrace” lines from observer position to marked objects
- Multi-Window Mode allowing any number of “Views” in parallel, e.g. a top-view with a rear-view in parallel.
- Single camera “Views” in fisheye-mode or planar
- Well defined API (open in development release) allowing complete flexibility during integration and product development, e.g.:
  - Customization of the HMI: Control methodology as well as HMI overlay
  - Additional graphics elements in the 3D Scene (OpenGL context is available for programmer)
  - Support for new type of cameras

System Features

- Almost completely GPU based processing
  - No CPU load in steady state
  - Only minimal usage of CPU while switching view
  - Easily integrates into existing products where the GPU is underused
- Multi-threaded operation
  - Separate thread for **TSV** Core
  - Separated memory
  - Asynchronous operation
  - Eliminates unnecessary screen redraws
- Scalability and portability throughout the entire design
  - OS agnostic
  - Scales from single-core embedded CPU to multi-core Desktop PC

- Adapts the graphics quality to available resources
- Bandwidth reduction techniques
  - Uses hardware extensions to avoid data copying
  - Optimized for low throughput DDR memory
- Layered structure
  - Allows integration of any video source
  - Provides interface for external sensors
- Gstreamer input support: TSV as gstreamer appsink for easy integration of cameras with gstreamer pipeline

Rendering

- Takes advantage of OpenGL (ES) 2.0 rendering features
- High render quality:
  - Texture filtering
  - High polygon count
  - High definition camera support
  - Seamless stitching through quality blending
- Support of "direct texture" GPU capabilities to avoid data copying (support for different "direct texture" APIs)

Integration

- Small **TSV** Core library written in C
- Reference **TSV** Demo and Evaluation Application based on **TSV** Core as with application source code in C++
- Low resource consumption
- No window system required and support for Wayland, X-Windows, GLX or integration in Qt GUI
- HMI independent architecture

Supported target platforms and turn-key solutions

Turn-key solutions are available supporting rapid product developments, e.g. on the following target platforms:

- NXP i.MX6 with Linux OS (e.g. on Toradex "Apalis" iMX6D IT Computer On Module with "Ixora" Carrier Board)
- NXP S32V234 with Linux OS (e.g. on MicroSys "miriac" SBC-S32V234A Development Kit)
- Intel mobile and X86 desktop processors with Linux or Windows OS

Demo-, Evaluation- and Prototype releases for Windows PCs are available for evaluation and demonstration purposes.

Contact

TES Electronic Solutions GmbH
Wanadalenweg 20
20097 Hamburg
Germany
www.tes-dst.com
graphics@tes-dst.com

