

## Overview

**TES 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.

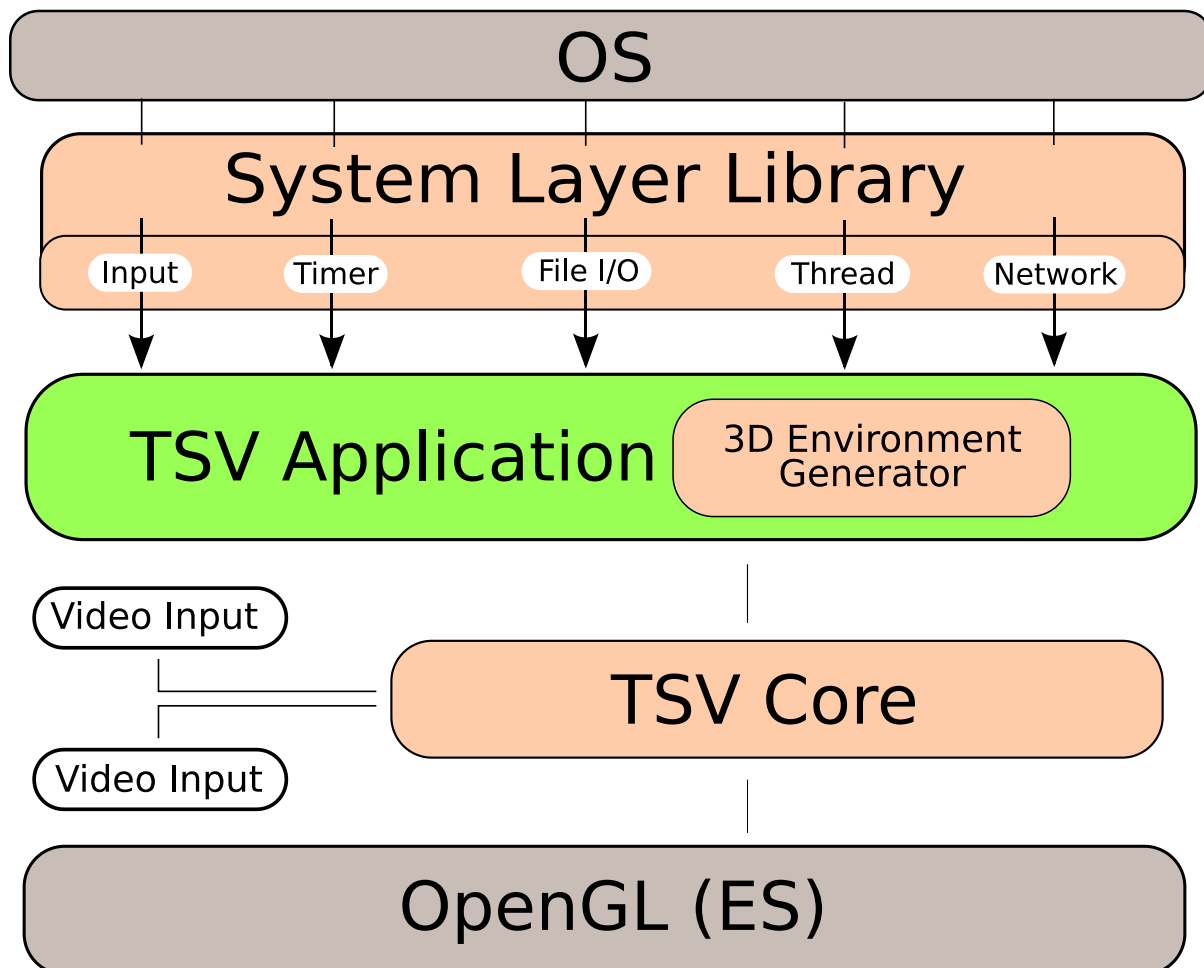
Designed for integration in modern embedded real-time video processing systems and providing well defined interfaces it integrates easily within the customers software system, e.g. from video input and application layer point of view. The multi-platform architecture allows porting to any operating system.

**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
  - **SystemLayer library** : shared/static library with C++ header files
    - Support library providing OS abstraction layer for standard OS features
  - **3D Environment mesh generator** : dynamically creates bowl projection surface
- **Source**
  - **TSV Application** : C++ reference application implementation
  - **Video Input** : Frame grabber examples
    - Video4Linux
    - MJPEG Software Decoder

### 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 fast setup
- Animated and freely adjustable observer positions, any number possible
- Freely exchangeable 3D vehicle model with on-the-fly adjustable transparency
- Mirroring of single camera input or of complete output scene
- Three stitching modes (Border-, Sector- or Seam-Blending)
- Dynamic environment model, changeable on-the-fly
- 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) e.g. for distance lines or highlighting of objects
  - Support for new type of cameras

### System Features

- Almost completely GPU based
  - 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

### 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

### 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
- HMI independent architecture

### Resource Usage

The actual resource usage of **TSV** depends highly on the camera and display systems used and the 3D model complexity.

A typical example includes:

#### Case 1

*TSV with 4 IP MJPEG cameras, combined 80Mbit data rate @30 fps and 1Mpx resolution. Low polygon car model, 1280x1024 screen resolution at 60Hz refresh rate. Implemented on i.MX6 Quad.*

- **GPU Usage**
  - 3-6 GFlops (proportional to pixel amount depicted on screen)
- **RAM Bandwidth**
  - 4\* 2 MB/s Read/Write (4 cameras MJPEG data)
  - 4\* 45 MB/s Read/Write (4 cameras RAW data)
  - 120 MB/s Write, 240Mb/s Read (frame buffer & display)
  - **300 MB/s Write, 420MB/s Read in TOTAL**
- **RAM Consumption**
  - 4\* 10 MB for camera data buffers
  - 2 MB for 3D environment and car model
  - 18 MB for camera video textures
  - **60 MB in TOTAL**

#### Case 2

*TSV with 4 analog cameras, PAL resolution @30 fps. Low polygon car model. 800x480 screen resolution at 50Hz refresh rate. Implemented on i.MX6 Solo*

- **GPU Usage**
  - 3-6 GFlops (Proportional to pixel amount depicted on screen )
- **RAM Bandwidth**
  - 4\* 15 MB/s Read/Write (4 cameras RAW data)
  - 45 MB/s Write, 73Mb/s Read (frame buffer & display)
  - **105 MB/s Write, 133 MB/s Read in TOTAL**

# TES 3D Surround View – Software Engine

## Product Brief

Nov 23<sup>rd</sup> 2017



- **RAM Consumption**
  - 2 MB for 3D environment and car model
  - 4 MB for camera video textures
  - **6 MB in TOTAL**

### 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 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**  
**Frankenstr. 3**  
**20097 Hamburg**  
**Germany**  
[www.tes-dst.com](http://www.tes-dst.com)  
[graphics@tes-dst.com](mailto:graphics@tes-dst.com)