

改进的车载全息透明影像方法^①



王焱焱¹, 胡 贵², 梁树宇², 李达明²

¹(北京工业大学 信息学部, 北京 100124)

²(阿波罗智能技术(北京)有限公司, 北京 100193)

通信作者: 王焱焱, E-mail: wyy15733179934@163.com

摘 要: 汽车本身的结构, 决定了汽车周围存在大量视野盲区, 驾驶员无法对行车周围环境做出准确判断, 不利于安全驾驶. 全息透明影像能够为驾驶员提供行车周围以及车底全部盲区信息辅助安全驾驶. 为解决图像拼接处有明显拼接缝的问题, 提出了基于距离的 Alpha 图像拼接算法, 并为拼接算法重新设计了三维模型. 对透明底盘功能做了如下 3 方面优化. 改进自行车模型算法, 在不影像效果的前提下, 减小了计算复杂度. 利用查表法提高方向盘角度转换为轮转角的精度, 解决了透明底盘与周围拼接错位问题. 采用分层渲染的方法, 优化透明底盘功能拼接缝问题. 实验表明该技术能够有效地提升渲染效果.

关键词: 全息透明影像; 图像拼接; 三维模型; 自行车模型; 查表法; 车载系统

引用格式: 王焱焱, 胡贵, 梁树宇, 李达明. 改进的车载全息透明影像方法. 计算机系统应用, 2022, 31(10): 60-71. <http://www.c-s-a.org.cn/1003-3254/8785.html>

Improved Method of Vehicle Holographic Transparent Image

WANG Yan-Yan¹, HU Gui², LIANG Shu-Yu², LI Da-Ming²

¹(Faculty of Information Technology, Beijing University of Technology, Beijing 100124, China)

²(Apollo Intelligent Technology (Beijing) Co. Ltd., Beijing 100193, China)

Abstract: The structure of cars determines that there are a large number of blind spots around them, and thus a driver cannot make accurate judgments on the surrounding environment, which is not conducive to safe driving. The holographic transparent image can provide the driver with information on all blind spots around and under the vehicle to assist in safe driving. To solve the problem of obvious seams in image stitching, a distance-based Alpha image stitching algorithm is proposed, and a three-dimensional (3D) model is redesigned for the stitching algorithm. The transparent chassis function is optimized in the following three aspects. The bicycle model algorithm is improved to reduce the computational complexity on the premise of no image effect. The table look-up method is used to improve the accuracy of the conversion of the steering wheel angle into the wheel angle, and the problem of stitching dislocation between the transparent chassis and the surroundings is solved. The method of layered rendering is adopted to optimize the seam problem of the transparent chassis function. The experiments indicate that this technology can effectively improve the rendering effect.

Key words: holographic transparent image; image stitching; 3D model; bicycle model; table look-up method; vehicle system

1 引言

1.1 研究背景

近年来, 随着国家经济与科学技术的不断发展, 人

们物质生活水平不断提升的同时, 我国的汽车相关行业也在蓬勃发展. 汽车普及给人们带来便利的同时, 也产生了交通安全问题. 根据国家统计局最近数据显示,

① 收稿时间: 2022-01-14; 修改时间: 2022-02-24, 2022-03-30; 采用时间: 2022-04-11; csa 在线出版时间: 2022-07-25

截至2020年末,共计发生24.47万起交通事故,造成6.17万人死亡和25.07万人受伤,经济损失达到13.1361亿元^[1]。很多交通事故的发生,并不是因为驾驶员驾驶技术不过关,汽车盲区使得驾驶员无法准确了解到汽车周围环境。汽车视野盲区示意图^[2],如图1所示。有调查显示,由汽车视野盲区造成的事故占比高达32%^[3]。因此,汽车厂商纷纷使用各类手段解决此痛点,车载环视影像应运而生。随着环视影像系统技术的成熟与成本的下降,越来越多的汽车厂商愿意在新款车型中装载环视影像系统。针对9个自主品牌代表企业旗下车型的驾驶辅助系统应用情况进行分析,产品布局数量占比已经超过50%^[4]。



图1 汽车视野盲区示意图

1.2 研究现状

全景环视最早由Kato等人在2006年提出^[5]。该概念的提出立即得到了各方的关注。次年日产公司发布首个商用产品^[6],环景监视系统AVM (around view monitor)。后续本田、福尔德、奔驰以及迈隆等公司也推出了相似产品。该系统也由图像直接堆叠过渡到图像融合拼接。2013年富士通公司提出三维全景环视^[7,8],从此环视影像由只有鸟瞰图的二维过渡到三维。

车载环视影像,在增加透明底盘功能后,消除了汽车周围的所有盲区,发展为车载全息透明影像。近几年来,高级辅助驾驶前装载量逐渐上升,环视影像作为高级辅助驾驶的主要功能之一,也迎来了快速发展。松下^[9]、现代汽车^[10]、LG电子^[11]、华为^[12]、百度^[13]、广汽^[14]等企业;北京交通大学^[15]、同济大学^[16,17]、普渡大学^[18,19]等国内外高校以及研究机构都开始从理论方面入手,

结合实践进行深入研究。

图像拼接技术中图像配准是核心技术,根据大量学者的研究,将图像配准算法分为两类,包括基于特征的图像配准算法、基于区域的图像配准算法^[20-22]。由于基于特征的图像配准算法具有计算量小、速度快等优势,该方式在车载环视影像中应用最广泛^[23-25],例如SIFT、SURF等算法。

首先,车载全息透明影像使用的是视角很大的鱼镜头,得到的图像存在明显的畸变,特征值计算比较困难。传统的解决方案先对原图去畸变,然后再对图像裁剪,最后计算特征值。这些操作极大地增加了后续渲染的复杂度。其次,在渲染过程中,每一帧图像都需要有矩阵乘运算,运算量大。再次,相邻两路摄像头的中心轴夹角接近90°,通过特征值求得的仿射变换矩阵存在很大误差,导致车载环视影像在图像拼接处,存在拼接缝的问题,尤其是相对于汽车距离较近的4个车角处。

针对传统解决方案存在的缺陷和问题,提出了基于距离的Alpha图像拼接算法,该算法只需在程序启动时计算一次,极大地减小了计算量。并且根据该拼接算法的特性,重新设计了三维模型。对透明底盘功能做了如下3方面优化:改进自行车模型算法,达到了减小计算复杂度,提高计算效率的目的;利用查表法,解决透明底盘与周围拼接错位问题;采用分层渲染的方法,优化透明底盘功能拼接缝问题。

2 软硬件系统结构与全息透明影像基本原理

2.1 硬件系统结构

全息透明影像的硬件系统,主要包括四路鱼眼摄像头、自动驾驶计算平台、车载视频显示器、整车动力底盘、车身及关键零部件等几部分。自动驾驶计算平台与汽车电子器件的连接结构,如图2所示。

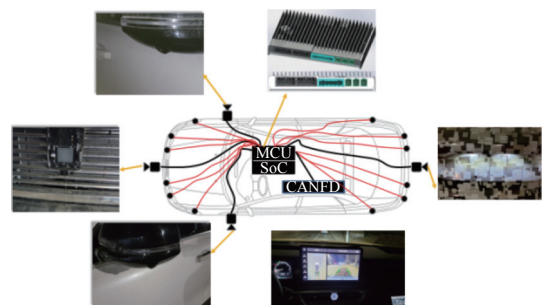


图2 自动驾驶计算平台与汽车电子器件的连接结构

(1) 传感器接入四路鱼眼摄像头通过 GMSL 接入、12 路超声波雷达采用 12 路硬线的方式接入。

(2) HMI 显示交互: 视频数据通过一路 LVDS 输出给中控车机, 交互类软开关等信息通过网关路由的 CANFD 总线传输。

(3) 整车交互: 整车动力底盘、车身等关键零部件通过网关路由的 CANFD 总线实现控制和状态报文的实时交互。

四路鱼眼摄像头的视场角为 190° – 200° ; 安装角度为 20° – 35° ; 有效覆盖范围为 3–6 m; 前鱼眼摄像头安装在车头前部中央 (前格栅附近); 左鱼眼摄像头安装在左后视镜下方; 右鱼眼摄像头安装在右后视镜下方; 后鱼眼摄像头安装在车尾后部中央 (后备箱把手附近)。自动驾驶计算平台的安装位置示意图, 如图 3 所示。



图 3 计算平台安装位置示意图

2.2 软件系统结构

软件系统结构主要分为 3 层, 依次为硬件层、中间件层和应用层。软件系统结构示意图, 如图 4 所示。

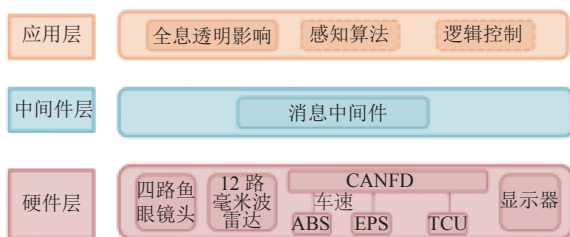


图 4 软件系统结构示意图

全息透明影像位于应用层, 它通过消息中间件获得摄像头图像、车速、方向盘转角以及档位等信息。消息中间件屏蔽硬件差异, 向上提供统一接口, 提供一致化的服务能力。ABS、EPS 以及 TCU 分别将车速、方向盘转角、档位等车身信息通过 CANFD 总线传递到消息中间件。显示器通过消息中间件获得需要显示的图像。

2.3 全息透明影像基本原理

全息透明影像使用 OpenGL 进行图形渲染。OpenGL 是一个包含了一系列可以操作图形、图像的 API^[26]。

OpenGL 做渲染时自身是一个巨大的状态机, 首先需要给状态机设置绘画的属性, 其次使用当前属性进行渲染。

OpenGL 的基本属性分别为: 坐标点、纹理和纹理坐标。当用 OpenGL 渲染简单三角形时, 需要 3 个顶点坐标、一张纹理和每个顶点坐标点对应的纹理坐标。顶点确定三角形的形状, 纹理确定三角形的颜色, 纹理坐标则确定使用纹理什么位置的颜色。渲染三角形的数据格式为 $\{X, Y, Z, U, V\}$, 其中 (X, Y, Z) 为顶点坐标; (U, V) 为顶点坐标对应的纹理坐标。三角形渲染流程图如图 5 所示。

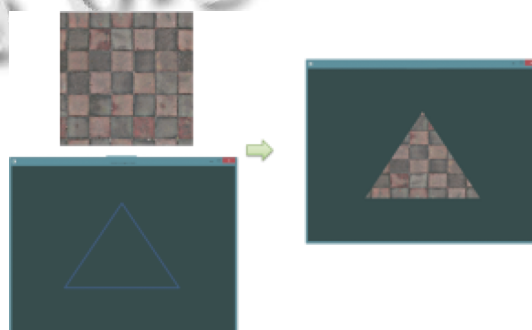


图 5 三角形渲染流程图

全息透明影像渲染使用的三维模型是由很多小三角形组成的。全息透明影像渲染使用的纹理是四路鱼眼摄像头的图像。利用基于距离的 Alpha 图像拼接算法, 可以求得三维模型中每个坐标点对应的纹理坐标。全息透明影像的渲染示意图如图 6 所示。

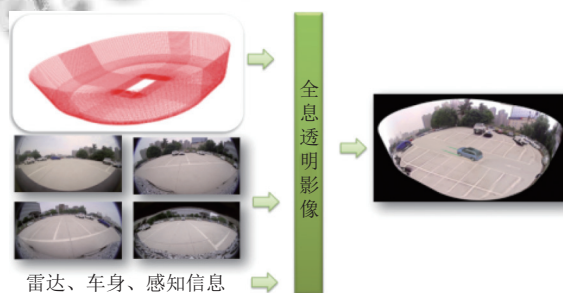


图 6 全息透明影像渲染示意图

3 基于距离的 Alpha 融合算法与三维模型

3.1 世界坐标系到纹理坐标系的计算方法

无论使用什么方法做图像拼接, 图像配准的主要任务是找到图像中具有相同特征的物体。也就是在世界坐标系下, 同一点体在两张图象上的位置。如果通过

计算可以求得世界坐标下的一点,在两张图像上的位置,那么将这两个位置上的像素做融合即可。

相机的成像过程是三维坐标点在世界坐标系、相机坐标系、图像坐标系和像素坐标系中的转换^[27]。世界坐标系以汽车正中心的地面投影点为原点建立,车头方向为X轴的正方向,垂直于X轴向汽车左侧为

Y轴正方向,向上为Z轴正方向。为了不同车型相机标定的准确,增加了汽车坐标系。汽车坐标系是以后轴中点在地面上的投影点作为原点建立的。

下面将介绍世界坐标系下的点 (x, y, z) 到对应相机图像纹理坐标 (u, v) 的计算过程。该过程需要用到相机的内外参数^[28]。相机的内外标定参数,如表1所示。

表1 四路鱼眼摄像头内外标定参数

参数	前鱼眼摄像头	左鱼眼摄像头	右鱼眼摄像头	后鱼眼摄像头
摄像头名称	front_short_camera	left_short_camera	right_short_camera	rear_short_camera
光心z轴偏移系数	1.072361	1.13377	1.0632808	1.111783
uv方向焦距	(595.97, 594.42)	(611.23, 610.37)	(592.1240, 592.0028)	(605.1539, 603.9822)
光轴中心像素坐标	(635.60, 361.09)	(636.36, 361.85)	(637.2952, 359.0182)	(637.8326, 360.1058)
径向畸变系数1	0.27255139	0.3191981	0.2503984	0.30630976
径向畸变系数2	-0.23509289	-0.26879808	-0.21842492	-0.25648171
切向畸变系数1	-0.00060594617	0.00070116267	0.00053668109	0.00087862421
切向畸变系数2	-0.00076437625	-0.00023533803	-0.0005157222	0.0011422584
像素宽度	1280	1280	1280	1280
像素高度	720	720	720	720
图像格式	RGBA	RGBA	RGBA	RGBA
绕Z轴旋转	-0.00139279	1.58483	-1.57613	3.13661
绕Y轴旋转	0.264607	-0.455093	0.848665	0.499128
绕X轴旋转	0.0149163	-1.15687	-0.219143	-0.0182796
位移向量	(3.696, 0.0016, 0.67)	(2.160, 1.030, 1.082)	(2.1915, 0.99396, 1.141)	(-0.873, 0.0629, 0.976)

首先,需要将顶点坐标从世界坐标系转换到对应相机坐标系。然而相机坐标系是以汽车坐标系为基础建立的,所以首先需要将顶点坐标转到汽车坐标系下。计算公式如下:

$$\begin{cases} x_{car} = x + length \\ y_{car} = y \\ z_{car} = z \end{cases} \quad (1)$$

其中, $(x_{car}, y_{car}, z_{car})$ 为顶点坐标在汽车坐标系下的位置; $length$ 为汽车中心到后轴的距离。

利用相机外参中绕Z轴旋转角度、绕Y轴旋转角度、绕X轴旋转角度和位移向量,可以求得3个旋转矩阵和1个位移矩阵。通过仿射变换就可以求得顶点坐标在相机坐标系下的位置。计算公式如下:

$$\begin{cases} rota = rota_x \times rota_y \times rota_z \\ \begin{pmatrix} x_c \\ y_c \\ z_c \\ k \end{pmatrix} = trans \times rota \times \begin{pmatrix} x_{car} \\ y_{car} \\ z_{car} \\ 1 \end{pmatrix} \end{cases} \quad (2)$$

其中, (x_c, y_c, z_c) 为顶点在相机坐标系下的位置; $rota_x$ 为绕X轴旋转矩阵; $rota_y$ 为绕Y轴旋转矩阵; $rota_z$

为绕Z轴旋转矩阵; $trans$ 为位移矩阵; k 为常量。

此时已经转换到相机坐标系。顶点由相机坐标系通过计算转换到像素坐标系,此时需要用到相机的内参。鱼眼摄像头是一种特殊的摄像头,首先阐述小孔成像模型的求解过程。在小孔成像模型下,首先需要将摄像机坐标系下的点 (x_c, y_c, z_c) 转换为归一化平面上的点。如式(3)所示:

$$\begin{cases} x' = \frac{x_c}{z_c} \\ y' = \frac{y_c}{z_c} \end{cases} \quad (3)$$

其中, (x', y') 为归一化平面上的点。

其次将归一化平面上的点转换为像素坐标系下的纹理坐标:

$$\begin{cases} u = f_x \times x' + c_x \\ v = f_y \times y' + c_y \end{cases} \quad (4)$$

其中, (u, v) 为纹理坐标; f_x, f_y 为uv方向焦距。 c_x, c_y 为光轴中心像素坐标。相机坐标系到像素坐标系计算过程原理图,如图7所示。

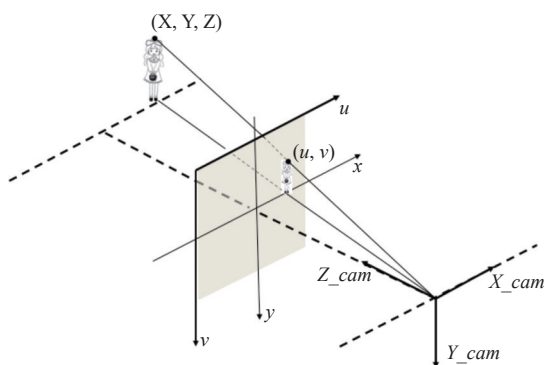


图7 相机坐标系到像素坐标系计算过程原理图

鱼眼摄像头图像存在明显的畸变. 因此需要在公式中增加去畸变处理和 Z 轴光心偏移系数. 具有超大视场角的鱼眼相机, 由于其球状镜头结构, 以及制造工艺的问题, 小孔成像的经典模型不适用于鱼眼相机的成像模型, 因此需要增加新的参数^[29]. 鱼眼相机成像模型, 如图 8 所示.

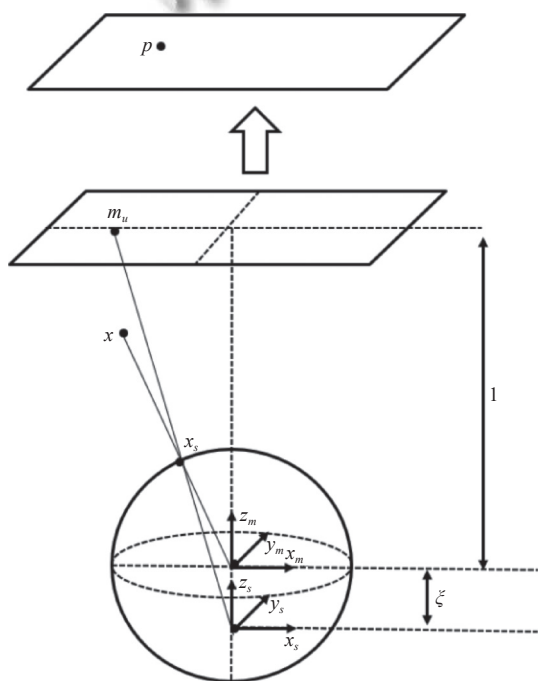


图8 鱼眼相机成像模型

鱼眼相机归一化平面的计算方法如式 (5) 所示:

$$\begin{cases} x' = \frac{x_c}{z_c + \xi} \\ y' = \frac{y_c}{z_c + \xi} \\ \xi = xi \times \sqrt{x_c^2 + y_c^2 + z_c^2} \end{cases} \quad (5)$$

其中, xi 为光心 Z 轴偏移系数.

鱼眼镜头成像存在的几何失真称为畸变. 根据 Brown 畸变模型, 畸变主要分为径向畸变和切向畸变^[30]. 径向畸变是从图像中心向外发散分布的畸变, 产生原因是光线在鱼眼透镜边缘光线的折射角比靠近中心的地方更大. 畸变的数学模型可以用主点周围的泰勒级数展开式的前几项进行描述, 通常使用前两项, 即 k_1 和 k_2 . 切向畸变是由于透镜本身与相机传感器平面(成像平面)或图像平面不平行而产生的, 畸变模型可以用两个额外的参数 p_1 和 p_2 来描述. 去畸变公式如下:

$$\begin{cases} m = 1 + k_1 r^2 + k_2 r^4 \\ x'' = x' m + 2p_1 x' y' + p_2 (r^2 + 2x'^2) \\ y'' = y' m + p_1 (r^2 + 2y'^2) + 2p_2 x' y' \\ r^2 = x'^2 + y'^2 \end{cases} \quad (6)$$

其中, k_1 为径向畸变系数 1; k_2 为径向畸变系数 2; p_1 为切向畸变系数 1; p_2 为切向畸变系数 2.

鱼眼镜头纹理坐标的计算公式如下:

$$\begin{cases} u = f_x \times x'' + c_x \\ v = f_y \times y'' + c_y \end{cases} \quad (7)$$

其中, u, v 为纹理坐标; f_x, f_y 为 uv 方向的焦距. c_x, c_y 为光轴中心像素坐标.

通过上述公式可以求得顶点 (x, y, z) , 由世界坐标系到像素坐标系 (u, v) 的变换.

3.2 三维模型的设计与优化

渲染车载全息透明影像就是在渲染一个 3D 物体. 三维模型相当于是在车载环视影像的一个“墙壁”, 将四路鱼眼图像按照一定的规则贴合到这个墙壁上, 就形成了具有立体效果的影像. 三维模型功能示意图, 如图 9 所示.

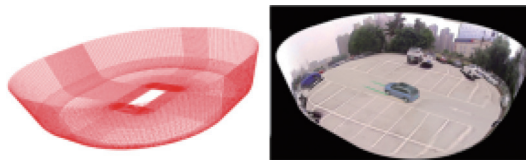


图9 三维碗状模型功能示意图

根据基于距离的 Alpha 融合算法特性, 我们对三维模型的需求有如下 3 点:

(1) 可以快速得找到拼接线, 并且可以简单地在拼接线附近找到图像融合区域.

(2) 减少点的数量. 以 TDA4 为核心的开发板, 索引的数据类型为 `unsigned short`, 占 2 个字节, 数值范围为 [0, 65535]. 在同一个索引缓冲对象中, 最多可以索引 65536 个顶点, 21845 个三角形. 因此三维碗状模型点的数量很多时, 需要对顶点数据进行分块, 再将数据拷贝进顶点缓冲对象中, 这样会生成多个内存 ID. 在渲染时就需分开渲染, 浪费资源.

(3) 三维碗状模型渲染结果需要平滑柔顺, 不能产生黑边、拼接缝明显等问题. 在汽车的 4 个角落处, 很容易由于三维碗状模型, 导致在相邻的两张纹理上不能采样到正确的颜色, 最终在做图像融合时产生明显的黑线.

通过调研发现, 根据几何模型的不同, 可以将三维模型分为经纬度三维模型^[31-33]、立方体三维模型^[34,35]和柱面三维模型 3 类^[36,37].

经纬度三维模型及其改进方式模型可以满足如上需求. 通过控制经线, 就可以找到拼接线, 控制角度找到拼接区域. 经纬度三维模型整体与细节, 如图 10 所示. 该模型的鸟瞰视图与 3D 视图渲染结果, 如图 11 所示.

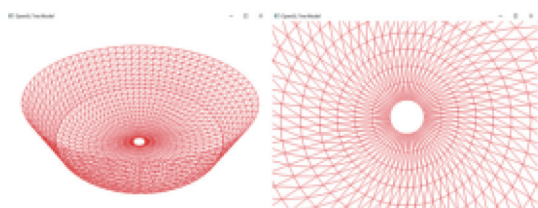


图 10 经纬度三维模型



图 11 经纬度三维模型渲染结果

通过观察可以发现, 汽车周围有一圈不规则的黑边, 这是由于在经纬度三维模型中, 汽车周围三角形的边与汽车底盘不平行, 导致有些三角形的部分点在汽车下方. 但是鱼镜头无法拍摄到汽车底部, 这些点在纹理采样时就会得到错误的 RGB 值, 就变为黑色. OpenGL 的特性会在三角形中间做线性插值处理, 会导致三角形从有颜色逐渐过渡为黑色. 解决此问题可以通过增加三维模型中三角形的密度, 但这会增加三

维模型中点的数量. 通过实验测试发现, 将点增加为之前的 3 倍, 效果依旧不明显.

根据问题, 重新设计三维模型, 在原有经纬度三维模型中间, 嵌入一个汽车大小的长方形, 确保三角形的所有顶点都位于汽车外部. 内嵌长方形三维模型整体与细节展示, 如图 12 所示. 该模型的 3D 视图渲染结果, 如图 13 所示.

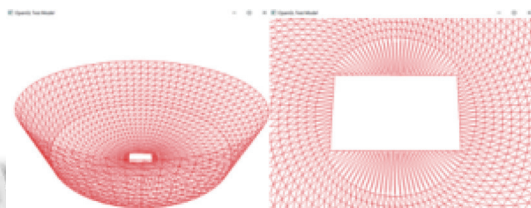


图 12 内嵌长方形三维模型



图 13 内嵌长方形三维模型渲染结果

通过观察可以发现, 有些地方存在明显的畸变. 这是因为, 只有顶点处做了去畸变处理, 三角形内部只做了线性插值, 并没有去畸变. 内嵌长方形与外接圆之间的三角形过大, 这会导致渲染结果存在畸变.

鉴于此, 重新优化三维模型. 将内嵌长方形更换为正方形网格, 并在正方形与外接圆之间插入点, 减小三角形的大小. 内嵌正方形三维模型整体与细节, 如图 14 所示. 该模型的鸟瞰视图与 3D 视图渲染结果, 如图 15 所示.

通过观察可以发现, 在汽车的 4 个角落产生了横向黑色细线. 车头前的两条黑线, 是由于靠近车角的正方形在前镜头的图像中无采样到正确的 RGB 值, 变为黑色. 与其他颜色融合后变为一条黑色细线. 后侧同理.

综合上述所有问题, 结合前几个模型各自的优点, 重新设计三维模型.

(1) 经纬度三维模型的优点在于, 可以用快速找到拼接线的位置, 并且可以方便地在拼接线周围找到图

像融合区. 因此在图像拼接区域使用经纬度三维模型
的方案.

(2) 嵌入长方形三维模型的优点在于, 可以让渲染流
程简单结果整齐. 因此在正中间使用汽车大小的长方形.

(3) 长方形外接圆会产生畸变, 因此长方形周边用
直线代替弧线.

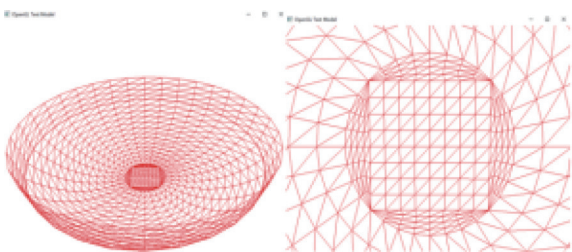


图 14 内嵌正方形三维模型



图 15 内嵌正方形三维模型渲染结果

按上述方案构建三维模型, 首先在模型中间使用
汽车大小的长方形. 其次在长方形的 4 个角落处, 以角
为圆心画四分之一圆弧. 最后长方形周围, 将相邻两个
圆弧用直线连接. 拼接区三维模型构建过程示意图, 如
图 16 所示. 该模型的渲染结果如图 17 所示.

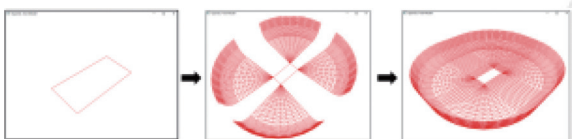


图 16 拼接区三维模型构建过程示意图

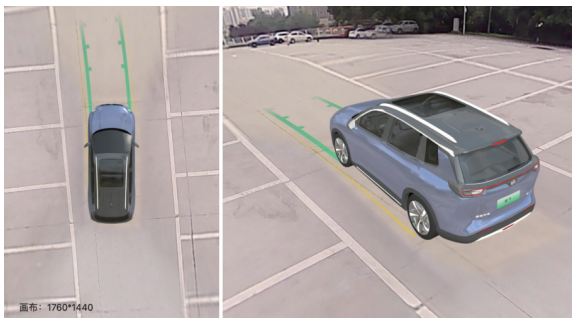


图 17 拼接区三维模型渲染结果

3.3 图像拼接与图像融合

在相邻两个图像之间做拼接融合, 需要找到一条
拼接线. 首先在三维模型中确定一条拼接线. 拼接线示
意图, 如图 18 所示.

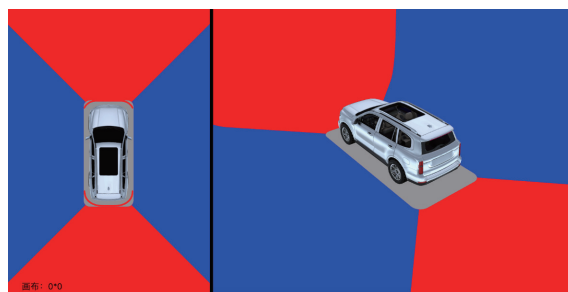


图 18 拼接线示意图

鱼镜头的视角很大, 这样会导致任意两个相邻
的摄像头图像, 会有很大图像重合的部分. 为了使得渲
染结果更加完美, 需要在两幅图像相接的地方做图像
融合. 在拼接缝两侧寻找适量列数的点, 作为图像融合
区. 在该区域进行图像的融合. 图像融合区示意图, 如
图 19 所示.

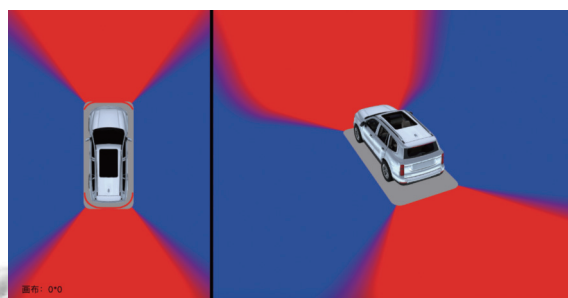


图 19 图像融合区示意图

图像融合区的顶点, 可以在拼接线的左右两张纹
理上分别求得一个纹理坐标 (U_1, V_1) 和 (U_2, V_2) . 利用
着色器语言 GLSL (OpenGL shading language) 中的
 $texture(ourTexture, vec2(U, V))$ 函数可以得到纹理
 $ourTexture$ 中纹理坐标为 $vec2(U, V)$ 的颜色. 由此可得
到顶点在两张图像上的对应的颜色. 公式如式 (8):

$$\begin{cases} outColor1_{rgb} = texture(ourTexture1, vec2(U_1, V_1)) \\ outColor2_{rgb} = texture(ourTexture2, vec2(U_2, V_2)) \end{cases} \quad (8)$$

其中, $outColor1_{rgb}$ 为顶点在拼接线左侧图像上对应
颜色的 RGB 值; $outColor2_{rgb}$ 为顶点在拼接线右侧图像
上对应颜色的 RGB 值. 将两个颜色进行叠加则完成图
像融合. 图像融合示意图, 如图 20 所示.

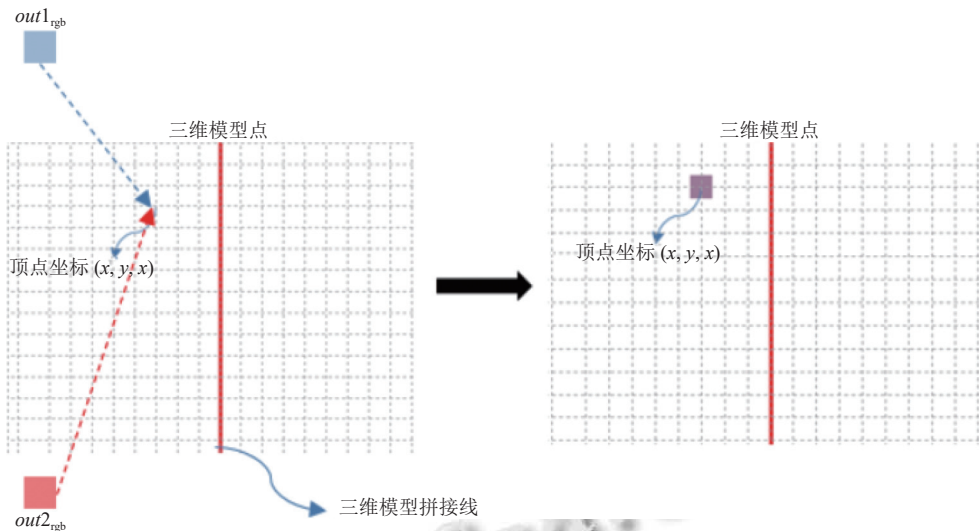


图 20 图像融合示意图

不做图像融合的数据格式为 $\{X, Y, Z, U, V\}$; 图像融合区的数据格式为 $\{X, Y, Z, U_1, V_1, U_2, V_2, Alpha\}$. 纹理坐标的计算只需在程序启动时计算一遍, 并将坐标点与纹理坐标按照数据格式组合. 将组合好的数据拷贝到 GPU 缓存中供渲染时使用.

由于 Lens Shading 问题导致图像中间亮, 边缘逐渐变黑. 因此需要为边缘增加亮度, 防止拼接区域出现黑色细线. Lens Shading 问题补偿公式如下:

$$\begin{cases} \ell = c_x^2 + c_y^2 \\ \ell_1 = (U_1 - c_x)^2 + (V_1 - c_y)^2 \\ \ell_2 = (U_2 - c_x)^2 + (V_2 - c_y)^2 \\ tex1_{rgb} = tex1_{rgb} + a \times (\ell_1 / \ell) \\ tex2_{rgb} = tex2_{rgb} + a \times (\ell_2 / \ell) \end{cases} \quad (9)$$

其中, a 为常量.

4 透明底盘功能的优化策略

4.1 透明底盘功能概述

透明底盘功能是将车底盲区进行填补, 让驾驶员可以清楚地了解到汽车底部的情况. 鱼眼镜头无法照射到汽车底部的物体, 所以传统的方案就无法实现透明底盘功能. 但是, 汽车运动起来后, 前几帧鱼眼图像存在当前帧车底图像信息. 所以, 透明底盘功能可以利用前一帧部分图像信息填充当前帧汽车底部盲区.

透明底盘功能实现过程, 首先预测汽车的行驶轨迹, 计算出汽车底部长方形 4 个点在上一帧鸟瞰图中

的纹理坐标. 其次, 用 4 个点作为顶点、上一帧鸟瞰图作为纹理和求得的纹理坐标进行渲染. 透明底盘功能渲染示意图如图 21 所示.

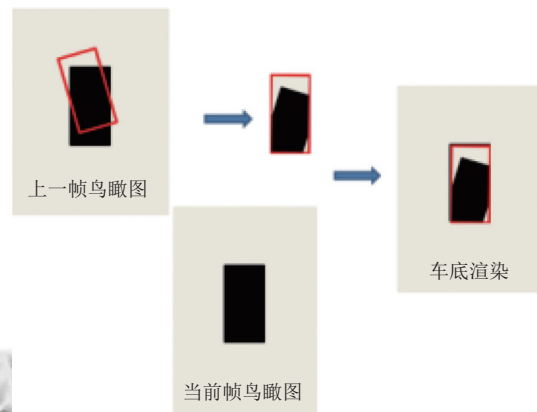


图 21 透明底盘功能渲染示意图

4.2 自行车模型优化与透明底盘算法实现

车辆运动学模型与动力学模型的建立是出于车辆运动的规划与控制考虑的. 此模型广泛应用于自动驾驶汽车轨迹规划模块中^[38,39]. 该模型可以精准的推测汽车的行驶轨迹.

自行车模型需做如下假设:

- (1) 车辆只在二维平面上做运动, 即只在 XY 水平面的运动.
- (2) 车的两侧车轮转角一致. 即将车模从两侧压扁为二维.
- (3) 车辆行驶速度变化缓慢, 忽略前后轴载荷的转移.

(4) 车身及悬架系统是刚性的。
假设结果如图 22 所示。

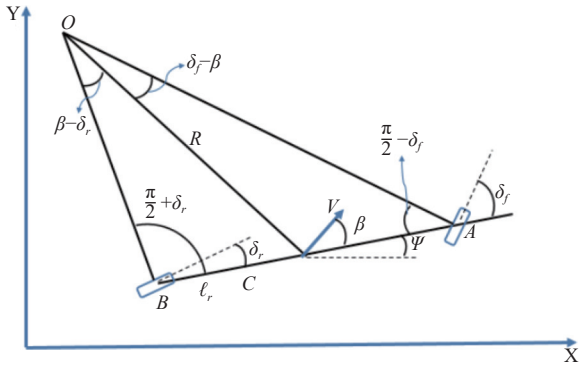


图 22 自行车模型

其中, O 为 OA 、 OB 的交点, 是车辆的瞬时转动中心, 线段 OA 、 OB 分别垂直于两个滚动轮的方向; β 为车辆速度方向和车身朝向两者间所成的角度, ψ 为车身与 X 轴的夹角. V 为质心速度; R 为转向半径; l_r 为后悬长度; l_f 为前悬长度; δ_r 为后轮偏角; δ_f 为前轮偏角. 自行车模型的动力学关系为:

$$\begin{cases} \frac{\sin(\delta_f - \beta)}{l_f} = \frac{\sin(\frac{\pi}{2} - \delta_f)}{R} \\ \frac{\sin(\beta - \delta_r)}{l_r} = \frac{\sin(\frac{\pi}{2} + \delta_r)}{R} \end{cases} \quad (10)$$

联立可得:

$$\begin{cases} \beta = \tan^{-1} \left(\frac{l_f \tan \delta_r + l_r \tan \delta_f}{l_f + l_r} \right) \\ (\tan \delta_f - \tan \delta_r) \cos \beta = \frac{l_r + l_f}{R} \end{cases} \quad (11)$$

当速度很慢时, 车辆的转弯半径几乎不变, 此时可以假设车辆的方向变化率与车辆的角速度相等, 则车辆的角速度为:

$$\dot{\psi} = \frac{V}{R} \quad (12)$$

在惯性坐标系 XY 下, 可得车辆运动学模型:

$$\begin{cases} \dot{X} = V \cos(\Psi + \beta) \\ \dot{Y} = V \sin(\Psi + \beta) \\ \dot{\Psi} = \frac{V \cos \beta}{l_r + l_f} (\tan \delta_f - \tan \delta_r) \end{cases} \quad (13)$$

其中, 有 3 个输入: δ_f 、 δ_r 、 V . 汽车的转向有两种模式^[40], 分为前轮转向与后轮转向两种类型. 当车辆为前轮转向时, 则车辆只有前轮转角 δ_f , 后轮转角 $\delta_r = 0$. 因此可以假设前轮转向自行车模型如图 23 所示.

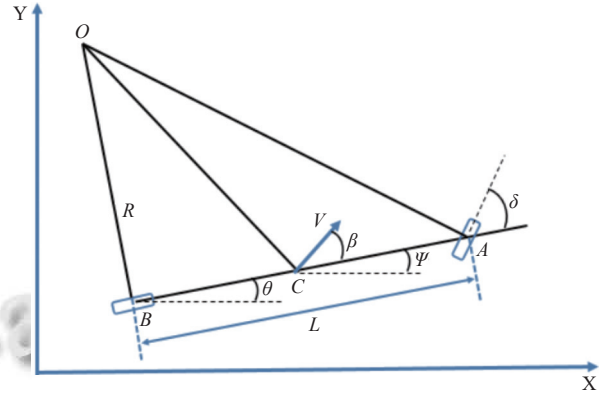


图 23 前轮转向自行车模型

前轮转向自行车模型总结规律. 由正弦法则求得车辆旋转半径 R , 计算公式如下:

$$R = \frac{L}{\tan \delta} \quad (14)$$

前轮转向自行车模型如下:

$$\begin{cases} \dot{X} = \frac{dX}{dt} = V \times \cos \theta \\ \dot{Y} = \frac{dY}{dt} = V \times \sin \theta \\ \dot{\theta} = \frac{d\theta}{dt} = \frac{V}{R} \end{cases} \quad (15)$$

此方程, 每一帧都需要计算, 为了能够更少计算量, 简化该计算公式. 由于单帧耗时时长很短、汽车速度很慢并且汽车电器件传入渲染引擎的数据本身就存在很大误差, 所以采用离散的方法. 改进自行车模型公式如下:

$$\begin{cases} t = t_1 - t_0 \\ S = \frac{(V_0 + V_t) \times t}{2} \\ \theta = (\theta_0 + \theta_t) / 2 \\ R = L / \tan \theta \\ d_x = R \times \sin \theta \\ d_y = R - R \times \cos \theta \\ d_\theta = S / R \end{cases} \quad (16)$$

式 (16) 中各参数, 如表 2 所示.

表2 各参数及意义

参数符号	参数意义	参数符号	参数意义
t_1	当前帧摄像头时间 (s)	t_0	上一帧摄像头时间 (s)
V_t	当前帧汽车速度 (m/s)	V_0	上一帧汽车速度 (m/s)
θ_t	当前帧车轮角度	θ_0	上一帧车轮角度
L	汽车轮间距	R	转弯半径
d_x	X轴方向移动距离	d_y	Y轴方向移动距离
d_θ	汽车旋转的角度		

利用 d_x 与 d_y 可以求得位移矩阵 $trans$; 利用 d_θ 可以求得Z轴旋转矩阵 $rota_z$. 当前帧汽车坐标系下的点, 在经历一帧图像时间后, 在上一帧汽车坐标系下位置的计算公式如下:

$$\begin{pmatrix} X_{car1} \\ Y_{car1} \\ Z_{car1} \\ k \end{pmatrix} = trans \times rota_z \times \begin{pmatrix} X_{car} \\ Y_{car} \\ Z_{car} \\ 1 \end{pmatrix} \quad (17)$$

其中, $(X_{car}, Y_{car}, Z_{car})$ 为当前帧汽车坐标系的坐标点; $(X_{car1}, Y_{car1}, Z_{car1})$ 为坐标点在上一帧汽车坐标系的位置; k 为常量.

结合鸟瞰图的相机矩阵, 就可以求得在鸟瞰图中的纹理坐标 (U, V) . 计算过程如式 (18):

$$\begin{cases} model = \text{鸟瞰图的相机矩阵} \\ \begin{pmatrix} x \\ y \\ z \\ k \end{pmatrix} = model \times \begin{pmatrix} X_{car1} \\ Y_{car1} \\ Z_{car1} \\ 1 \end{pmatrix} \\ \begin{pmatrix} U \\ V \end{pmatrix} = \begin{pmatrix} 0.5 \times x + 0.5 \\ 0.5 \times y + 0.5 \end{pmatrix} \end{cases} \quad (18)$$

4.3 方向盘角度到轮转角精度问题的优化

消息中间件传送到渲染引擎的角度是方向盘角度, 而计算公式需要的是汽车的轮转角. 所以需要一步方向盘角度到轮转角的操作. 如果采用正比例关系, 当方向盘角度比较小时, 不会有明显问题. 但当方向盘角度特别大时, 随着时间的积累, 物体预测位置就会存在误差. 为此提出查表法来解决此问题.

经过多次的测试得到方向盘角度与轮转角的原始数据, 并对数据进行处理, 得到一组很准确的方向盘角度与轮转角关系对应表, 如图 24 所示.

图 24 中可以看出当方向盘角度较小时, 方向盘角度与轮转角, 基本呈正比例关系. 但当方向盘角度过大时, 方向盘角度与轮转角, 不再成正比例关系. 这也印

证了最初的猜想. 通过查表找到方向盘角度所对应的轮转角. 方向盘角度位于两项数据之间的, 将这两项数据对应的轮转角, 做加权平均, 从而求得更为准确的轮转角. 如图 25 所示.

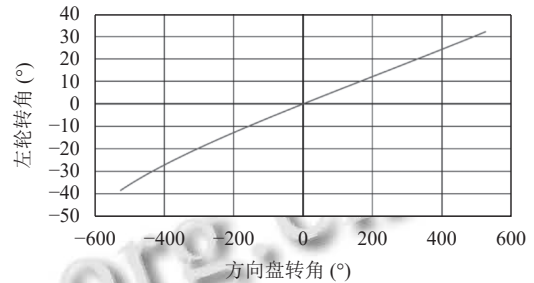


图 24 方向盘角度与轮转角关系曲线



图 25 轮转角优化效果对比图

4.4 透明底盘功能拼接缝优化策略

通过观察, 发现渲染图像中汽车周围与汽车底部拼接缝极其明显, 并且汽车底部像格栅一样. 拼接缝与拼接格栅问题示意图, 如图 26 所示.



图 26 拼接缝与拼接格栅问题示意图

汽车底部的格栅化, 是由于透明底盘功能拼接缝明显. 透明底盘的渲染方案会导致明显的拼接缝在车底积累, 形成格栅. 所以, 优化掉拼接缝问题, 格栅问题

也将解决。

世界坐标系位置相同的点,不同帧间亮度可能存在不同,并且车辆动态模型对行驶轨迹的推测存在误差,导致透明底盘功能拼接缝明显。

解决此问题,采用分层渲染的方法。将底部长方形周围区域在上一帧顶视图的图像,渲染在当前帧图像渲染结果的上层。首先找到底部长方形外侧适量圈数坐标点。其次利用自行车模型计算出坐标点对应的纹理坐标。最后,Alpha值由长方形边缘向外逐渐由1变为0。输出像素四通道的计算公式如下:

$$out_{rgba} = (tex_{rgb}, (l/L)) \quad (19)$$

其中, out_{rgba} 为渲染输出值; L 为向外选择点的圈数; l 为当前点所在圈数。透明底盘拼接缝优化效果图,如图27所示。

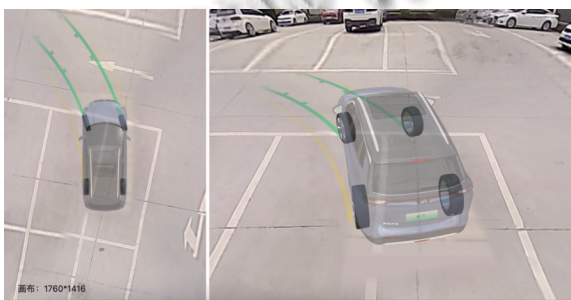


图27 透明底盘拼接缝优化效果图

5 结论与展望

(1) 为了方便展示,所有三维模型都进行了稀疏化处理。实际工程应用的拼接区三维模型共有36000个面和18848个点。

(2) 四路鱼眼镜头图像进入系统后进行了亮度均衡处理。亮度均衡不是本论文的研究内容,不展开讨论。实验所使用的亮度均衡算法主要借鉴于文献[41]。

(3) 在TDA4为核心的开发板,上车进行测试,实验结果表明,在输出图像分辨率为1440×720,左右视图均为3D模式时,可以达到30帧,CPU占用率低于20%。

本文基于距离的Alpha图像拼接算法,优化了三维模型、改进了透明底盘功能。该技术可有效提高渲染效果,减少计算复杂度。该研究技术对于车载全息透明影像有很大意义。该系统仍然存在许多问题,例如近处物体畸变严重、在拼接处存在物体消失等问题。相信这些问题日后都会得到解决。

参考文献

- 1 中华人民共和国国家统计局. 中国统计年鉴 2021. 北京: 中国统计出版社, 2021.
- 2 乐丹怡, 吴金洪, 李秋苓, 等. 汽车视觉盲区预测及解决方案. 科技创新导报, 2011, (20): 232. [doi: 10.3969/j.issn.1674-098X.2011.20.197]
- 3 徐丽丽, 张兴强. 道路交通事故中道路条件因素影响分析. 道路交通与安全, 2005, (1): 35-38.
- 4 么丽欣. 自主品牌驾驶辅助系统应用分析和发展建议. 汽车文摘, 2020, (10): 30-33. [doi: 10.19822/j.cnki.1671-6329.20200140]
- 5 Kato K, Suzuki M, Fujita Y, *et al.* Image synthesis display method and apparatus for vehicle camera: US, 7139412. 2006-11-21.
- 6 Xu Y, Zuo XY, Cheng AY, *et al.* Research and implementation of a vehicle top-view system based on i.MX6Q. Automotive, Mechanical and Electrical Engineering. London: CRC Press, 2017. 401-404.
- 7 Shimizu S, Kawai J, Yamada H. Wraparound view system for motor vehicles. Fujitsu Scientific & Technical Journal, 2010, 46(1): 95-102.
- 8 富士通半导体. 汽车360°多视角立体全景视频成像技术. 中国电子商情(基础电子), 2014, (9): 44-45.
- 9 Nobori K, Ukita N, Hagita N. A surround view image generation method with low distortion for vehicle camera systems using a composite projection. Proceedings of the 15th IAPR International Conference on Machine Vision Applications (MVA). Nagoya: IEEE, 2017. 386-389.
- 10 Sung K, Lee J, An J, *et al.* Development of image synthesis algorithm with multi-camera. Proceedings of the IEEE 75th Vehicular Technology Conference (VTC Spring). Yokohama: IEEE, 2012. 1-5.
- 11 Lee K, Kim D, Do H, *et al.* Effective photometric alignment for surround view monitoring system. Proceedings of the IEEE 9th International Conference on Consumer Electronics (ICCE-Berlin). Berlin: IEEE, 2019. 390-391.
- 12 华为技术有限公司. 一种全景图像的生成方法、车载图像处理装置及车辆: 中国, 202180001139.6. 2021-08-24.
- 13 北京百度网讯科技有限公司. 车载环视模型的调整方法、装置、电子设备和存储介质: 中国, 202111000386.2. 2021-11-23.
- 14 广州汽车集团股份有限公司. 一种车载环视摄像头供电电路及车载环视摄像头系统: 中国, 202022248706.3. 2021-06-04.
- 15 Gao Y, Lin CY, Zhao Y, *et al.* 3-D surround view for advanced driver assistance systems. IEEE Transactions on

- Intelligent Transportation Systems, 2018, 19(1): 320–328. [doi: [10.1109/TITS.2017.2750087](https://doi.org/10.1109/TITS.2017.2750087)]
- 16 Zhang L, Chen JT, Liu DY, *et al.* Seamless 3D surround view with a novel burger model. Proceedings of 2019 IEEE International Conference on Image Processing (ICIP). Taipei: IEEE, 2019. 4150–4154.
- 17 Zhang TJ, Zhang L, Shen Y, *et al.* OECS: Towards online extrinsics correction for the surround-view system. Proceedings of 2020 IEEE International Conference on Multimedia and Expo (ICME). London: IEEE, 2020. 1–6.
- 18 Zhang BY, Appia V, Pekkucuksen I, *et al.* A surround view camera solution for embedded systems. Proceedings of 2014 IEEE Conference on Computer Vision and Pattern Recognition Workshops. Columbus: IEEE, 2014. 676–681.
- 19 Liu YC, Zhang BY. Photometric alignment for surround view camera system. Proceedings of 2014 IEEE International Conference on Image Processing (ICIP). Paris: IEEE, 2014. 1827–1831.
- 20 Brown LG. A survey of image registration techniques. ACM Computing Surveys, 1992, 24(4): 325–376. [doi: [10.1145/146370.146374](https://doi.org/10.1145/146370.146374)]
- 21 Zitová B, Flusser J. Image registration methods: A survey. Image and Vision Computing, 2003, 21(11): 977–1000. [doi: [10.1016/S0262-8856\(03\)00137-9](https://doi.org/10.1016/S0262-8856(03)00137-9)]
- 22 Szeliski R. Image alignment and stitching: A tutorial. Foundations and Trends® in Computer Graphics and Vision, 2007, 2(1): 1–104.
- 23 周宝通. 车载3D全景辅助驾驶系统关键技术研究及应用 [硕士学位论文]. 北京: 北方工业大学, 2017.
- 24 何惠洋. 车载全景环视系统中的图像拼接技术研究 [硕士学位论文]. 西安: 西安工业大学, 2020.
- 25 栾婧. 基于鱼眼摄像头的汽车环视技术研究 [硕士学位论文]. 长春: 吉林大学, 2018.
- 26 Carr M. Visualization with OpenGL: 3D made easy. IEEE Antennas and Propagation Magazine, 1997, 39(4): 116–120. [doi: [10.1109/74.633000](https://doi.org/10.1109/74.633000)]
- 27 Yang Z, Zhao Y, Hu X, *et al.* A flexible vehicle surround view camera system by central-around coordinate mapping model. Multimedia Tools and Applications, 2019, 78(9): 11983–12006. [doi: [10.1007/s11042-018-6744-4](https://doi.org/10.1007/s11042-018-6744-4)]
- 28 Heng L, Li B, Pollefeys M. CamOdoCal: Automatic intrinsic and extrinsic calibration of a rig with multiple generic cameras and odometry. Proceedings of 2013 IEEE/RSJ International Conference on Intelligent Robots and Systems. Tokyo: IEEE, 2013. 1793–1800.
- 29 Ying XH, Hu ZY. Can we consider central catadioptric cameras and fisheye cameras within a unified imaging model. Proceedings of the 8th European Conference on Computer Vision. Prague: Springer, 2004. 442–455.
- 30 Zhang Z. A flexible new technique for camera calibration. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2000, 22(11): 1330–1334. [doi: [10.1109/34.888718](https://doi.org/10.1109/34.888718)]
- 31 陈佳兴. 基于嵌入式技术的三维全景环视系统研究 [硕士学位论文]. 天津: 天津工业大学, 2018.
- 32 冯闪闪. 汽车三维全景视觉辅助系统研究 [硕士学位论文]. 苏州: 苏州大学, 2019.
- 33 李鸿鹏. 基于鱼镜头的全景辅助驾驶系统研究. 光电技术应用, 2018, 33(3): 10–12, 16. [doi: [10.3969/j.issn.1673-1255.2018.03.003](https://doi.org/10.3969/j.issn.1673-1255.2018.03.003)]
- 34 冯聪. 360°车载环视系统图像拼接技术研究 [硕士学位论文]. 哈尔滨: 哈尔滨工业大学, 2018.
- 35 Lin CC, Wang MS. A vision based top-view transformation model for a vehicle parking assistant. Sensors, 2012, 12(4): 4431–4446. [doi: [10.3390/s120404431](https://doi.org/10.3390/s120404431)]
- 36 Baek I, Kanda A, Tai TC, *et al.* Thin-plate spline-based adaptive 3D surround view. Proceedings of 2019 IEEE Intelligent Vehicles Symposium (IV). Paris: IEEE, 2019. 586–593.
- 37 刘冬, 秦瑞, 陈曦, 等. 3D车载环视全景生成方法. 计算机科学, 2017, 44(4): 302–305. [doi: [10.11896/j.issn.1002-137X.2017.04.061](https://doi.org/10.11896/j.issn.1002-137X.2017.04.061)]
- 38 Yang H, Rakha H, Ala MV. Eco-cooperative adaptive cruise control at signalized intersections considering queue effects. IEEE Transactions on Intelligent Transportation Systems, 2016, 18(6): 1575–1585.
- 39 杨彬, 宋学伟, 高振海. 考虑车辆运动约束的最优避障轨迹规划算法. 汽车工程, 2021, 43(4): 562–570.
- 40 李双, 刚宪约, 于海兴. 汽车操纵动力学三自由度模型与转向特性仿真. 机械设计与制造, 2015, (3): 260–264. [doi: [10.3969/j.issn.1001-3997.2015.03.070](https://doi.org/10.3969/j.issn.1001-3997.2015.03.070)]
- 41 谢绍川. 一种汽车全景环视影像匀色方法. 电子技术与软件工程, 2021, (13): 131–134.

(校对责编: 牛欣悦)