

## 拼接多幅鱼眼图像，产生全景视图

本白皮书讨论 Altera 和 Manipal Dot Net 开发的创新体系结构，利用 FPGA 拼接多幅鱼眼图像，产生全景视图。这一体系结构为驾驶员提供了车辆周围的全景图像，帮助他们更轻松的驾驶车辆。

### 引言

鱼镜头具有超宽视角特性，性价比较高，在汽车成像系统中得到了越来越广泛的应用。一种应用是呈现车辆周围 360° 全景图像，通过将鱼镜头阵列采集到的多幅图像拼接起来实现这一功能。这为驾驶员提供了车辆周围的全景图像，帮助他们更轻松的驾驶车辆。

由于鱼镜头 (1) 具有非常宽的视角，因此，只需要很少的镜头就能够产生全景视图。但是，正面的半球景象被映射到平面上之后，鱼眼图像会产生严重的失真。由于存在这种失真，拼接多幅鱼眼图像是非常繁琐的任务，涉及到进行大量的计算和图像处理。一般而言，很难解决这一问题，因此，需要做一些近似和简化假设才具有可行性。实际应用假设，景象中被拼接的目标距离镜头足够远，可以忽略立体差异。

### 鱼眼图像产生全景视图

鱼眼透镜将远景（广角）图像全部映射到非鱼眼透镜上，而不是部分映射，从而实现了非常宽的视场 (FOV)。早期的工作 (2) 产生不同的鱼眼映射（例如，线性投影映射），开发灵活的体系结构校正远景鱼眼图像。鱼眼映射导致的径向失真是指图像放大率随光轴的距离而减小。视在效应也被称为“桶形失真”，是指图像 (图 1 所示) 映射在球面上。结果，鱼眼图像未能保留广角图像最重要的特性，即，将场景中的直线映射为图像中的直线 (3)。

图 1. 鱼眼图像



全景视图的特点是具有非常大的水平 FOV，通常用于产生 360° 场景，例如，城市轮廓等。鱼眼图像具有较大的 FOV，非常适合产生全景视图。然而，从鱼眼图像中产生全景视图并不是简单的校正图像来获得相应的远景图像。

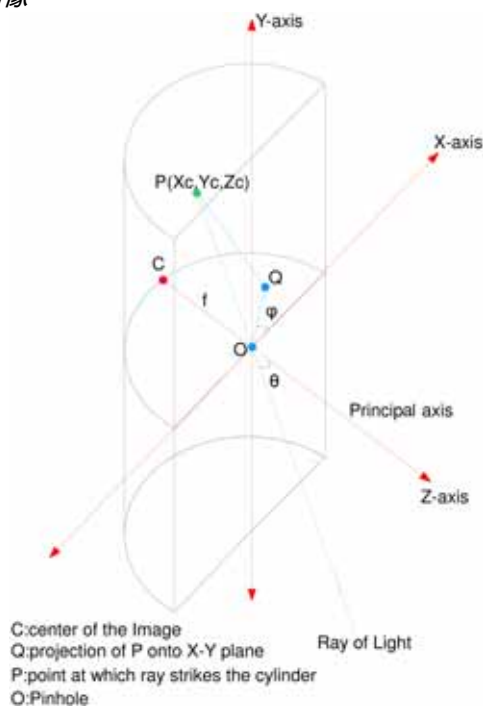
实际上，由于远景映射，这类图像的边沿被拉伸，远景图像并不是全景的。这种拉伸被称为全景失真，距离主轴越远，失真就越严重。因此，从远景图像中产生全景视图需要将具有较小 FOV (以降低远景失真)

的几幅远景图像沿水平轴拼接起来。产生全景视图的一种方法是将鱼眼图像校正为远景图像，然后进行拼接。但是，由于远景图像本身具有较小的 FOV，因此，这一解决方案不能发挥鱼眼镜头的广角特性优势。

另一种解决方法是直接拼接鱼眼图像，产生远景视图。而这需要进行特殊的校正处理，不同于将鱼眼图像校正为远景图像的方法。全景视图的一些特性要求采用特殊的校正方法。虽然全景视图有较大的水平 FOV，其垂直 FOV 并不大。因此，为能够产生全景视图，校正鱼眼图像的方法应在水平方向有很小甚至没有远景失真，而在垂直方向有适度的远景失真。采用下面的抽象方法可以获得比较精确的公式。

当光线进入针孔镜头，投射到半径等于镜头焦距的球面上时，就会形成鱼眼图像。另一方面，当光线投射到与针孔的距离等于镜头焦距的平面上时，就会形成远景图像。为校正鱼眼图像，在水平方向消除远景失真，那么，入射表面在垂直方向必须是环形的。由于在垂直方向允许有一定的远景失真，因此，入射表面在垂直方向可以是平面。垂直入射，并且半径等于焦距的柱面入射表面能够满足这些要求。通过“解开”柱面（称为合成表面）上形成的图像，将其与其他鱼眼镜头呈现的相似图像拼接起来，可以获得全景视图。

图2. 投射到柱面，产生全景图像



假设镜头主轴在 z 轴方向，水平和垂直方向分别对应 x 和 y 轴。光线进入镜头，与主轴夹角为  $\theta$ ，投射到 xy 平面，与 x 轴成  $\varphi$  角，如图 2 所示。通过针孔时，光线投射到柱面上，其半径等于焦距  $f$ ，光轴在 y 轴上。 $(x_c, y_c, z_c)$  是入射光线打到柱面上时的坐标点。因此， $(x_c, y_c, z_c)$  点与  $(f, \theta, \varphi)$  的关系如方程 (1)、方程 (2) 和方程 (3) 所示。

$$x_c = \frac{f \tan \theta \cos \varphi}{\sqrt{1 + \tan^2 \theta}} \quad (1)$$

$$y_c = \frac{f}{\sqrt{1 + \tan^2 \theta \cos^2 \varphi}} \quad (2)$$

$$z_c = \frac{f \tan \theta \sin \varphi}{\sqrt{1 + \tan^2 \theta \cos^2 \varphi}} \quad (3)$$

全景（未解开）图像  $(x_q, y_q)$  相对于  $(x_c, y_c, z_c)$  的坐标由方程 (4) 和方程 (5) 给出。

$$x_q = f \tan^{-1}\left(\frac{x_c}{z_c}\right) = f \tan^{-1}(\tan \theta \cos \varphi) \quad (4)$$

$$y_q = y_c = \frac{f \tan \theta \sin \varphi}{\sqrt{1 + \tan^2 \theta \cos^2 \varphi}} \quad (5)$$

假设镜头映射函数采用线性映射，鱼眼图像  $(x_f, y_f)$  相对于  $(x_c, y_c, z_c)$  的坐标由方程 (6) 和方程 (7) 给出。

$$x_f = f \theta \cos \varphi \quad (6)$$

$$y_f = f \theta \sin \varphi \quad (7)$$

从方程 (4)、方程 (5)、方程 (6) 和方程 (7) 中消掉  $\theta$  和  $\varphi$ ，实现校正。方程 (8) 和方程 (9) 表述了全景视图和鱼眼图像之间的相对坐标关系。

$$x_f = f \left[ \tan^{-1} \left\{ \frac{\sqrt{\left(\frac{y_q}{f}\right)^2 + \sin^2\left(\frac{x_q}{f}\right)}}{\cos\left(\frac{x_q}{f}\right)} \right\} \right] \frac{\sin\left(\frac{x_q}{f}\right)}{\sqrt{\left(\frac{y_q}{f}\right)^2 + \sin^2\left(\frac{x_q}{f}\right)}} \quad (8)$$

$$y_f = f \left[ \tan^{-1} \left\{ \frac{\sqrt{\left(\frac{y_q}{f}\right)^2 + \sin^2\left(\frac{x_q}{f}\right)}}{\cos\left(\frac{x_q}{f}\right)} \right\} \right] \frac{\frac{y_q}{f}}{\sqrt{\left(\frac{y_q}{f}\right)^2 + \sin^2\left(\frac{x_q}{f}\right)}} \quad (9)$$

## 使用 LUT 拼接鱼眼图像

利用方程 (8) 和方程 (9)，设计人员能够将校正图像上的每一像素点映射到输入图像的每一像素点。校正后的鱼眼图像在水平方向没有全景失真。采用环形鱼眼图像，这些方程足以产生  $180^\circ$  FOV 的全景视图。但是，如果鱼眼图像不是环形的，或者要求全景视图的 FOV 大于  $180^\circ$ ，那么，必须将两幅以上的这类图像拼接起来。

拼接图像时，需要对图像进行配准，以确定重叠区。图像必须来自两个以上相同的镜头，其主轴都位于同一水平面。而且，镜头必须分开一点距离，彼此有些转动。假设景象中的所有目标都足够远，这种情况可以建模为一个镜头沿垂直轴旋转得到不同的图像。在这些假设下，方程 (8) 和方程 (9) 表示的校正功能将图像（将其映射到柱面上）变换为彼此水平平移。校正后，只需要指出图像之间对准时的水平移位就能够解决图像配准问题。

但是，由于镜头对准误差，实际中，只有水平方向对准还不能完全配准图像。因此，除了水平平移，还需要进行垂直平移才能完全对准。一种基于相似度的方法 (4) 确定了图像彼此对准所需要的配准参数（水平和垂直平移）。

确定了经过校正的鱼眼图像的配准参数后，方程 (8) 和方程 (9) 采用水平和垂直平移常数，适当的平移  $(x_q, y_q)$ ，将输出拼接图像的每一像素映射到多幅输入图像的一幅中。很明显，这种映射方法与图像内容无关，而取决于镜头的特性（包括 FOV 和输入图像分辨率）、显示（包括分辨率），以及配准参数等。因此，可以通过在启动时进行一次计算，结果存储为查找表 (LUT)，实现映射。

拼接鱼眼图像时，采用以下 4 个步骤来计算 LUT：

1. 使用方程 (8) 和方程 (9)，离线计算校正一幅图像的 LUT (它不需要知道配准参数信息)，存放在存储器中。
2. 系统启动时，系统中的所有镜头采集图像，使用存储的 LUT 进行校正。
3. 配准校正后的图像，计算配准参数。
4. 使用配准参数和存储的 LUT，产生用于直接拼接鱼眼图像的最终 LUT。

1 注意，虽然方程 (8) 和方程 (9) 产生实数输入像素位置，但是，并不需要 LUT 来存储浮点值。这是因为采用了 9 点插值方法，支持 FPGA 所需要的高效像素插值和定点表示。

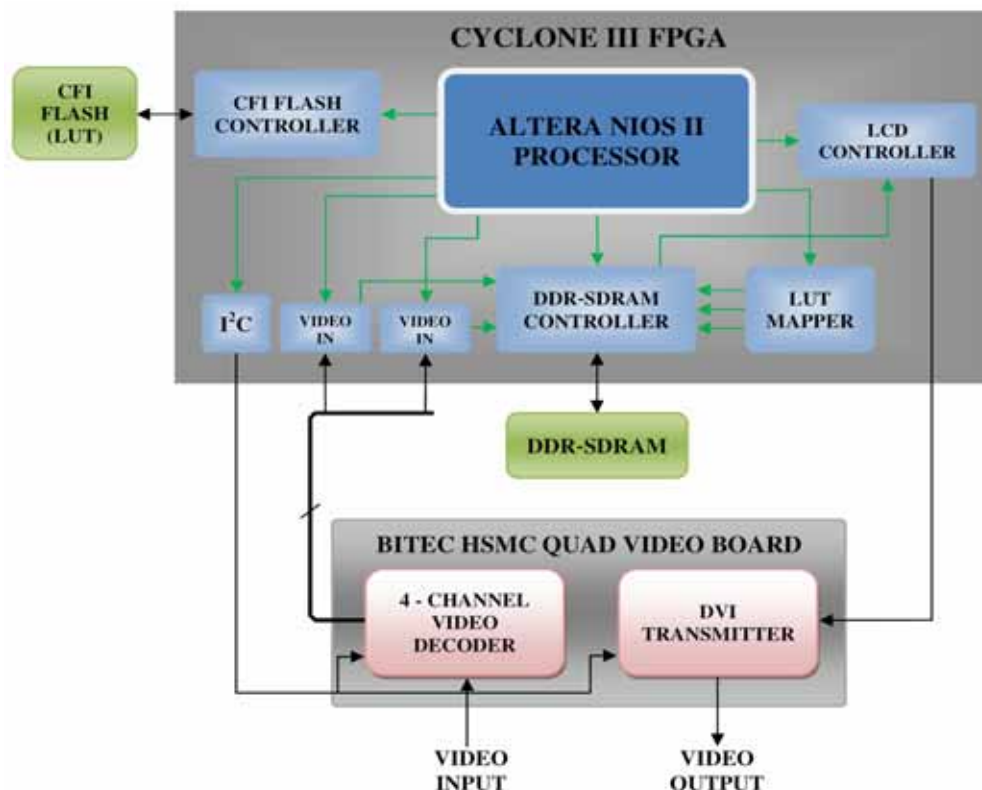
f Altera 白皮书 “采用灵活的体系结构实现汽车后视镜镜头鱼眼校正” 详细介绍了这些内容 (2)。

### 设计实现

实现鱼眼图像拼接理想的方法是使用含有 Altera® Cyclone® FPGA 系列和 Altera Nios® II 嵌入式处理器的系统。Nios II 体系结构是 RISC 软核结构，全部在 Altera FPGA 可编程逻辑和存储器模块中实现，能够处理从 DSP 到系统控制的多种嵌入式计算应用。系统设计人员利用 Nios II 处理器的软核 IP 特性，针对特殊应用需求，设置并产生定制 Nios II 内核。采用 Altera 的 Nios II 嵌入式评估套件 (NEEK) 作为开发平台。

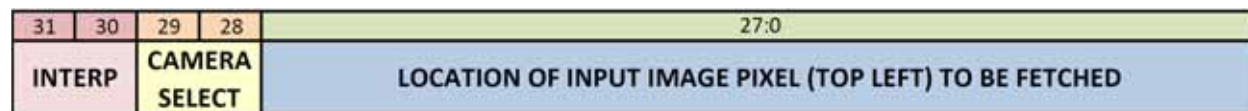
FPGA 设计体系结构基于 Nios II 软核嵌入式处理器、Bitec 四路视频输入模块、I<sup>2</sup>C 配置模块、定制 LUT 映射器硬件 IP、DDR-SDRAM 控制器和 LCD 控制器，如图 3 所示。

图 3. 系统结构图



在这一实现结构中，系统启动时，通过 Nios II 处理器读取存储在系统闪存中的校正 LUT，进行图像配准，计算配准参数，计算能够直接拼接多幅鱼眼图像的 LUT，显示全景视图。拼接 LUT 由含有其他信息的 32 位字 (图 4) 组成。插值位 30 和 31 表示要进行的插值类型，而镜头选择位 28 和 29 表示索引四个镜头中的哪一个。第 27 位到第 0 位表明，使用等于像素行列积的数值来获取图像中的某一像素。

图 4. LUT 输入

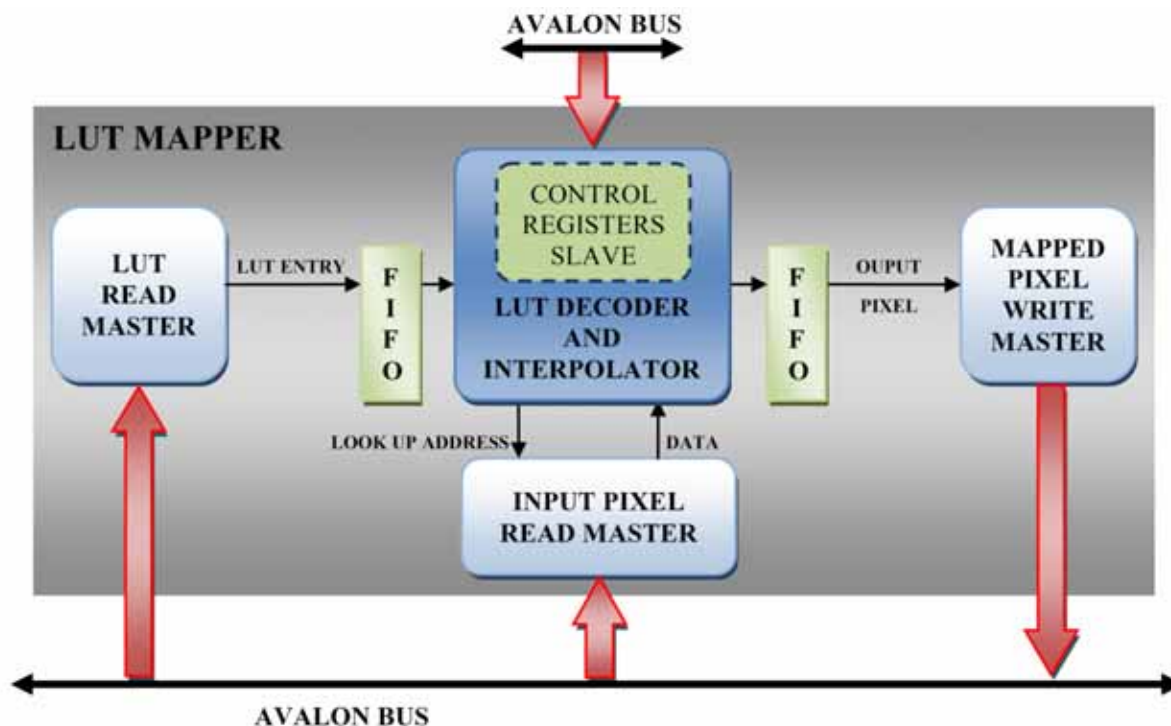


基于拼接 LUT 输入，重新计算输出像素 (使用 9 点插值法)，然后显示。一般采用 DDR-SRAM 作为存储输入图像的存储器。虽然 9 点插值并不需要进行大量的计算，但是，访问 DDR-SRAM 进行插值计算来恢复输入像素所需要的时间比较长。这是因为，Nios II 处理器并没有针对 DMA 类型的读写操作进行优化。此外，Nios II 处理器也不能充分利用 LUT 输入存储器访问的内在冗余特性。因此，设计了定制硬件 IP 模块来实现 LUT 映射和 9 点插值方法。

### LUT 映射器

使用 Altera Cyclone III FPGA 中不到 1500 个逻辑单元 (LE) 以及三个 M9K 存储器模块，LUT 映射器设计 (图 5) 包括两个 Avalon<sup>®</sup> 存储器映射 (MM) 读主机，一个 Avalon-MM 写主机和一个 Avalon-MM 从机。对于写主机要写入的每一像素，通过一个读主机来读取一个 LUT 输入。另一读主机读取输入图像的像素。从机配置控制寄存器，由它定义 LUT 映射器的工作。用户设计的 LUT 映射器 SOC 组件与 DMA 控制器相似，承担处理器的模块存储器复制工作，对所复制的数据不进行任何处理。

图 5. LUT 映射器组成



### LUT 读主机

LUT 读主机符合 Avalon-MM 读主机协议。其主要目标是顺序读取 LUT 输入，然后将其写入 FIFO 缓冲，不进行任何处理。该方法采用流水线，将来自 Avalon 总线的数据缓冲到 LUT 解码器中，从而提高了吞吐量。

### LUT 解码器和插值器

LUT 解码器和插值器模块从 FIFO 缓冲读取 LUT 输入，然后对其进行解码。例如，假设最多需要拼接四路镜头输入。指针指向存储器帧缓冲相对于每一镜头的基本地址，将其表示为 `frame_cam[camera]`，其中，`camera = 1、 2、 3 或者 4`。从 FIFO 缓冲读出的数据表示为 `lut_data`。然后，存储 LUT，使得：

```
interp = lut_data[31:30]
camera = lut_data[29:28]
offset = lut_data[27:0]
Let address = (frame_cam + offset)
Let the line width of each camera image be line_width.
```

从 FIFO 缓冲中读出 LUT 输入后，计算与 LUT 输入相应的输出像素，如表 1 所示。

表 1. 输出像素计算——9 点插值方法

| 插值 | 插值方法          | 输出像素值  |
|----|---------------|--|
| 0  | 没有插值，使用“真实”像素 | <code>[pixel_at (address)]</code>  |
| 1  | 2 像素垂直平均      | <code>[pixel_at (address) + pixel_at (address + line_width)] ÷ 2</code>  |
| 2  | 2 像素水平平均      | <code>[pixel_at (address) + pixel_at (address + 1)] ÷ 2</code>   |
| 3  | 4 像素水平和垂直平均   | <code>[pixel_at (address) + pixel_at (address + 1) + pixel_at (address + line_width) + pixel_at (address + line_width + 1)] ÷ 4</code> |

数据采用了 RGB565 格式 (16 位像素)，因此假设函数 `pixel_at( )` 中使用的指针指向 16 位数据类型。该模块据此计算像素值，将其写入到 FIFO 缓冲中。

### 输入像素读主机

该模块符合 Avalon-MM 读主机协议，支持流水线。LUT 解码器和插值器模块控制这一模块获取计算所需要的像素。获取的像素数量取决于插值比特。该模块读取数据的效率要高于 LUT 读主机 (`interp > 0` 时)，因此，该模块仲裁器优先级应高于 LUT 读主机仲裁器优先级。

### 映射像素写主机

该模块符合 Avalon-MM 写主机协议。映射像素写主机从 FIFO 缓冲读取像素 (由 LUT 解码器和插值器放置它)，然后顺序将其写入显示存储器。

### 设置镜头

在这个例子中，如图 6 所示，采用 OmniVision OV7950 镜头来产生输入视频流。每个鱼眼镜头都具有 120° 的 FOV，与主轴成 90° 角。

图 6. 设置镜头



## 结果

图 6 中描述的系统采集来自两个鱼眼镜头的视频。经过一次配准过程确定配准参数后，使用 LUT 映射器将得到的图像拼接起来，拼接后的图像显示在分辨率为 1024×768 的 LCD 屏上。使用最近邻居插值法，不采用任何合并算法，系统每秒大约处理 15 帧，进一步提高了速率。这一过程处理图 7 所示的输入帧，将其拼接在一起形成图 8 所示的全景图像。

图 7. 来自两个镜头的输入帧



图 8. 拼接输出帧



## 结论

使用 FPGA 和软核嵌入式处理器技术，Altera 联合 MDN 为鱼眼图像拼接产生全景视图开发了创新的体系结构。这一体系结构灵活、可更新，能够高效的使用 FPGA 资源。该体系结构的 Nios II 嵌入式处理器具有通用性，能够处理较多的功能，该技术非常适合汽车全景镜头等需要实现 360° 视图的应用。

## 参考

1. “鱼眼透镜”，Wikipedia：  
[http://en.wikipedia.org/wiki/Fisheye\\_lens](http://en.wikipedia.org/wiki/Fisheye_lens)
2. 采用灵活的体系结构实现汽车后视镜头鱼眼校正：  
[www.altera.com/literature/wp/wp-01073-flexible-architecture-fisheye-correction-automotive-rear-view-cameras.pdf](http://www.altera.com/literature/wp/wp-01073-flexible-architecture-fisheye-correction-automotive-rear-view-cameras.pdf)
3. 计算机图像变换和投影，David Salomon, Springer, 2006。
4. 图像拼接——对比和新技术，奥克兰大学技术报告，1998年10月：  
<http://cit.r.auckland.ac.nz/techreports/1998/CITR-TR-30.pdf>

## 致谢

- Judd Heape，高级技术营销经理，工业和汽车业务部，Altera 公司。
- Dave Elliott，高级营销经理，工业和汽车业务部，Altera 公司。
- MDN SOPC 组



**ALTERA**

101 Innovation Drive  
San Jose, CA 95134  
[www.altera.com](http://www.altera.com)

版权 © 2009 Altera 公司。保留所有版权。Altera、可编程解决方案公司、程式化 Altera 标识、专用器件名称和所有其他专有商标或者服务标记，除非特别声明，均为 Altera 公司在美国和其他国家的商标和服务标记。所有其他产品或者服务名称的所有权属于其各自持有人。Altera 产品受美国和其他国家多种专利、未决应用、模板著作权和版权的保护。Altera 保证当前规范下的半导体产品性能与 Altera 标准质保一致，但是保留对产品和服务在没有事先通知时的升级变更权利。除非与 Altera 公司的书面条款完全一致，否则 Altera 不承担由此处所述信息、产品或者服务导致的责任。Altera 建议客户在决定购买产品或者服务，以及确信任何公开信息之前，阅读 Altera 最新版的器件规范说明。