

«*In The Name Of GOD*»



دانشگاه صنعتی امیرکبیر  
(پلی تکنیک تهران)

## [Project-01-Report]

[3D COMPUTER VISION]

Hasan Masroor | [403131030] | November 12, 2025

# Project #1: Single-View AR Cube

## 1. Image Choice and Manual Annotations



تصویر انتخابی یک راهرو متقارن را نشان می‌دهد که کف آن با کاشی‌های شطرنجی پوشیده شده و الگوی پرسپکتیو واضحی را ایجاد می‌کند. این تصویر برای کالیبراسیون هندسی تکنما بسیار مناسب است، زیرا شامل دو دسته اصلی از خطوط موازی است که هر کدام به نقاط گزینه‌ای همگرا می‌شوند. سطح کف تقریباً مسطح است و می‌توان آن را به عنوان صفحه مرجع در مراحل بعدی در نظر گرفت. همچنین، الگوی منظم و واضح کاشی‌ها باعث می‌شود انتخاب نقاط و برآورد دقیق خطوط و نقاط گزینه با دقت بالاتری انجام گیرد.

در این مرحله، با کلیک دستی بر روی تصویر، ابتدا چهار گوشی ناحیه‌ی کف (به عنوان صفحه‌ی مرجع) و سپس هشت نقطه مربوط به چهار خط در دو جهت اصلی انتخاب شدند.

## 2. Vanishing Points, Horizon, and Intrinsics

در این بخش، هدف محاسبه‌ی نقاط گزینه دو جهت اصلی تصویر و تعیین خط افق بر اساس خطوط انتخاب شده در بخش قبل بود. ابتدا با استفاده از چهار خط موازی در دو جهت عمود برهم (فرمز و آبی)، معادلات خطوط در فضای تصویری محاسبه شد. سپس با به دست آوردن نقاط تقاطع هر جفت از خطوط هم‌جهت، دو نقطه‌ی گزینه استخراج گردید. اتصال این دو نقطه، خط افق تصویر را تشکیل داد که در تصویر نهایی با رنگ سیز نشان داده شده است. بر اساس اصل عمود بودن دو جهت در فضای سه‌بعدی، فاصله‌ی نقاط گزینه از مرکز تصویر برای برآورد تقریبی فاصله‌ی کانونی ( $f$ ) استفاده شد و همچنین ماتریس دورین ( $K$ ) شامل  $f$ ، مرکز تصویر تخمین زده شد.

Vanishing Point 1: (1532.99, 783.84)

Vanishing Point 2: (-790.32, 726.76)

focal length ( $f$ ): 1161.11 pixels

شکل زیر نتیجه‌ی نهایی این مرحله را نشان می‌دهد که در آن نقاط گزینه، خط افق و نقطه‌ی اصلی تصویر مشخص شده‌اند. این اطلاعات در مراحل بعدی برای تخمین هموگرافی و بازاری سه‌بعدی استفاده می‌شوند.



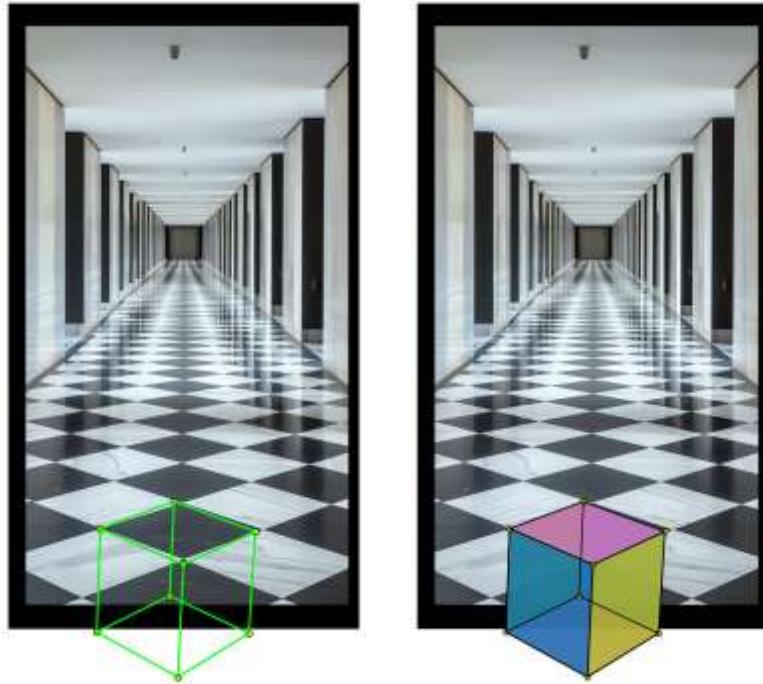
### 3. Planar Homography and Pose Recovery

در این مرحله، با استفاده از نقاط چهارتایی انتخاب شده روی کف صحنه و مختصات مرجع در صفحه جهانی، هموگرافی بین صفحه کف و تصویر با روش DLT نرمال شده محاسبه شد. سپس با بهره گیری از ماتریس درونی که در بخش قبل به دست آمده بود، مؤلفه های چرخش ( $R$ ) و انتقال ( $t$ ) دوربین نسبت به صفحه زمین استخراج شدند و پس از نرمال سازی بردارها و تصحیح جهت دستگاه مختصات، محور های دوربین روی تصویر اصلی ترسیم گردیدند تا موقعیت تقریبی و جهت دید دوربین به صورت بصری مشخص شود. این محورها با رنگ های قرمز، بنفش و آبی در خروجی زیر قابل مشاهده هستند:



## 4. Cube Definition, Projection, and Rendering

در این مرحله یک مکعب واحد با طول ضلع  $s=1$  روی صفحه مرجع تعریف شد. مختصات سه بعدی رأس‌ها با ماتریس تصویر  $P=K[R|t]$  به صفحه تصویر نگاشته و به صورت wireframe و سایه‌دار رسم شدند. نتیجه‌ی wireframe نشان‌دهنده‌ی تطابق درست پرسپکتیو مکعب با کف صحنه است. در حالت سایه‌دار، شدت روشنای با مدل بازتاب لامبرت و جهت نورا محاسبه شد. تمام رأس‌های مکعب دارای عمق مثبت بودند و از خط افق فاصله داشتند و در نتیجه پروژکشن پایدار و بدون ناپایدرای عددی بدست آمد. مکعب نهایی با پرسپکتیو صحیح در تصویر قرار گرفته و واقع‌گرایی قابل قبول ایجاد کرده است.



## 5. Face Shading Implementation

در این بخش، روشنای هر وجه مکعب بر اساس مدل بازتاب لامبرت (Lambertian Reflection) محاسبه شد تا ظاهر واقع‌گرایانه‌ای از نوردهی و سایه‌ها بر روی مکعب ایجاد شود. جهت منبع نور در مختصات دوربین به صورت  $[0,0,-1]=l$  در نظر گرفته شد، که بیانگر تابش نوری از رویه‌روی صحنه به سمت مکعب است. برای هر وجه، بردار نرمال  $n^8$  از حاصل ضرب برداری دو ضلع مجاور آن به دست آمد و سپس شدت روشنای با رابطه‌ی زیر محاسبه شد:  $I = \max(0.2, n^8 \cdot l)$  مقدار 0.2 به عنوان حداقل روشنای لحاظ گردید تا حتی سطوحی که در خلاف جهت نور قرار دارند کاملاً تاریک نشوند. در نهایت، رنگ نهایی هر وجه از حاصل ضرب شدت روشنای 1 در رنگ پایه‌ی آن وجه به دست آمد. فایل shading\_report.json شامل اطلاعات عددی هر وجه است که دربرگیرنده‌ی شاخص وجه، بردار نرمال، شدت روشنای و رنگ نهایی محاسبه شده می‌باشد. در ادامه تصویر خروجی و نمونه‌ای از این نتایج فایل را آورده ایم:

```
"face_index": 0,  
"face_name": "bottom",  
"vertices": [  
    0,  
    1,
```

```
    2
  ],
  "normal_vector": [
    -0.047015,
    0.98904,
    0.139963
  ],
  "brightness": 0.2,
  "base_color": [
    0.9,
    0.9,
    0.9
  ],
  "final_color": [
    0.1800000000000002,
    0.1800000000000002,
    0.1800000000000002
  ]
},
```

