

(Planar) 3-D model recognition

Siniša Kolarić

`<http://www.tecgraf.puc-rio.br/~skolaric>`

September 2006

Abstract

In this presentation, I present a way to recognize a (planar) 3-D model in a set of detected 2-D line segments.

NOTE. This is accompanying material to my Trabalho 2(a) for the course INF2064 "Tópicos de Computação Gráfica III - Realidade Aumentada e Cooperativa" held by prof. Marcelo Gattass.

The problem

Given a set S of 2-D line segments, can we make the computer to "see" a given 3-D model in that set S ?

In another words, from an arrangement of parts or elements (in this case, 2-D line segments), can we make the computer to detect (interpret, recognize) our planar 3-D model in that arrangement?

One solution using libraries IM and JV

RECOGNIZE()

```
1  while IMVIDEOCAPTUREFRAME(frame)
2      do 2D_lines_detected ← JVEDGEDetect(LOG(frame))
3          model_is_recognized ← JVINTERPRET(2D_lines_detected)
4          if model_is_recognized
5              then plot the (planar) 3-D model
6              else plot just 2D_lines_detected
```

We already know the inner workings of lines 1 and 2 from previous Trabalhos. Line 3 (function `jvInterpret()`) is explained in the following slide.

juInterpret()

- Defines constraints on the interpretation tree (that is, it limits the size of the search space by setting local constraints, like the parallelism of certain lines, or orthogonality of certain lines, and similar)
- Additionally checks the legality of the potential match, that is, a match that passed the first criteria above (local constraints), for example by finding out whether the match has its corresponding homography through `juFindTransf()`, and whether this homography gives an error that is too big
- Decides what gets plotted next: just the detected 2-D lines (means an interpretation hasn't been found), or the 3-D model (means an interpretation has been found)

fvInterpret()

- The callback `interpretation_function()` passed to `fvInterpret()` can be called many times (within one single call to `fvInterpret()`) by the function `BuscaEmArvore()`, since `BuscaEmArvore()` is a recursive DFS and calls itself many times
- `interpretation_function()` returns 1 if a specific interpretation F is satisfactory, and 0 otherwise

References

- *Three-dimensional computer vision*, by Olivier Faugeras
- *Introductory techniques for 3-D computer vision*, by Emanuele Trucco, Alessandro Verri
- WWW, various sites