Plane sweep stereo

Input: a set of views, [image, intrinsics, extrinsics].
Output: a depth map.
1. Choose the reference view.
2. Init the depth map with unknown depths and infinite costs.
3. for all depths do
4. Project each image onto the plane at depth d.
5. Compute per pixel cost values from the projections into a cost image.
6. Filter the cost image (blur).
7. for all pixels in the depth map do
8. Update the depth and cost values, if the cost in the current cost image is less than the stored in the depth map.
9. end for
10. end for

Software

Written in C++ for GNU Compiler Collection (GCC) and Linux. Implementation is available under the MIT-licence at: http://www2.it.lut.fi/project/rtmosaic/

Requirements

OpenGL 2.0
- Includes GL Shading Language.
- ARB_texture_float
  - Floating-point textures: half (16-bit) and single (32-bit) precision.
- ARB_texture_rg
  - One and two channel textures and render targets.
- ARB_pixel_buffer_object
  - Performance improvements: avoid a copy, and pipeline texture upload/download.
- NV_fence
  - Optional performance optimization: avoid busy-waiting in the (proprietary) GL driver.
In practise, requires the proprietary Nvidia or ATI drivers.

API

A single view with image and geometry:

```cpp
class GPUStereo {
  public:
    void configure(Config::Section &);
    void initGL(void);
    Image<FloatValue> *createDepthMap(
      std::vector<GPUStereoFrame> &views,
      int reference);
    void destroyDepthMap(Image<FloatValue> * &);
    void cleanupGL(void);
};
```

GPUStereo application programming interface:

1. Project to depth d, compute cost image
2. Filter
3. Update
4. Update depth map
5. Store

Demonstration

Crossed-eye Stereo Images

Fig. 3: Reconstruction from the Middlebury Reindeer.

Fig. 4: Reconstruction from the Middlebury Dolls.

Visual SLAM demo

The first version published in [1].

References