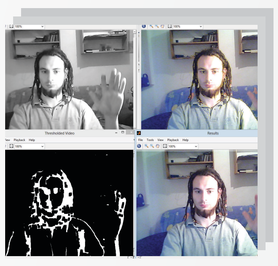
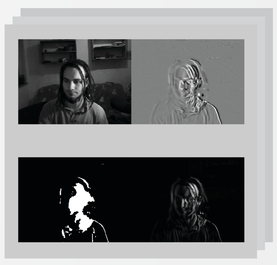
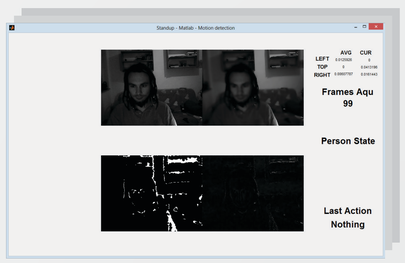
StandUp

Alarm Clock Vision





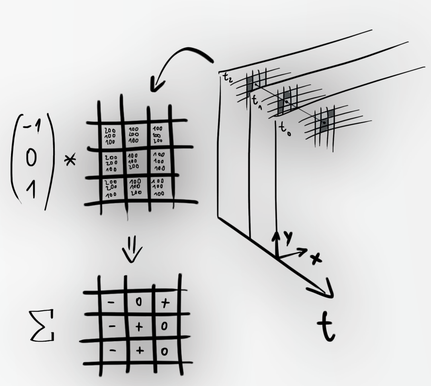


Bildverarbeitung

Kantenerkennung in der Zeit

Zur Unterscheidung der Bewegungsrichtung der Person kann man einen Sobel-Filter auf eine Bildfolge anwednen.

Der folgende Matlab-Code filtert drei aufeinander folgenden Bilder:



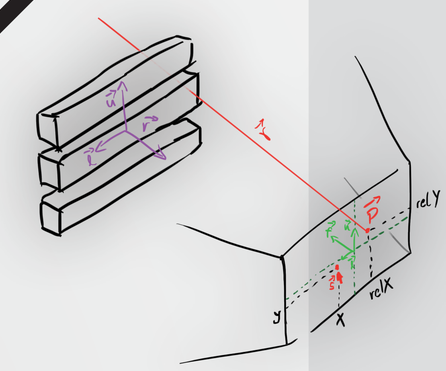

```
% Calculates a sobel fitered time image from
% an image series timeFrames with width and height.
%
% returns:
%    image =>    a remapped result-image of the sobel-filter results
%    flow       =>    the origninal sobel-filter results
%    neg        =>    the rising edges dependent form a lower grey-value range
%    pos        =>    the falling edges dependent form a higer grey-value range
%
function [ image, flow, neg, pos ] = timeFilter( timeFrames, width, height )
     % copy image size
     image = timeFrames(:,:,1);
     image = double(image);
     % create sobel filter-matrix
     sobelFilter  = [-1.0, 0.0, 1.0];
     % loop through all pixels in the image series
     for i=1:height
          for j=1:width
               % apply sobel-filter to the pixels in time and store the result
               image(i,j) = ...
                    sobelFilter(1) * double(timeFrames(i,j,1)) + ...
                    sobelFilter(3) * double(timeFrames(i,j,3));
          end
     end
     % copy results as image-flow
     flow = image;
     % adjust image values to positive values
     image = (image+255)*0.5;
     % get the rising and falling edges dependent form a grey-value range
     neg = image<49;
     pos = image>216;
     % return all the rising and falling pixels in range
     neg = neg.*image;
     pos = pos.*image;

end
```

Computergrafik

Bar Visualization

Eine weitere alternative Uhrdarstellung ist durch die Balkenuhr umgesetzt worden. Sekunden, Minuten und Stunden werden hier durch skalierte Balken angezeigt.

Die Visualisierung ist in einer Render-Texture gezeichnet und auf die CEGUI gelegt.
Außerdem bewegt folgender Code die Uhr abhängig von der Mausposition:




```
// mouse orientation
OIS::MouseState state =
      StandupApplication::getInstance()->getMouse()->getMouseState();
static Ogre::Ray& ray = Ogre::Ray();
// get x and y coordinates from mouse
float x = state.X.abs;
float y = state.Y.abs;
// get the position in relavive screen-coordinates
float relX = x/(float)state.width;
float relY = y/(float)state.height;
// flip the coordinates horizontal and vertical
relX = (1-relX);
relY = (1-relY);
// invert the projection of the screen-coordinate into a ray
mCamera->getCameraToViewportRay(
      relX * mCamera->getViewport()->getWidth(),
      relY * mCamera->getViewport()->getHeight(), &ray);
// get the direction of the ray
dir = Ogre::Vector3(
      ray.getDirection().x,
      ray.getDirection().y,
      ray.getDirection().z);
// ensure that it is normalized
dir.normalise();
// get the cross-product form the direction- and the
// up-vector as the left pointing vector
Ogre::Vector3 left = dir.crossProduct(mCamera->getUp());
// normalise the result
left.normalise();
// apply the origin as the light-position
Light->setPosition(ray.getOrigin());
// apply the three orthonormal vectors as
// the new orientation of the clock
mClockNode->setOrientation(
      Ogre::Quaternion(dir, mCamera->getUp(), left));
```

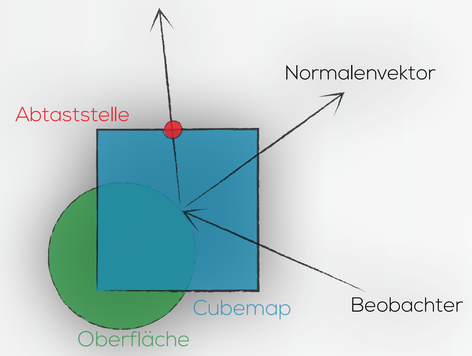
Cubemap Reflexionen

Für den Zweck der Darstellung einer Reflexion an einer Oberfläche wird in der Cubemap ein 360°-Panorama der Umgebung gespeichert.

Der Vektor von der Kamera zu dem zu zeichnenden Punkt auf der Oberfläche wird an dem Normalenvektor der Oberfläche gespiegelt.

Der resultierende Vektor zeigt von der Mitte des Würfels nach außen. Die Stelle, an der dieser Vektor den Würfel schneidet, markiert den Bildpunkt, der abgetastet wird.

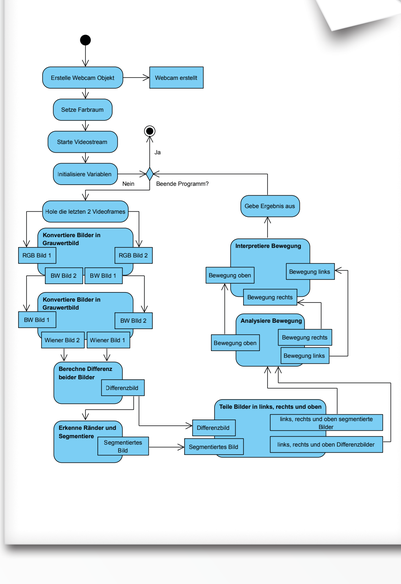
Das Resultat wird als Parameter für die Abtastfunktion der Cubemap verwendet.



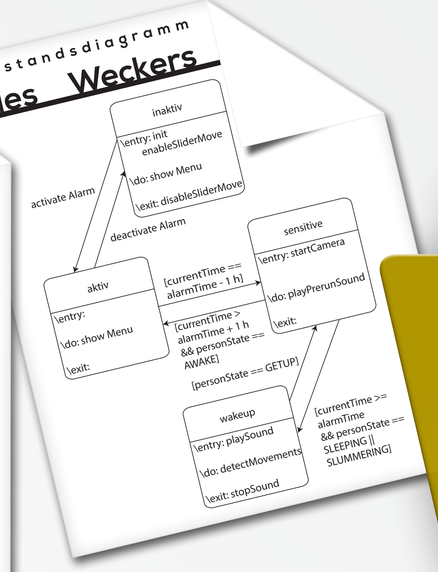
Bibliotheken und APIs

Ogre3D	3D Grafik-Engine in C++
CEGUI	Fenster und Widgets für Grafik-APIs
fmod	Audio-Engine

Aktivitätsdiagramm der Bildverarbeitung

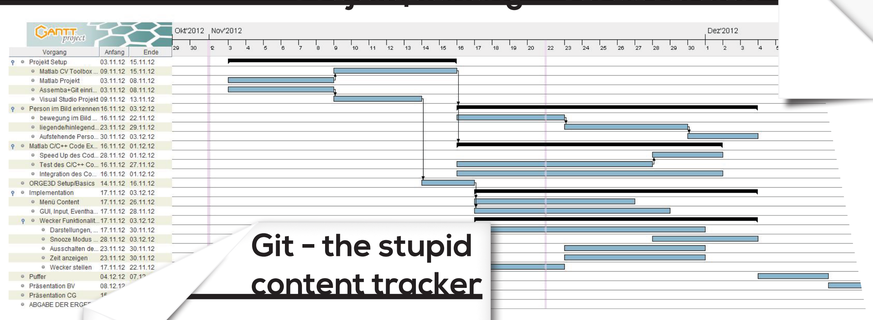


Zustandsdiagramm des Weckers

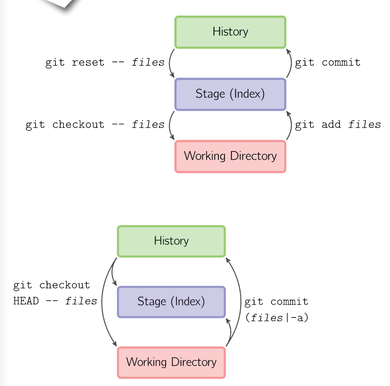


Projekt management

Projektplanung - Vorabversion



Git - the stupid content tracker



marklodato.github.com/visual-git-guide/index-en.html

Das Team

Hans Ferchland (Projektleiter)
Roman Hillebrand
Hady Khalifa

HSHL - Computervisualistik und Design
Computer Vision Projekt WiSe 2012/ 13