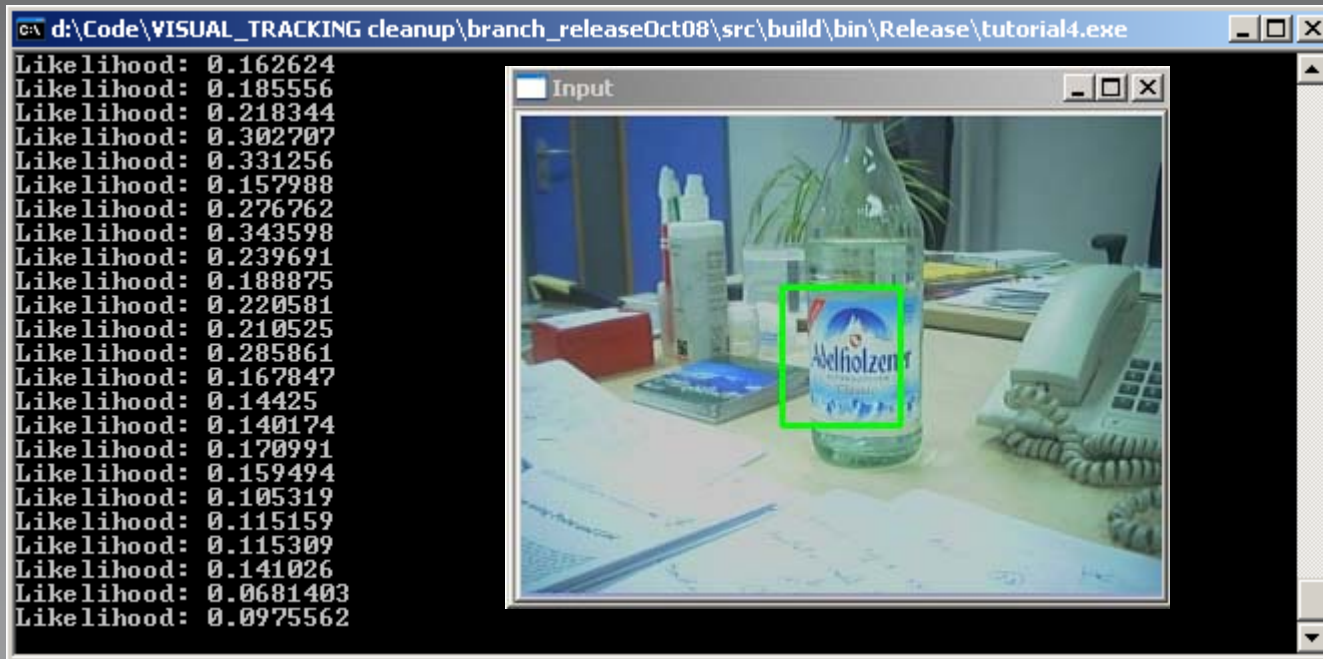
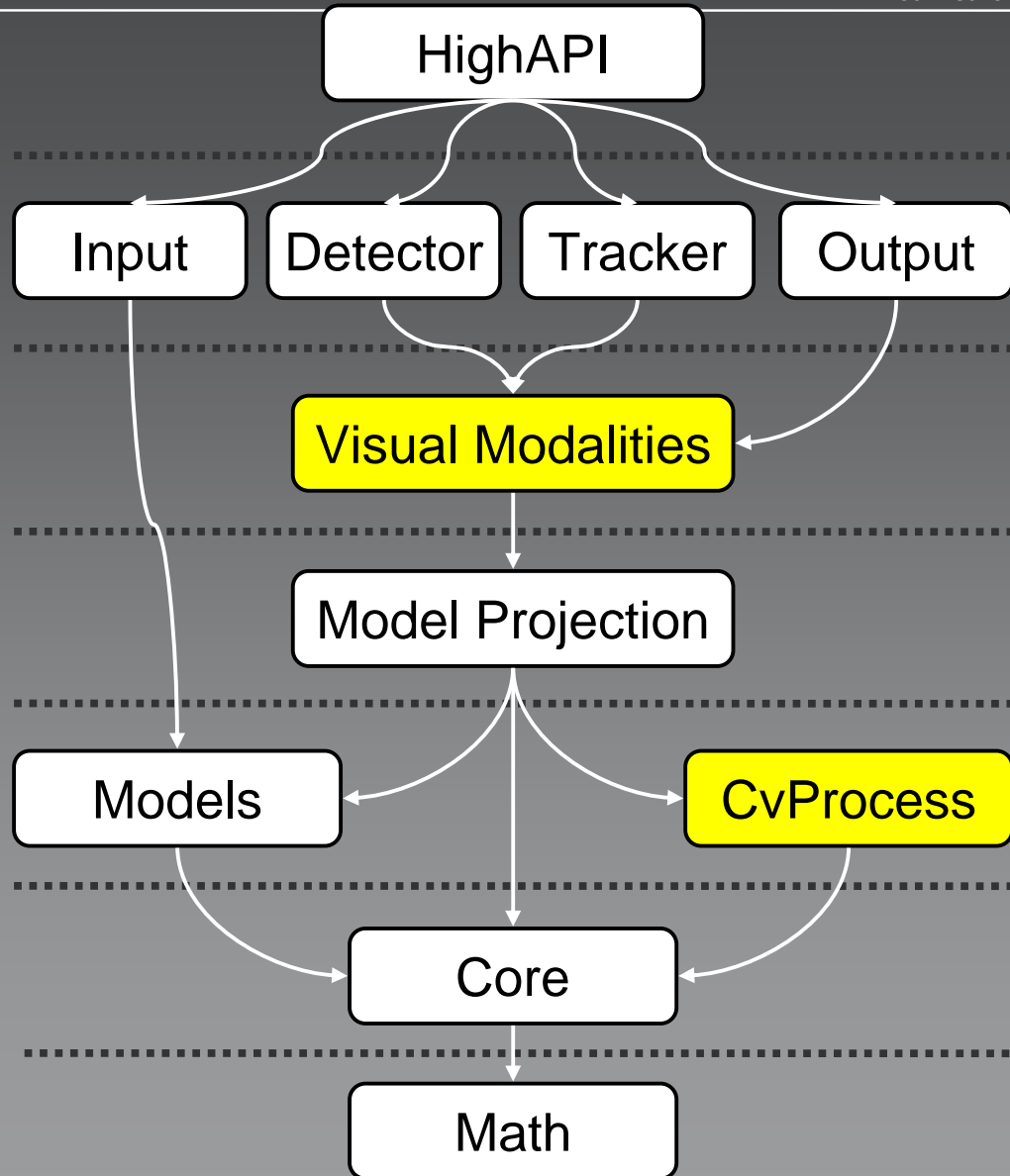


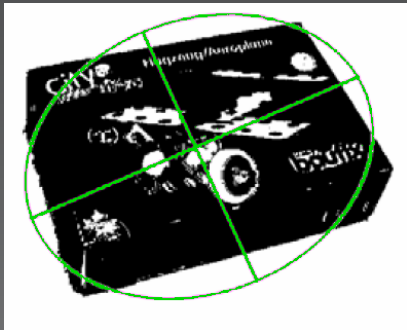
## OpenTL – Tutorial 4

- GOAL: Setting a color histogram modality, evaluating the likelihood function

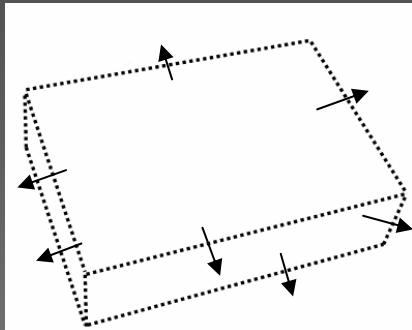




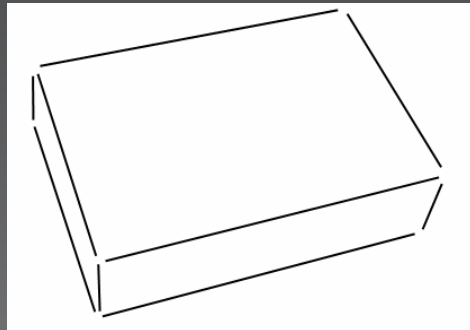
Shape moments



Intensity gradients



Contour lines



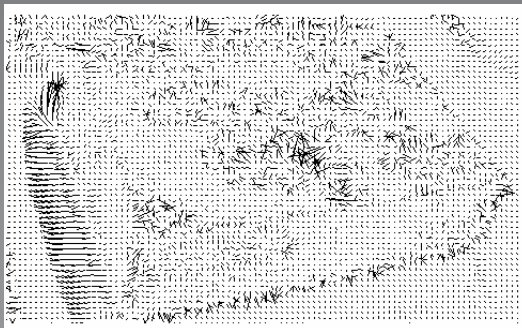
Color statistics



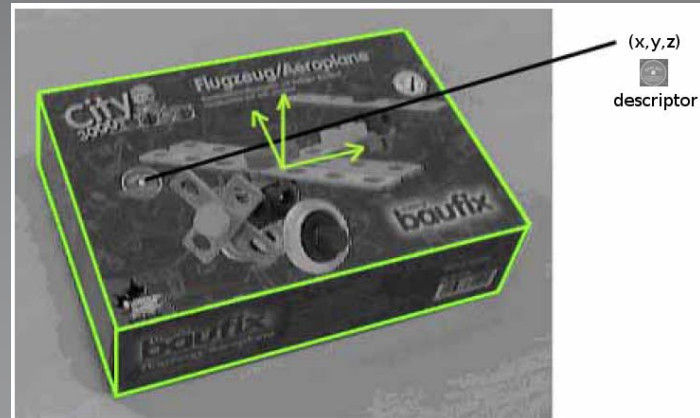
Texture template



Optical flow



Local keypoints



## Visual Modalities

- `opentl::modalities` namespace

Modality-related data processing (edge, color statistics, feature points, ...), all derived from a common abstraction

## Modality class abstraction

```
opentl::modalities::Modality
{
    • preprocess(opentl::models::SensorData & sensData)
      Model-free processing of sensor data (image)

    • sampleModelFeatures(std::vector<State *> * stateVec)
      Select visible features from shape/appearance model

    • match[Pixel/Feature/Object]Level(std::vector<State *> * stateVec)
      Model projection and matching with image data

    • updateModelFeatures(std::vector<State *> * stateVec)
      Update image features (after tracking)
}
```

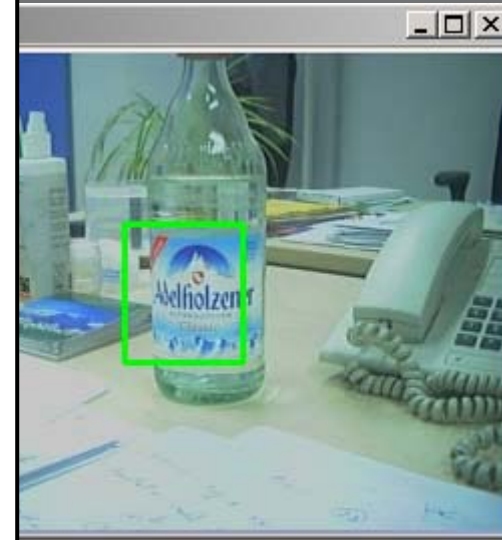
# Color histogram modality

```
opentl::modalities::ColourHist2D
```

```
{
```

- `preProcess()`  
Convert image from RGB to HSV
- `sampleModelFeatures()`  
Collect HSV histogram from the object model
- `matchFeatureLevel()`  
Compute image histogram under pose hypothesis,  
and compare with off-line model histogram
- `updateModelFeatures()`  
Update model histogram from image data,  
at the on-line estimated pose

```
}
```



## Color histogram modality

Parameters: ColourHist2D::Param

- `matchFeatVariance` = variance of Gaussian Likelihood (in Bhattacharyya histogram distance) =  $(0.5^2)$
- `preProcess_DestColorFormat` = `cvprocess::CvtColour::HSV`
- `sampleModelFeatures_Bins1,2,3` = bins of HSV color histogram (default: 10,10,1)
- `matchFeatWithMdetect` = use off-line histogram (true)
- `matchFeatWithMtrack` = use on-line histogram (false)
- `matchFeatMdetect/trackWeight` = fusion weights (1.0, 0.0)

## Likelihood

- `opentl::Modalities::Likelihood`

Provide Likelihood function evaluation for a modality processing tree, with data fusion



## Likelihood

- `Likelihood::addChild(Modality *, modalities::Modality::T_LEVEL)`

Purpose: add a child modality to the likelihood, at a given matching level

In this case,

`Modality::FEATURE_LEVEL`

## Likelihood

- Likelihood::implicitModel(States, Values)

Purpose: evaluate the implicit model  $P(z|s)$ ,  
i.e. the Gaussian probability of the  
measurement residuals

States = state hypotheses (s)

Values = output (vector of double)  $P(z|s)$