



FixedCam option added to the AVED project

FixedCam and AveragingTime options added the 3-13-2006 on nanomia /home/aved/Jerome/

Usage: *runclip -a fixedCam-6 my_clip.mov*

Project reference: AVED-04.20.2006

Here are the steps and the files considered to add the option to the AVED project: the FixedCam option allows to compute the averaging of the background only ones each n frames. The n frames parameter is given by the AveragingTime. This concerne mainly fixedCam videos for what the background doesn't change as often as for a motion camera.

1. File mbarivision/DetectionParameters.H

```
class DetectionParameters {
public:
    //! @param itsAveragingType = type of camera (fixedCam or Transect)
    std::string itsAveragingType;
    //! @param itsAveragingTime = time between each frame caught by the system
    int itsAveragingTime;
}

class DetectionParametersModelComponent: public ModelComponent {
private:
    ModelParam<std::string> itsAveragingType;
    ModelParam<uint> itsAveragingTime;
}
```

2. File mbarivision/DetectionParameters.C

```
DetectionParameters::DetectionParameters()
{
    this->itsAveragingType = p.itsAveragingType;
    this->itsAveragingTime = p.itsAveragingTime;
}

DetectionParametersModelComponent::DetectionParametersModelComponent(ModelManager &mgr)
:ModelComponent(mgr, std::string("DetectionParameters"), std::string("DetectionParameters")),
  itsAveragingType(this, "MDPavgType", ""), // Added 3-2-2006
  itsAveragingTime(this, "MDPavgTime", 6), // Added 3-2-2006
}
```

```

void DetectionParametersModelComponent::exportOptions(const int whatToExport, const bool recurse)
{
    manager->requestOption(this, itsAveragingType);
    manager->requestOption(this, itsAveragingTime);
}

void DetectionParametersModelComponent::reset(DetectionParameters *p) {

    if(itsAveragingType.getVal().length() > 0)
        p->itsAveragingType = itsAveragingType.getVal().data();
    if(itsAveragingTime.getVal())
        p->itsAveragingTime = itsAveragingTime.getVal();
}

```

3. File mbarivision/ModelOptionDefs.C

```

{ MODEOPT_STRING, "MDPavgType", MOC_MBARI,
  "AveragingType: fixedCam (the system catch 1 frame each n second; default: n=6) or Transect (the system catch 1 frame
  each second)",
  "mbari-avg-type", '\0', "<fixedCam|Transect>",
  "" },
{ MODEOPT_INT, "MDPavgTime", MOC_MBARI,
  "AveragingTime: Time the elapsed by the system to catch 1 frame in second; ",
  "mbari-avg-time", '\0', "<int>", "6" },

```

4. File mbarivision/mbarivision.C

This file is the main one. A compteur allows to know how many frames are computed and in this way to know when the background averaging has to be compute.

```

uint compteur = 0;
for (int curFrame = ifs->minframe(); curFrame <= ifs->maxframe(); ++curFrame)
{
    /*-----*/
    /* Adeed the 03-02-2006 for Averaging Type */
    if(!strcmp("fixedCam", detectionParms.itsAveragingType.c_str())) {
        ++compteur;
        if (compteur%(detectionParms.itsAveragingTime) == 0)
        {
            avgCache.push_back(img);
            LINFO("----->Caching frame %06d timecode: %s",ifs->frame(), tc.c_str());
        }
    }
    else { // Default mode is Transect: Frame added each second
        avgCache.push_back(img);
        LINFO("Caching frame %06d timecode: %s",ifs->frame(), tc.c_str());
    }

    /* end */
    /*-----*/
}

```

5. File scripts/librunclip

This script is called by the runclip script.

avgtype="" # variables for the averaging option -> fixedCam or not
timeToAvg=6 # time to wait for catching a frame: default 1 frame each 6 seconds

```
# Check arguments
while getopts Xfdxivj:c:t:e:p:a:s: option
do
  case $option in
    a) avgtype=${OPTARG%-*}
       timeToAvg=${OPTARG##*-}
       echo "Process with $avgtype settings with 1 frame caught each $timeToAvg seconds";;

    if [ $avgtype = "fixedCam" ]; then
      if [ $timeToAvg = 0 ]; then
        timeToAvg=6
      fi
      useroptions="$useroptions --mbari-avg-type=$avgtype"
      useroptions="$useroptions --mbari-avg-time=$timeToAvg"
    fi
  fi
```