Video Imagery Processing Use Case

This use case describes the processing of video data for tasks such as fish counting, anomaly detection and summarization, dehazing, and other image processing techniques that can provide insight into the undersea environment.

The project files and source code are stored in a Github repository which will be made public when it is ready.

This project gives two Python 3 use cases for video and image process for the ONC Oceans 3.0 API and Sandbox. An Oceans 3.0 account is required (it's free) to obtain an API token to run the examples. The token can be obtained by creating an account and accessing the Web Services API tab. Update the `params.json` file(s) with your token to run the use cases.

Dehaze

ONC has videos and camera located above and below water. Often, when processing image below water there are suboptimal viewing conditions due to poor lighting conditions, sediment, mechanical malfunctions, etc. This makes it difficult for ONC scientists to analyze these images for evaluating biodiversity, etc.

Dehazing is the process of remove haze from an image so objects that are difficult to see are illuminated.

Dehaze Algorithm

The algorithm for dehaze is in the main script `dehaze_script.py`. The tuneable parameters for a) connecting to the API, b) search the API, and c) dehaze algorithm are contained in `params.json`.

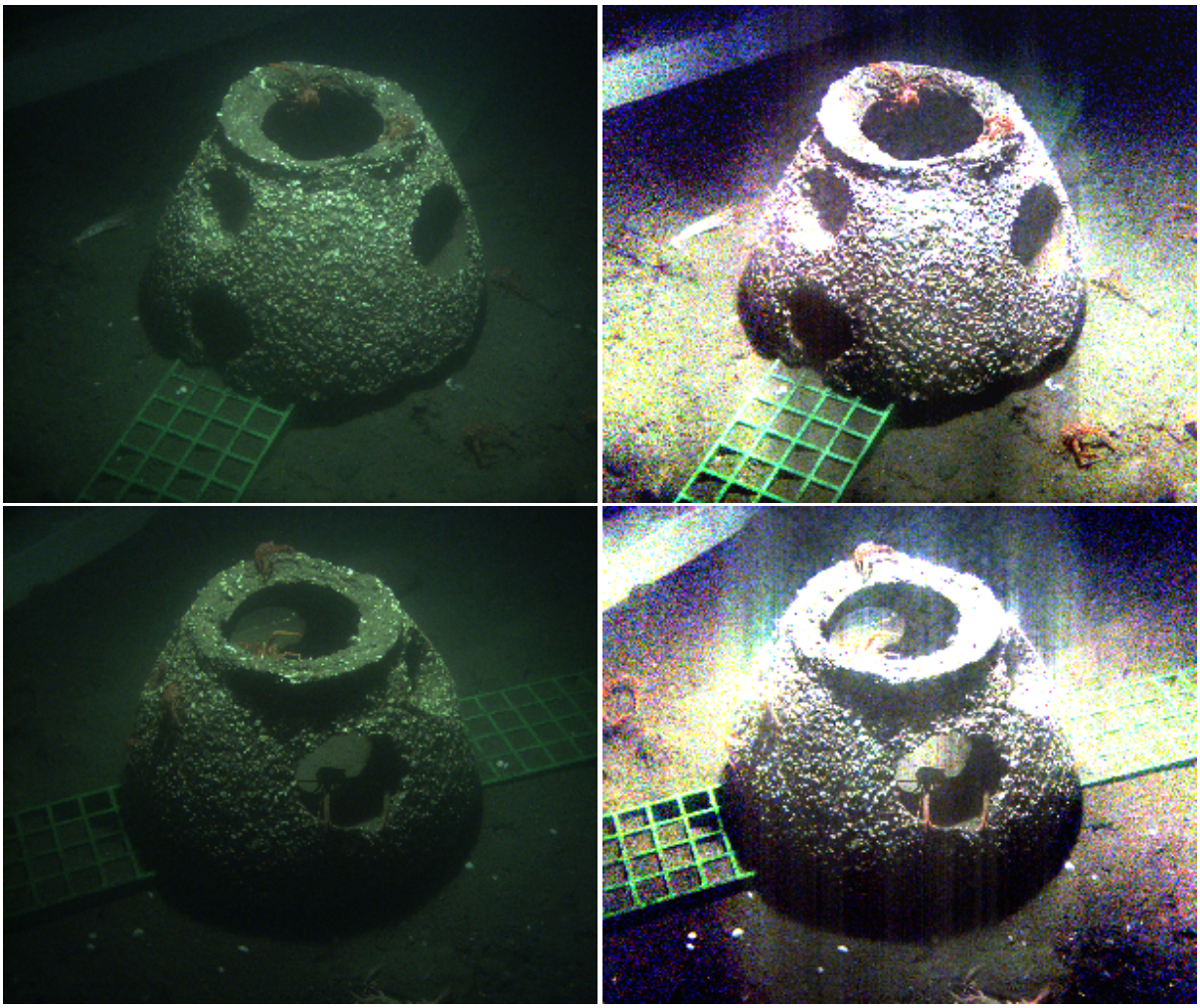
Params

- preprocess : Enable preprocessing (Default=True)
- wsize : Size of kernel/neighbourhood to consider when calculating the dark channel (Default=19)
- ratio : How many "bright" pixels in the dark channel (Default=0.001)
- omega : A percentage of the dark channel for the hazy image normalize by the atmospheric light (Default=0.98)
- refine : Refine the transmission map using a guided filter (Default=False)
- t_0 : Since the dark channel tends toward 0, this prameter prevents the image from being too dark (Default=0.1)

Workflow

1. The images are preprocessed by taking the zero-mean of the RGB channels and clipping between `[-1,1]` and normalized.
2. The dark channel (how dark the image is) is calculated by using a minimum filter.
3. The atmospheric light is calculated by finding the max RGB where the dark channel intensity is at it's brightest
4. The transmission (how the light is distributed) is calculated by applying omega to the normalized dark channel.
5. (optional) The transmission map can be refined using an edge-preserved guided filter. This is the only use of OpenCV in the algorithm, so if you have not installed opencv successfully you can run with `refine=0`.
6. A non-hazy image is reconstructed using the transmission and atmospheric light.

Note: The chromatic values are often adjusted due to this process so the original colors are often not preserved exactly.



Video Summarization

ONC deploys video feeds to multiple locations and has accumulated terabytes of video data. However, not all this video data is useful as it may have poor visibility or no events of interest. It is a time-consuming process to look through months and years worth of video data. This is produces a storage problem as a single day of video can produce 3 GB of video data. One way to automatically handle this is to use video summarization techniques to preprocess the data and remove or sample frames that might be of interest to ONC scientists.

Video Summarization Algorithm

The algorithm used was by Dash & Albu. The algorithm models the background using a Gaussion Mixture Model (GMM). This produces a lot of noise in underwater videos because of current and particulars in the water. To offset that, per pixel activation and ajustment functions were added. When a certain number of pixels are activated, the samples frames are added to the output summarized video. For most cases, roughly 50% of the video is reduced.

Params

- samplingRate : The algorithm with analysis every N frames (Default=16)
- sampleThreshold : Number of significant events required to trigger video write (Default=10)
- keepOriginal : Keep original video and tag summarized video with summ, otherwise overwrite (Default=1)
- learningRate : The rate the GMM learns (decrease to register smaller events) (Default=0.1)
- use_gpu : Use the GPU for processing the frames (Default=0) NOTE: Not tested in the sandbox.

Workflow

1. The frame is selected and reduced in size and grayscaled.
2. The Gaussion Mixture Model is found using the OpenCV MOG function.
3. The GMM is adjusted based on accumulation and decay rates to take into account noisy backgrounds and clipped to between `[0-1]`
4. Per pixel adjustment and activation functions are applied so that pixels that have consistent intensity changes between frames are activated as events.
5. High level events are counted and if they exceed the sampleThreshold the frame is written to the summary
6. (optional) Once the video is completely analysed, the original is overwritten if keepOriginal is not set.

AXISCAM00408C...00Z_debug.mp4

AXISCAM00408C...00Z_debug.mp4