**Содержание задачи**

Разработать (подобрать) алгоритм(ы) обработки изображений, который осуществляет одновременно подавление шума и подчеркивание границ. Такого рода подход описан в научных публикациях (Digital Detail Enhancement) и, в общем виде, состоит из двух конвейеров параллельной обработки видеопотока. Один конвейер удаляет шум, другой – выделяет границы. Затем результаты работы обоих конвейеров складываются по некоторому закону.

Удаление шума

Выделение границ

Необходимы алгоритмы обработки для конвейеров и алгоритм смешивания результатов их работы так, чтобы максимально удалить шум с изображений сохранив при этом реальные объекты и их детали.

Алгоритм(ы) должен иметь хотя бы один параметр, который регулирует воздействие на обрабатываемое изображение. От полного отсутствия обработки до максимально возможного.

К результатам работы должны прилагаться методы оценки результатов работы алгоритма по объективным показателям (уровень шума, частотно контрастная характеристика и т.п.).

**Описание исходных данных**

Исходные данные распределены по папкам. Каждая папка содержит бинарный файл, содержащий несколько последовательных кадров некоторой сцены и файл описания. Имя бинарного файла может быть произвольным и имеет расширение yuv или bin. Формат данных в бинарном файле задан в специальном файле описания, который находится в той же папке. Файл описания всегда имеет имя dataDescription.ini и содержит следующие данные, но не ограничивается ими:

[Global]

byteOrdering=LE

width = 320

height = 240

numOfFrames = 32

где, byteOrdering - порядок следования байт (LE - little-endian или BE - big-endian) предполагается, что все данные 16-битные одноцветные

        width - ширина кадра изображения в пикселях

        height - высота кадра изображения в пикселях

        numOfFrames - количество кадров изображения, подлежащих обработке

Исходные изображения одноцветные, имеют разный контраст и содержат временнЫе и пространственные шумы.

**Ограничения**

Допускается использовать внешние библиотеки Python с открытым исходным кодом или скрипты Matlab, для которых доступен исходный код. Для обработки данных допустимо использовать только целочисленные расчеты. Преобразование в формат с плавающей точкой и обратно должно быть исключено.

**Вспомогательные скрипты для лёгкого импорта данных (Python)**

Для облегчения работы по импорту сырых данных предоставляется скрипт на языке Python 3 загружающий бинарный файл в структуры данных importRAW.py.

Для облегчения работы по формированию результирующего видеофайла предоставляется скрипт на языке Python 3, который создает такой видеофайл SampleVideo.py.

Данные для обработки и примеры программ можно скачать по ссылке.

[https://r4.res.outlook.com/owa/prem/images/generic_16x16.pngvideo-enhancement\_task.7z](https://yukonaow-my.sharepoint.com/:u:/g/personal/a_chizh_yukonww_com/ERL9MkToV-9Ai0wZEupcJBwBUEDhI-lakbAZYE2qGjcuLQ?e=GaEdkH)

**Результат работы**

1. Скрипт на языке Python 3 или Matlab, преобразующий бинарные файлы в видеофайл в формате avi с минимальным сжатием, и обработанные сырые данные в виде видеофайлов.
2. Отчет, содержащий
   * описание работы алгоритма
   * критерии оценки качества изображения и значения этих критериев до и после обработки
   * список используемых источников информации

**Порядок приемки и критерии оценки**

Приемка осуществляется комиссией, которая изучает отчет и применяет скрипт к тестовому набору данных. Результаты оцениваются по количественным критериям предложенным исполнителем и экспертной оценкой по следующим критериям:

* Можно различать объекты и идентифицировать их как холодные, так и теплые.
* Четко видеть теплые объекты, чтобы можно было определить, что за объект (птица, зверь, человек) и его основные параметры (порода, размер, количество отростков на рогах и т.п.).