3.2.4.4 WP4-43 HDR & Multi Spectral Imaging

ID	WP4-43	**Category**	System	**KET**	2.4.1 Data fusion and processing
Name	HDR & Multi Spectral Imaging	**License**	3-clause BSD	**TRL**	4
Owner	BUT	**Contributor**	BUT	**Contact**	BUT

Short description

The proposed algorithm / library implements High Dynamic Range (HDR) merging and tone mapping of images and videos. Tone mapping algorithm can be used for a wide range of scenes. Library is implemented in C/C++ and it is platform independent (only dependency is OpenCV). Proposed tone-mapping algorithm is based on Durand and Dorsey [36], which is based on the idea of decomposition of image into two separate layers - a base layer and a detail layer. The library also implements a real-time HDR merging algorithm developed at BUT. This technique minimizes ghosting artifacts typical for HDR video.

Technical specifications & Interfacing

Input	RGB/Grayscale/Bayer Image, 8bit Depth, 3x Exposition
Output	1x Image Data, 8bit Depth
Input Interface	Software C/C++API
Output Interface	Software C/C++API

Data flow

Algorithm, as shown in Figure 36, starts with decomposition of the luminance layer (I) for every RGB pixel. Next, the luminance layer is transferred into logarithmic domain. The base layer is obtained by application of the bilateral filter. Next, the base layer is compressed using point-by-point multiplication with compression factor (which is computed from current frame global parameters) and details are added subsequently. Last step is color restoration, based on approach introduced by Mantiuk at al. [37], which prevents color shifts and helps to preserve color appearance.

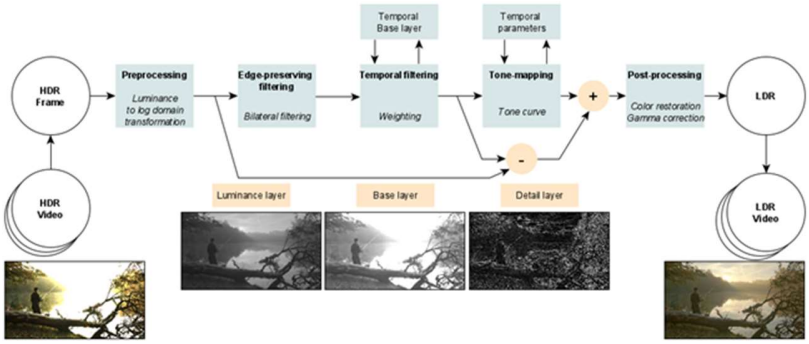

Figure 36: HDR and Multispectral imaging data flow

Usage Example

```
// Tone mapping usage example
float temporalAdaptation = 0.8f;
float baseContrast = 3.5f;
hdr::tmo::TemporalTonemapDurand tmo(temporalAdaptation, baseContrast);
// ...
// Load images from filesystem
// ...
for (const auto &file : files)
{
    cv::Mat mat = cv::imread(file.string(), cv::IMREAD_UNCHANGED);
    durand.apply(mat);
    auto toneMapped = durand.getImage();
```

```
    // ...
    // display or save processed images
    // ...
}
```

Measured Improvements

Even though proposed tone mapping is focused on video, it can process a wide range of individual HDR images with high-quality results. To evaluate tone mapping quality, we chose a metric that compares the HDR image against the tone mapped image. The selected metric is FSITMB–TMQI (A feature similarity index for tone-mapped images -Tone Mapped Quality Index) [38], which enables to meet the goals– preserve as much detail as possible in tone-mapped image. The dataset was collected from a freely available HDR images of various origin to test a wide variety of scenes and lighting conditions. We have included state of the art tone mappers for images, which provide high-quality results. For global the TMOs, the Drago, Reinhard and Mantiuk TMO has been selected. iCAM06, Fattal, Mantiuk08, and original implementation of Durand was selected as local operators.

Proposed tone mapping is most often the first or among the first three. Overall, it can be observed that introduced modifications improved the quality of the original algorithm, which is the worst local tone mapping in this experiment (also confirmed by other works). In the original implementation of Durand TMO, mapping is controlled mainly by minimum luminance without respecting the maximum, which often leads to images that contain a large number of saturated pixels. Proposed algorithm typically provides better results than iCAM06, which is the state of the art local TMO algorithm. Table 1 shows evaluation results of FSITMB-TMQI metric for different tone mapping algorithms (Green - the best score, Blue - second-best score, Red -third-best score).

Table 1: Evaluation of FSITMB-TMQI metric for different tone mapping algorithms

Scene Name	ICAM06	Drago	Durand	Fattal	Mantiuk06	Mantiuk08	Reinhard05	**Proposed**
Adjuster	0,857	0,822	0,835	0,841	0,833	0,835	0,810	0,868
Apartment float	0,885	0,875	0,836	0,889	0,842	0,874	0,873	0,870
Atrium Night	0,842	0,827	0,836	0,832	0,818	0,834	0,803	0,852
Balls	0,908	0,880	0,891	0,905	0,882	0,904	0,777	0,919
Foggy Night	0,872	0,875	0,891	0,885	0,847	0,890	0,794	0,901
Bonita	0,906	0,836	0,888	0,761	0,854	0,886	0,705	0,915
Cannon	0,949	0,918	0,913	0,928	0,894	0,940	0,890	0,938
Belgium House	0,873	0,857	0,851	0,882	0,835	0,849	0,863	0,866
National Cathedral	0,854	0,789	0,839	0,760	0,822	0,832	0,795	0,869
Design Center	0,885	0,872	0,879	0,834	0,866	0,875	0,846	0,885
Desk	0,824	0,777	0,819	0,763	0,797	0,807	0,804	0,841
Display	0,854	0,841	0,848	0,895	0,815	0,835	0,871	0,870
Forest Path	0,916	0,910	0,883	0,941	0,892	0,910	0,878	0,928
Golden Gate	0,837	0,873	0,777	0,844	0,839	0,844	0,748	0,901
Memorial	0,899	0,845	0,902	0,862	0,884	0,892	0,810	0,932
Mpi Atrium 3	0,811	0,799	0,807	0,791	0,792	0,805	0,794	0,828
MtTamWest	0,880	0,859	0,844	0,902	0,835	0,866	0,874	0,891
Nancy Church 1	0,860	0,876	0,879	0,871	0,858	0,880	0,727	0,875
Navy	0,845	0,799	0,825	0,583	0,813	0,836	0,835	0,866
Ocean	0,818	0,835	0,814	0,831	0,805	0,808	0,800	0,811
Rosette	0,861	0,838	0,857	0,642	0,856	0,880	0,842	0,911
Seymour Park	0,931	0,911	0,893	0,928	0,894	0,925	0,902	0,943
Synagogue	0,914	0,905	0,877	0,929	0,871	0,923	0,908	0,943
Tree	0,851	0,816	0,805	0,892	0,783	0,821	0,835	0,843
Average Rank	2,833	5,250	4,667	4,542	6,333	4,167	6,250	1,958

Referencing API - GIT / Open

Available at: https://github.com/inosko/hdr-sdk