Near-lossless 3D-image Compression Using Hypergraphs

Luc GILLIBERT and Alain BRETTO

Université de Caen, GREYC UMR-6072, Bd Marechal Juin BP 5186, 14032 Caen cedex, France.
lgillibe@info.unicaen.fr, alain.bretto@info.unicaen.fr

Abstract

We extend the hypergraph-based image representation to 3D-images. This extended representation conducts to a generalisation of the HLC lossless compression algorithm [1] for near-lossless 3D-image compression: HNLC.

Let $I$ be a 3-dimensional matrix represented image. We build a hypergraph $H^\alpha(I)$, called the extended hypergraph representation of the image for the tolerance $\alpha$, as it follow: the vertices of $H^\alpha(I)$ are the voxels of the 3D-image and the hyper-edges are the maximal rectangle parallelepipeds such that inside a rectangle parallelepiped all the voxel colours are at a distance inferior to $\alpha$ of the colour of the upper higher left corner of the rectangle parallelepiped. There is an algorithm building the hypergraph. The complexity of this algorithm is $O(n^2)$ in the worst case ($n =$ voxels number).

A rectangle parallelepiped can be stored as a couple of points. This representation required several bytes. For that reason, we introduce an integer $K$ and we use this process for compressing and image $I$:

1. Build $H^\alpha(I)$ and order its hyper-edges such that the biggest parallelepiped comes first. We denote this ordered hypergraph $H^\alpha_0(I) = \{R_1, \ldots, R_m\}$.
2. Extract from $H^\alpha_0(I)$ the partial hypergraph $H^\alpha_K(I) = \{R_x \in \{R_1, \ldots, R_m\}; x \in X\}$. The set of indices $X$ is chosen such that for all $x \in X$, $R_x$ contains at least $K$ voxels that are not in $\bigcup_{i \in X, i < x} R_i$. Store the hyper-edges of $H^\alpha_K(I)$.
3. Traverse the image and, for each voxel that is not in a rectangle of $H^\alpha_K(I)$, write the voxel colour in a data segment. Finally compress the hypergraph and the data segment with PPMd [2].

With that process it is possible to only store rectangle parallelepipeds giving information on at least $K$ voxels. The others voxels of the image are stored in a data segment by a common way: one byte for one voxel. We compared our algorithm to JPEG-LS.

We used the GIF files from the animals section of Horton-Szar animation library. With a tolerance of 2 or 8, HNLC combined with PPMd is better than JPEG-LS.

References
