Rasterized Image Databases with LSH for Compression, Search and Duplicate Detection

Zoya Bylinskii
MIT CSAIL

Maria Shugrina
MIT CSAIL

Andrew Spielberg
MIT CSAIL

Wei Zhao
MIT CSAIL

ABSTRACT

TODO.

1. PROBLEM STATEMENT

We formally define the problem as follows. Consider an input set of i images, and without loss of generality, assume each image is square and is composed of m^2 pixels. We wish to store our set of images in a database using lossy compression such that we minimize the total space used by the database subject to certain *quality* constraints based on similarity metrics. There are many possible formulations of these quality constraints; we detail those that we considered for this paper in section TODO. Beyond mathematical metrics, subjective methods are also interesting to consider for formulating the quality constraints; one could imagine a scenario in which image quality is assessed by humans through a crowdsourced system, perhaps using an engine such as Amazon's Mechanical Turk TODO: cite. We consider such subjective similarity metrics outside the scope of this paper and focus on the mathematical metrics for now.

We wish to construct a database that trades off minimizing the amount of required space with maximizing read and write speeds of the data. In the following sections, we discuss models for estimating these values. We construct our compressed representation in the following way: First, we segment each image into a certain number of image patches. Next, we store a set of precomputed image patch "exemplars" in an auxiliary table. Finally, rather than represent each patch in each image explicitly, we approximate it by instead storing a pointer to the "most similar" exemplar. We discuss similarity later in section TODO as we discuss our quality constraints. For this paper, we consider all patches to be the same size and each image to contain the same number of patches.

2. RELATED WORK

Image databases, in particular rasterized.
Image compression, in particular JPEGs.
Image similarity functions.
Image quality functions.
Hashing, in particular [1].

3. ANALYSIS

Assume for now that we choose to store p patches in our auxiliary table. In practice, we choose p to be a function $p\colon \mathit{Function} \rightarrow \mathbb{N}$ which maps from our similarity metric to a number of patches to store. Assume also that each patch is square and composed of n^2 pixels, where n is a user defined parameter. We further assume that each pixel requires 8 bytes to store and that each pointer is 8 bytes (a standard integer for a 64-bit system). Under this "image only" scheme, in the case where we have i images, the cost c_i to store all the images in our database is:

$$c_i(i, m) = 8im^2 \tag{1}$$

In the case where we store pointers to patches, we have two tables: one table to store pointers to image patch exemplars, and a second table to store the exemplar data themselves. Under this "patch pointer"scheme, in the case where we have i images and p patches, the cost c_p to store all the images in our database is:

$$c_p(i, p, m, n) = 8i(\frac{m}{n})^2 + 8pn^2 \tag{2}$$

.

The first term is the cost of storing the pointer data, while the second term is the cost of storing the patch exemplars themselves.

4. REFERENCES

[1] A. Andoni and P. Indyk. Near-optimal hashing algorithms for approximate nearest neighbor in high dimensions. *Commun. ACM*, 51(1):117–122, Jan. 2008.