# Using volumetric stereo and pattern segmentation techniques to estimate the number of jellybeans in a jar

Mark O'Meara, Alex Cope

January 30, 2104

## Motivation

The challenge of estimating at a glance the number of jellybeans in a large transparent jar is a classic parlor game, one that has not flagged in appeal for many years. Even as recently as last November, prizes as large as $125,000 have been offered at carnivals for accurate counts of immense numbers of colored beans in gigantic containers [1]. But far from being the sole domain of human intuition, the inherent difficulty of this problem presents several component issues that we believe are particularly conducive to computer vision techniques. And although this specific problem is relatively trivial in its scope, we suspect that the techniques devised for its successful solution could subtend many other important problems involving large count estimations, like crowd size and bird flock/fish school estimations.

In our estimation, the "jellybean jar" problem can be decomposed into three relatively tractable subproblems: 1. 3-D modeling/measurement of the transparent container from multiple images, 2. Object segmentation of the similarly sized but multicolored jellybeans, and distinction between the partially occluded and front-layer beans, and 3. Spacefilling extrapolation and final estimation of the total jellybean count from information gleaned from these first two analyses.

A large part of the renowned difficulty of this problem is the struggle of the human eye to correctly segment out and keep track of individual jellybeans from the horde, and this will be a steep challenge for our team as well. However, we think a few aspects of the problem lend themselves especially well to camera analysis. For one, the issue of scale, a persistent ambiguity in 3D modeling even after self-calibration and bundle adjustment, is irrelevant to this problem, as the ratio of container-to-jellybean size does not depend on any measure of absolute size. Secondly, the convexity of the containers and bright variegated colors of the jellybeans means that voxel carving could be an effective method for accurately modeling the system. And finally, the quantitative nature of this challenge, and the single integer output, will lend itself well to easy measurement of our improvement and efficacy as we progress. The ability to vary the size of the container will also allow us to limit the difficulty of the problem at first, later scaling up to much larger sizes if accurate estimations are achieved. As to the problem of segmentation, we believe that multicolored jellybeans will present a sufficient challenge, but a higher level of difficulty, viz. having all the jellybeans be of one color, is easily formulatable if desired.

# Technical Approach

## Volumetric stereo methods

The first component of our project will be to construct accurate 3D models of the transparent containers holding the jellybeans. Although our approach may vary as we progress, our initial plan is to employ volumetric stereo techniques to generate these models (generating accurate point correspondences needed for SRM seems prohibitively difficult considering the complex color/contour patterns of the contained jellybeans). The heterogenous colors of the jellybeans seem particularly well suited to color-based voxel carving, and we will try this method first. However, we suspect that the reflective, non-Lambertian nature of the transparent jar might confound this approach, in which case we will try the regular space carving technique shown in lecture, casting the jar against a solid backgorund. Since the jars we will test will all be convex (or at least hyperbolically concave), we do not suspect that the concavity limitations of these carving methods will be a hindrance to this approach. Once we can consistently generate reasonable 3D models of the jars, we will be ready to progress to the next step: jellybean segmentation.

## Jellybean segmentation

Depending on the nature of the objects in the jar, different segmentation strategies will work with varying success. For example, with multicolored jellybeans, a relatively naive color threshold algorithm would work well. Because jellybeans have a regular shape, we could then examine the shape of these segmented objects to determine level of occlusion (i.e. which objects are at the top layer), which would be used to more accurately sample the number of objects. If this naive approach is not sufficient, or we want to count objects of homogenous color, then we plan to implement the watershed algorithm, in which the grey level of a pixel is used to estimate its height and segments are drawn over local minima [2], for more accurate segmentation.

## Number estimation

Some experimentation will be in order to produce the most accurate results here, but we plan on estimating the total number of jellybeans using the ratio between object and space on the top layer, as well as in the first partially occluded layer, to determine the number of jellybeans that could fit in the 3D space provided. Challenges we foresee include the separation of the top and partially occluded layer; if this technique is too inaccurate (accuracy measured by manual counting) we will also try measuring via a simulation the number of jellybeans based on the ratio of jellybean to jar volume.

# Progress

We decided to restructure our planned milestones; instead of starting with a 3D reconstruction of a jar, we opted to implement a baseline algorithm using a single image to estimate number of jellybeans under the assumption that the picture is taken head on and the container is cylindrical. We decided to move in this direction because we wanted to see how well a baseline algorithm would perform to see what sections of our approach we should focus the most on; we also wanted to familiarize ourselves with the image processing modules of OpenCV early in the project.

With that in mind, we implemented an algorithm which uses a watershed algorithm to segment the image. To preprocess the image, we first convert it to grayscale, then use adaptive histogram equalization to increase contrast locally around the jellybeans. We then dilate the image to remove noise. We then use an adaptive threshold to binarize the image and find the contours of the foreground objects, which we use as markers for the watershed algorithm. We then iterate through the segments, find the median area, and filter out outliers that deviate too far from the median (we assume these are not jellybeans).

To calculate the number of jellybeans from the segments, currently we are using a very simple heuristic that estimates the volume of a jellybean and the cylindrical jar from their respective areas and divides one by the other. The estimated number it outputs is 2,318. The actual number we counted was 632. Obviously, the algorithm is performing poorly now, and we've only tested it on one image, but we plan on making multiple significant improvements to the baseline algorithm in the upcoming weeks.

Figure 1: Original image.

## Remaining Milestones

Now that we've implemented a baseline algorithm, we have a lot of work left to refine the process and achieve much more accurate results.

Our next goal is to make a 3D reconstruction of a jar using space and voxel carving.

After that, we want to write a more accurate segmentation algorithm to segment the jellybeans on the surface of the jar and estimate layering based on occlusion.

At project end, we will be able to estimate based on the above data and geometric modeling the number of jellybeans in the jar.

Figure 2: Watershed algorithm applied to image.

# References


[1] Matheson, Whitney. (12 Nov 2013) "Silly Diversion: Can You Count the Jelly Beans?" USA Today [Online] Available: http://www.usatoday.com/story/popcandy/2013/11/12/jelly-beans/3506821/ [30 Jan 2014]

[2] Roerdink, Jos BTM, and Arnold Meijster. "The watershed transform: Definitions, algorithms and parallelization strategies" *Fundamenta Informaticae* 41.1 (2000): 187-228.