Methods for 3D Reconstruction from Multiple Images

Sylvain Paris

The presentation provides an overview of three phases of 3D reconstruction of an object from multiple views captured from multiple images. The phases or topics present include point matching, reconstruction from consistency functions, and reconstruction from regularized methods.

First, the presentation introduces the idea of matching two features in multiple images. The general method for accomplishing this task is through triangulation. Through triangulation, the coordinates of a point in 3D space can be computed based on two reference points, the angles that the reference points form with the point in question, and the distance or length between the reference points. The reference points in this case refer to the cameras used to capture the views. To simplify the problem of finding the two angles at the camera reference points, the cameras should be calibrated, but that is a different topic of discussion. The problem with this approach is that it is generally impossible to match all possible points. As such, the model that results from this method will generally have lots of holes, meaning the method achieves sparse reconstruction. However, sparse reconstruction is hardly useful, and what we hope to achieve is dense reconstruction.

Reconstruction from consistency functions attempts to create the 3D model of an object by ensuring consistency of the 3D model with the input images. This is measured by a consistency function, which optimizes a certain characteristic in the 3D model between input images. The metric could be color, texture, shininess, etc. The two main methods for achieving this. The first way is to find all the good points of correspondences from the 2D images to find the 3D point that they refer to. This method, as stated previously, requires a lot of different views, or else there will be a large amount of holes present in the background. Additionally, objects that are not Lambertian, cannot be approximated in this way because the multiple views obtained of the object can be drastically different depending on the viewpoint. In these cases, many methods assume that the objects are nearly Lambertian (meaning the energy in any direction from a point is constant), and they treat non-Lambertian portions of the object as noise. The second general method of achieving consistency is by beginning with a full, dense, and smooth shape, like a sphere or cube, and then removing points that are not correspondences, or reshaping the surface until the desired model is formed. Both methods described perform poorly on objects that have smooth surfaces. This is because every point in both images of the smooth surface matches every other point.

This brings us to the next point that the presentation introduces, which is regularized methods. The idea behind regularized methods is that if there is more information provided about the object, there are more specialized techniques that can be developed that caters to the information provided. One of these assumptions is that objects are generally smooth. In this way, the reconstruction problem can be modelled as an optimization problem, in which the final model should have smallest surface area possible because smooth surfaces will generally have smaller surface areas than more rough surfaces. One method of solving the reconstruction problem, with the assumption that objects should be as smooth as possible is begin with a set of smooth surfaces and evolve the surfaces into the object. This can be achieved by utilizing and integrating partial differential equations through the level set method to obtain the final result. In the level set method, a distance metric is used to characterize a surface in relationship with grid points. The model that minimizes this distance metric will be the final output. By using regularized methods, almost any object’s 3D model can be estimated from images. However, because this method assumes that objects are as smooth, edges and sharp corners are often times not preserved.

Another method of finding correspondences is through graph cuts. First, an energy function is defined. The objective is to minimize this energy function. First, all pixels in both images are assigned a label. The groups or matchings are constantly changed until the energy function is minimized. Pixel labels are changed either through alpha-beta swapping or alpha expansion. Alpha-beta swapping involves switching the labels of two distinct groups of pixels in one image. The idea is that the new groups will have a better fit with corresponding groupings in the other image (lower energy function). Alpha-expansion involves expanding a group of pixels of one label, by taking in pixels of other labels. This allows label groups to be of arbitrary size, so the method is very accommodating to most surfaces.

The final set of methods that the presentation covers is by reconstructing the 3D model piece-by-piece. In other words, small portions of the object is constructed part-by-part and then put together in the end. Generally, these methods save memory and time, when compared to some of the previous methods. However, the implementation is more complex and the end result may have outlines of the object that are not well-defined or do not correspond to the outlines from the input images.

Converting 2D to 3D: A Survey

The paper presents an overview of the various types of algorithms developed for constructing 3D models from 2D images. The third dimension of depth is required to go from 2D to 3D, and this is what most algorithms focus on computing. In order to compute depth, one or more depth cues are generally used, such as color, texture, or motion. The methods described in this paper are classified by the type of depth cue they employ to construct the 3D model. Furthermore, the algorithms discussed can be further classified into two groups: multi-ocular and monocular.

The first group of algorithms use multi-ocular depth cues, meaning that there are multiple views being used to reconstruct the model.

The first type of depth cue discussed in this category is disparity. The depth is computed by computing the disparity between pixels, which can be accomplished through the triangulation or by finding stereo correspondences. Additionally, global optimization algorithms generally attempt to globally optimize an energy function, which represents a constraint like smoothness. The methods used to optimize these energy functions are plentiful and include dynamic programming and graph cuts.

The second cue discussed is motion. The idea behind using this cue is that objects that are closer to the camera will move faster across the image plane than objects that are farther away. These algorithms attempt to estimate the motion field of objects as they move in the frames. One way to do this is by measuring optical flow, or the motion of brightness of the object. The primary constraint behind this method is that the apparent brightness of objects remains constant as objects move. Another method of measuring the motion field is to track small features between the frames as the object moves.

The third method discussed is defocus. This method generates a depth map based on the degree of blurring present in images. When a 3D point is projected onto an image plane and it is out of focus, the point is actually represented as a circular patch of pixels with constant intensity on the image plane. The circular patch is centered around a point, with a radius that changes with respect to the axes of the image plane. This radius is also known as the blur parameter Often times, this blur parameter is assumed to be constant within a given window. The blur is assumed to be caused by the convolution of the ideal point on the image plane and a camera point spread function, which is often times simply the Gaussian function. Once the blur parameter is known, then the distance of 3D points away from the camera can be computed. These methods require multiple images taken with cameras with different focal settings.

The fourth method discussed is focus. Focus involves taking images of a scene at different distances away from the camera, which simulates different focus levels. The idea is that if you start off further away from an object, the object will appear defocused. However, as you move the object closer and closer to the camera, it will become more focused, until it reaches maximum sharpness. At this point, moving the object closer to the camera will defocus the image. It is at the point right after the point of maximum focus and at the point of maximum sharpness that the depth map of each point on the object can be computed by taking the difference in value between these two distances away from the camera to interpolate each point of the object.

The final method in multi-ocular methods is depth-from-silhouette. These algorithms attempt to retrieve the contour that separates the object from the background. This requires multiple cameras for multiple views and the retrieved silhouettes are projected onto 3D space to form the model. The shape of the object can be retrieved by stitching together the silhouettes formed from each view, assuming the cameras are calibrated.

The second group of algorithms use monocular cues, meaning that only one image or view is used to construct the 3D model.

The first method described is depth-through-defocus. This method is very similar to its multi-ocular counterpart, the only difference is that one image is used. Multiple images are generally used to reduce the ambiguity in the blur parameter estimation when camera focal settings are not known. In the single image version, the blur parameters are estimated using a single image by applying a Gaussian second derivative filter. The resulting response has positive and negative peaks. The distance between consecutive positive and negative peaks, as well as the variance of the Gaussian filter, can be used to estimate relative depth.

The second method described is linear perspective. The idea behind this method is find the vanishing points in an image. Edge detection is employed to find the major edges in the image. The points at which the most edges intersect are deemed the vanishing point in the image. Areas that are closest to the vanishing points are deemed to be at a further depth, while areas furthest away from the vanishing points are the are the areas closest to the camera.

The third method described is depth through atmosphere scattering. The radiation and diffusion of small particles in the atmosphere affect the propagation of light through the atmosphere. This causes objects further away to be viewed as less distinct and more blue than objects closer. Through this, the relative depth map can be found.

The fourth cue described is shading. This method uses the variations in surface shading to find the depth map of the object. Generally, these methods use one of four light reflectance models to model the effects of light on the surfaces of object, with the Lambertian surface being the most common model due to its simplicity.

The fifth cue described is patterned texture. The idea behind this cue is that texture patterns vary in size depending on its distance and orientation with the image plane. Additionally, there are many surfaces in which there is a texture pattern, with areas of smooth surfaces around it, making this approach feasible. Texture patterns or texels appear larger the closer they are to the camera and smaller or shorter the further away they are from the camera.

The sixth cue described is bilateral symmetry. There are some objects that are bilaterally symmetrical, like the human face. The basic idea behind these methods is that if the object is split across its axis of symmetry, the resulting two halves can both be used as an image for computing stereo correspondences.

The seventh cue mentioned is occlusion. Occlusions can provide relative depth due to its property that it obscures part of the view of another object, meaning an object that is occluded is further away from the camera than the object that is occluding it.

The final set of methods described are statistical methods, or methods involving machine learning. This is achieved by first computing a set of global and local features at multiple scales. Depth, is calculated based on a statistical probability density function and the idea that pixels close to each other will have similar depths. Additionally, some pixels that are not close by can also have similar or related depth values, which is taken into consideration by evaluating at multiple scales.