Volume 5, Issue 3, March 2015 ISSN: 2277 128X
International Journal of Advanced Research in Computer Science and Software Engineering
Research Paper
Available online at: www.ijarcsse.com

Robust Object Detection and Tracking Using Sift Algorithm

Shweta Yakkali*, Vishakha Nara, Neelam Tikone, Darshan Ingle
SIESGST, Mumbai University
Maharashtra, India

Abstract*— The basic and important aspect of computer vision is matching an image which includes object and scene recognition. This paper shows the detection and tracking of unknown objects in a live camera feed. The object is defined by its location and extent in a single frame. In every frame that follows, the task is to determine the object's location and extent or indicate that the object is not present. In this paper, the object recognition system which can resolve the difficulty of rotations of object, scale changes and illumination are resolved with the help of "SIFT algorithm". It is implemented with different phases such as scale space extreme detection, key point localization. To evaluate the performance of the SIFT Algorithm we considered adding a counter to keep a track of the number of matching lines of the keypoints.

Keywords*— SIFT Algorithm, keypoints, descriptors, Euclidean distance, descriptor matching

I. INTRODUCTION

Feature Matching is an obstacle in computer systems. For analogous images, simple corner detectors can be used for image and feature matching. But for images of different scales and rotations, the Scale Invariant Feature Transform is essential. Our paper uses the SIFT algorithm abstracts features of an image in a way that is steady over image translation, rotation, scaling, illumination and camera viewpoint. SIFT algorithm is preferred as it is one of the most widely used algorithms for object recognition.

The SIFT algorithm accepts an image as input and recognizes the set of keypoints and determines its descriptors. This describes the implementation of the Scale-Invariant Transform Feature (SIFT) detector and descriptor. The implementation is designed to produce results compatible to Lowe's version. It is designed in Microsoft Visual Studio 2013 using the library: Emgu CV – OpenCV in .NET. We seek to clarify SIFT's ambiguities by replicating the algorithm in Microsoft Visual Studio 2013 and then making further improvements to the code. The SIFT detector excerpts from an image a collection of frames or keypoints. These are oriented rings on the surface of the image. As the image translates, rotates and scales, the frames track these rings and their changes. By mapping the frames to a reference, the result of such distortion on the feature appearance is removed. The descriptors so produced are invariant to translations and rotations are designed to be robust to residual small distortions. The features must be at least partially invariant to illumination, 3D projective transforms, and common object variations. The Sift algorithm uses a pipeline of functions, the scale-space extrema detection use difference-of-Gaussian function. Further the keypoint localization is performed wherein sub-pixel location and scale fit to a model. The next step is orientation assignment. Thereafter the keypoint descriptor is created from local image gradients.

The nearest neighbour matching of local image descriptors in a set of image descriptors are calculated from two different images, these image descriptors can be mutually matched by for each point finding the point in the other image domain that minimizes the Euclidean distance between the descriptors represented as 128-dimensional vectors. For better results and accuracy determination, matches for which the ratio between the distances to the nearest and the next nearest points is less than 0.8 are accepted.

After the designing of the user interface of the project, the SIFT algorithm is initially implemented on static images for the ease of calculation of keypoints and to check the accuracy of detection. The image which is to be searched(to fin image) and background image where the to find image will be searched(to find image) are fed. And the results are generated for static image. Subsequently, it is extended to work on the live video feed on the webcam since each frame can be treated as a static image which is extracted and each frame is tested for the object to be tracked. The inconvenience of creating and maintaining a database is removed as we detect the images in live feed and thus no memory storage for the database is required.

II. RELATED WORK

The development of image matching by using a set of local keypoints can be traced back to the work of Moravec (1981) on stereo matching using a corner detector to select interest points. The Moravec detector was improved by Harris and Stephens (1988) to make it more repeatable under small image variations and near edges. The Harris corner has since been widely for many other image matching tasks. While these feature detectors are usually called corner detectors, they are not selecting just corners, but rather any image location that has large gradients in all directions at a particular scale. Zhang et al. (1995) showed that it was possible to use Harris corners over a long image range by using a correlation

window around each corner to select likely matches. At the same time, a similar approach was developed by Torr (1995) for long-range motion matching. Schmid and Mohr (1997) showed that invariant local feature matching could be extended to general image recognition problems in which a feature was matched against a large database of images. They also used Harris corners to detect interest points, but rather than matching with a correlation window, they used a rotationally invariant descriptor of the local image region. This allowed features to be matched under arbitrary orientation change between the two images. Furthermore, they demonstrated that multiple feature matches could accomplish general recognition under occlusion and clutter by identifying consistent clusters of matched features. However the Harris corner is very sensitive to changes in image scale, so it does not provide a good basis for matching images of different sizes.

A considerable amount of research has been done in identifying representations that are invariant to scale change. Crowley and Parker (1984) developed a representation that identified peaks and ridges in scale space, Shokoufandeh, Marsic, and Dickinson (1999) have provided more distinctive feature descriptors using wavelet coefficients, and Lindeberg (1993, 1994) has dealt with the problem of scale selection, which assigns a consistent and appropriate scale to each feature. Lowe's SIFT algorithm uses the idea of detecting local oriented features in scale space, which was first shown to be effective in Christoph von der Malsburg's use of oriented Gabor filters over different scales linked in a graph. Mikolajczyk and Schmid (2003) have shown that of several currently used interest point descriptors, SIFT descriptors are the most effective. Object recognition is widely used in the machine vision industry for the purposes of inspection, registration, and manipulation. However, current commercial systems for object recognition depend almost exclusively on correlation-based template matching. While very effective for certain engineered environments, where object pose and illumination are tightly controlled, template matching becomes computationally infeasible when object rotation, scale, illumination, and 3D pose are allowed to vary, and even more so when dealing with partial visibility and large model databases.

An alternative to searching all image locations for matches is to extract features from the image that are at least partially invariant to the image formation process and matching only to those features. Many candidate feature types have been proposed and explored, including line segments [6], groupings of edges [11], [14], and regions [2], among many other proposals. Zhang et al. used the Harris corner detector to identify feature locations for epipolar alignment of images taken from differing viewpoints

III. PROPOSED SYSTEM

3.1 METHODOLOGY

The aim is to develop an object recognition system which will lead to the usage a new class of local image features as shown in the Figure 1. We will use features that have similar properties as of neurons in inferior temporal cortex that areused for object recognition in primate vision and these features include invariance to image scaling, translation, rotation, partially invariant to illumination changes and affine or 3D projection. Features are efficiently detected through a staged filtering approach those to identify stable points in scale space. Image keys are created that allow for local geometric deformations by representing blurred image gradients in multiple orientation planes and at multiple scales. The keys are used as input to a nearest-neighbour indexing method that identifies candidate object matches. Final verification of each match is achieved by finding a low-residual least-squares solution for the unknown model features.

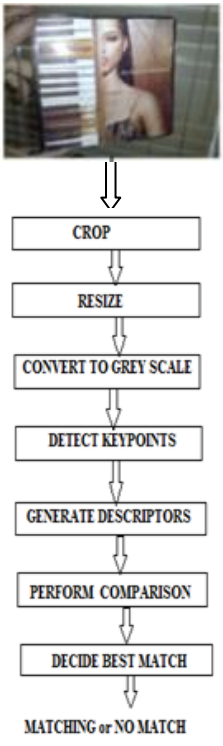

Figure 1: Block Diagram of Proposed System showing the flowchart of the methodology which will be followed.

3.2 SIFT DESCRIPTORS

Difference of Gaussians, neighbourhood maximization & minimization followed by magnitude & other filtering mechanisms are all used by the SIFT Algorithm to find unique locations on the image which are present throughout and hence should be found repeatedly under a variety of scales ,orientations and viewpoints. Moreover, the algorithm uses gradient magnitude and orientation maps to assign each of the keypoints with an orientation & scale which is used to create a descriptor of the patch around that keypoint. The proposed shared SIFT approach aims to use gradient and orientation maps from similar looking patches in a neighbourhood of the original keypoint from all of the descriptor training images to create a shared descriptor for that point. Figure 2 illustrates this process. We begin with the keypoints and these are individually extracted from all descriptor training images using the original SIFT approach. A patch correlation technique [2] is then used to find the most similar match to the patch surrounding the keypoint in the source image in all of the other descriptor training images. To avoid having to perfectly align the images and allowing sharing across images of slightly different sizes this search is conducted in a neighbourhood of the location of the keypoint in all image in all of the other descriptor training images. In the next step, gradient magnitude and orientation maps at these similar patches are combined to give orientation and scale information to that keypoint

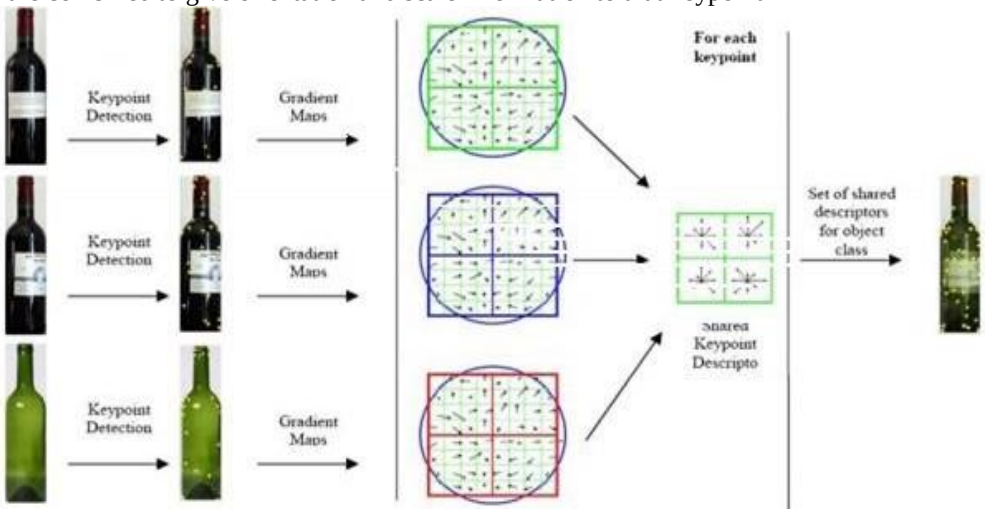

Figure 2: The above Figure shows how the shared SIFT algorithm works in the first stage keypoints of the objects are detected and next for each keypoint there is shared keypoint descriptor and in the end the final set of shared descriptors is generated.

Further, instead of creating separate histograms for each of the individual patches in the images, the keypoints information computed above and all the gradient maps from similar patches are used to create a shared or averaged histogram which provides a smoothened common representation of all the patches being shared. The shared sift descriptor for that location is then computed from this shared histogram. One major advantage of sharing patches across the descriptor images is that variance metrics can be used to judge how well the shared patch represents the original patches in the descriptor images.

Figure 3 shows the scene in which a blue bottle is taken and the algorithm gives all the keypoints pertaining to this object and also the objects which find relevance to the bottle in terms of colour, luminance etc. The blue points shown in the image are the keypoints pertaining to the object.

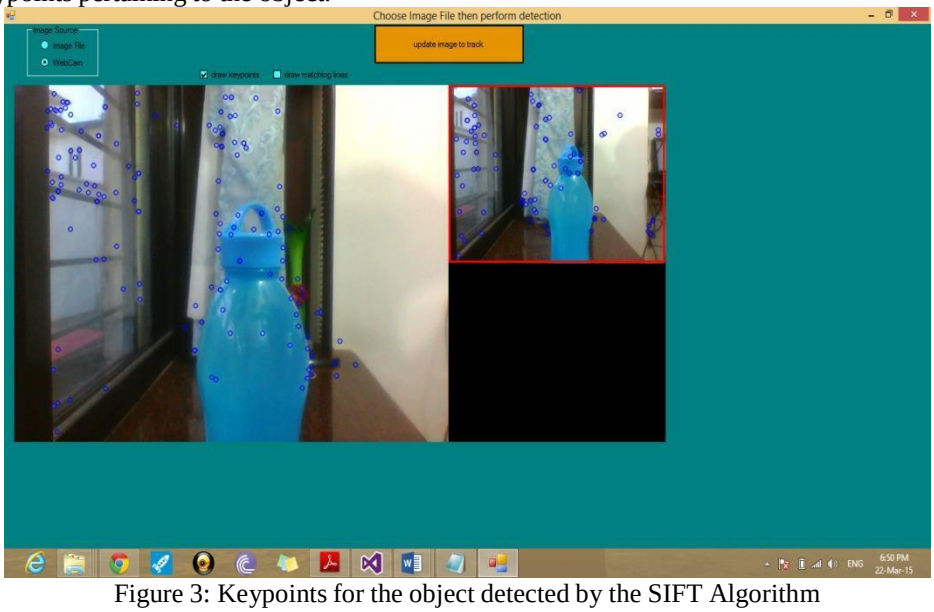

Figure 3: Keypoints for the object detected by the SIFT Algorithm

It is quite intuitive to see that a shared descriptor for the keypoints will be much closer in the 128 dimension space (All SIFT descriptors are 128 length vectors) to the keypoints it represents and similarly the shared descriptor that represents the keypoints in blue in Figure 5.2.3 will be further away from the keypoints it represents in feature space. The variance metric computes the average distance between a shared descriptor and the individual patches in the descriptor training images that it represents. The distance is a Euclidean distance computed in the 128 dimensional spaces using the equation no. 1.

$$Dist(d_1, d_2)) = \sum_{i=1}^{128} || {d_1}^i - {d_2}^i ||$$ … Equation no. 1

Where d_1 and d_2 represent any 128 dimension descriptors used in SIFT.

$$var(d) = \frac{\sum_{i=1}^{m} Dist(d, f_i)}{m}$$ … Equation no. 2

Where d represents the 128 dimension shared descriptor and f represents the set of m descriptors form the m patches it represents.

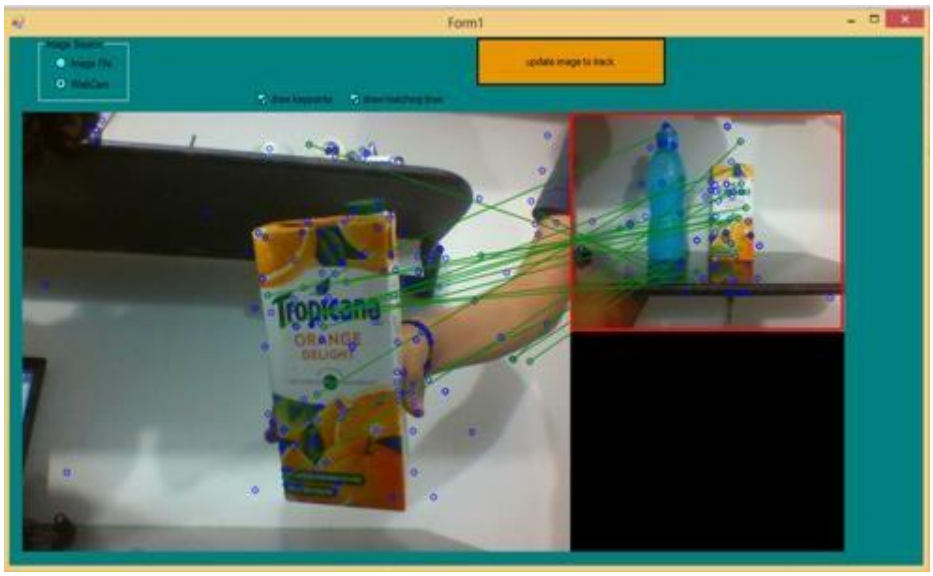

Figure 4: Keypoints and the matching lines for the object in the frame

3.3 MATCHING DESCRIPTORS IN FEATURE SPACE

In the original SIFT descriptor comparison approach, every descriptor in one image is compared to every other descriptor in the second image via the distance metric (Euclidean distance) shown in Figure 5.2.4. The base approach being the same, the altered matching algorithm also takes into account the variance of the shared descriptor. When comparing a new image to the dictionary of features of an object, the dictionary is scanned to find the best match for each SIFT keypoint in the new image. Further, the variance of the shared descriptor which was found as the best match in the dictionary is used to decide whether the keypoint in the new image qualifies as a match or not. There were a lot of experiments carried out using a variety of different decision techniques to label a keypoint as a match or not. One successful technique labelled a keypoint as a match if in feature space the Euclidean distance to the closest shared descriptor was less than the variance of that descriptor, which implies that the given keypoint is close in feature space to the shared descriptor and the keypoints that it represents in feature space. Figure 3 shows the matches resulting from matching the mug dictionary on the right to a new mug which it has not been trained upon in the descriptor training stage.

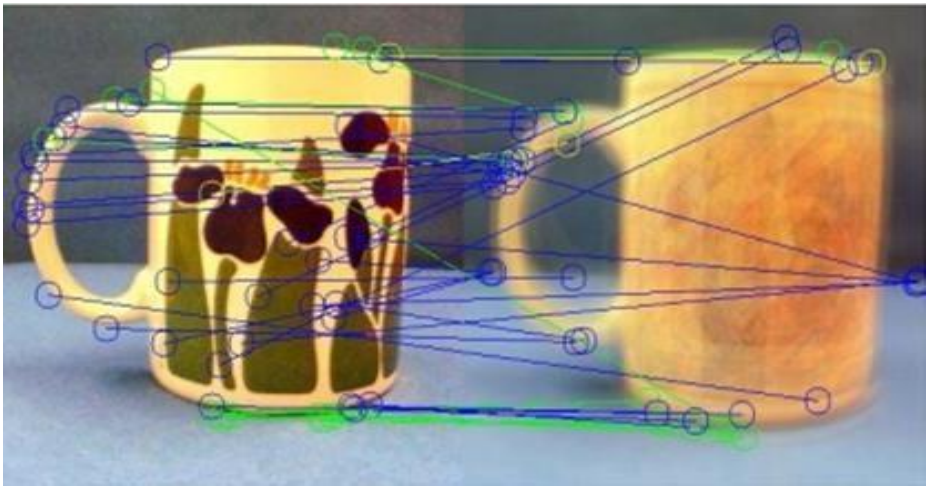

Figure 5: Matching pairs using a variance based threshold method between the mug dictionary and a new mug it has never seen before.

IV. IMPLEMENTATION

4.1 GUI DESIGN

In the user interface, there are two options from which the user can select. One being, "Image file" which takes two images as input and the other option is "Webcam" in which we select an image form the live feed of the webcam and the objects are matched from this image in the live feed. There is a checkbox to display the keypoints and matching lines. On selecting these checkboxes, the system will show the keypoints in both the scene image and the to find image. Also, the matching lines extend from the to find image to the scene image in the live feed indicating the matched keypoints. The user interface also comprises of the image box which shows the scene image and the to find image, in static image file option and the live feed video as well as the captured to find image. Ultimately, the Algorithm is performed on clicking the "Perform Algorithm" button.

4.2 SIFT ALGORITHM FUNCTIONS

The project has been coded in Microsoft Visual Studio 2013 using the library: Emgu CV – OpenCV in .NET. The image to find and the scene image are firstly converted to grayscale using imgSceneColor.Convert(Of Gray, Byte) and the coarse keypoints are detected using siftDetector.DetectKeyPointsRaw(). The brute force matching is performed to match the descriptor's indices and the nearest neighbouring keypoints.

The Features2DToolbox.DrawKeyPoints() function which draws the small circles on the locations of keypoints. And Features2DToolbox.DrawMatches will draw lines showing matched keypoints.

4.3 RESULTS

The following are the results obtained on the implementation of the Algorithm. Figure 6 shows the detection and tracking of the object which is the same blue bottle when the bottle is brought nearer to the web camera i.e. when the object is scaled. The Agorithm detects the object even after scaling.

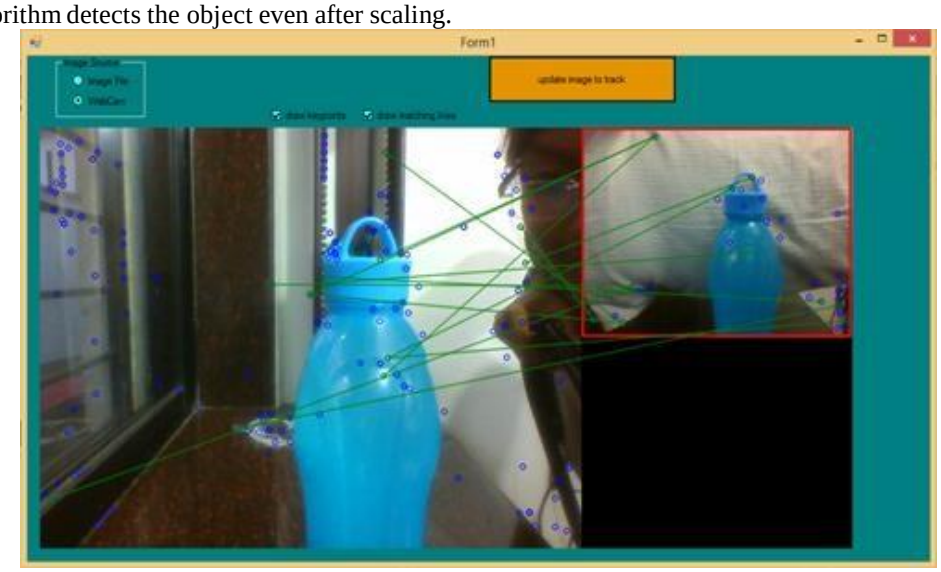

Figure 6: Detection and tracking of the object on scaling the object

Figure 7 illustrates the feature of the Algorithm which is that the Algorithm will still detect the object even if the object is taken away from the initial frame and is currently present in an all together different frame.

Figure 7: Detection of object when the object is in different frames.

V. CONCLUSION

SIFT can be used to discover analogous objects in two different images. The SIFT algorithm is evidently competent to recognize two objects as similar even the object is moderately concealed in either one of the images, has changed orientation, or the object is viewed at different angles. This approach transforms an image into a set of local feature vectors, each of which is invariant to image translation, scaling, and rotation, and partially invariant to illumination. and with the inclusion of counter, the accuracy percentage can be tracked. The implementation of such an algorithm could ease the computer vision.

As SIFT demonstrates numerous features, which are unique in object recognition field, the algorithm could then realize the demand of product quality control and object separation in industries. The SIFT features improve on previous approaches by being largely invariant to changes in scale, illumination, and local affine distortions. The large number of features in an image allow for robust recognition under partial occlusion in cluttered images. A final stage that excludes keypoints below a threshold value along with a counter is used to for accuracy determination.

ACKNOWLEDGEMENT

Authors gratefully acknowledge Professor Darshan Ingle for believing in us, believing in the idea and extending their support, guiding and helping us in every possible way.

REFERENCES

[1] Ballard, D.H., "Generalizing the Hough transform to detect arbitrary patterns," *Pattern Recognition*, 13, 2 (1981), pp.111-122.

[2] Basri, Ronen, and David. W. Jacobs, "Recognition using region correspondences," *International Journal of Computer Vision*, 25, 2 (1996), pp. 141–162.

[3] Beis, Jeff, and David G. Lowe, "Shape indexing using approximate nearest-neighbor search in high-dimensional spaces, *Conference on Computer Vision and Pattern Recognition*, Puerto Rico (1997), pp. 1000–1006.

[4] Crowley, James L., and Alice C. Parker, "A representation for shape based on peaks and ridges in the difference of low pass transform," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 6, 2 (1984), pp. 156–170.

[5] Edelman, Shimon, Nathan Intrator, and Tomaso Poggio, "Complex cells and object recognition," Unpublished Manuscript, preprint at http://www.ai.mit.edu/ edelman/mirror/nips97.ps.Z

[6] Grimson, Eric, and Thom´as Lozano-P´erez, "Localizing overlapping parts by searching the interpretation tree," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 9 (1987), pp. 469–482.

[7] Ito, Minami, Hiroshi Tamura, Ichiro Fujita, and Keiji Tanaka, "Size and position invariance of neuronal responses in monkey infer temporal cortex," *Journal of Neurophysiology*, 73, 1 (1995), pp. 218–226.

[8] Lindeberg, Tony, "Scale-space theory: A basic tool for analyzing structures at different scales", *Journal of Applied Statistics*, 21, 2 (1994), pp. 224–270.

[9] Lindeberg, Tony, "Detecting salient blob-like image structures and their scales with a scale-space primal sketch: a method for focus-of-attention," *International Journal of Computer Vision*, 11, 3 (1993), pp. 283–318.

[10] Logothetis, Nikos K., Jon Pauls, and Tomaso Poggio, "Shape representation in the inferior temporal co rtex of monkeys," *Current Biology*, 5, 5 (1995), pp. 552–563.

[11] Lowe, David G., "Three-dimensional object recognition from single two-dimensional images, "*Artificial Intelligence*, 31, 3 (1987), pp. 355–395.

[12] Lowe, David G., "Fitting parameterized three-dimensional models to images," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 13, 5 (1991), pp. 441–450.

