Estimating relative world views

Fred Nicolls

Department of Electrical Engineering
University of Cape Town, South Africa
fnicolls@eng.uct.ac.za

Abstract

The problem of estimating the relative orientation between two views of a compact object in the world is addressed. It is assumed that the structure of the object can be approximated by an ellipsoid, although future work will attempt to address this limitation. An energy minimisation approach is used to estimate the orientations of the cameras that is consistent with the observed data. The formulation is intensity-based, since the objects under investigation are small and there are too few corners for feature-based methods to work reliably.

1. Introduction

Conventional dense stereo techniques require that cameras be calibrated, with known relative viewpoints between the scene and the cameras. However, we are interested in obtaining 3-D reconstructions for the situation where a single camera observes a rigid object moving within its view. In this case one needs to know the relative orientation of the moving object between two successive views of it. This is equivalent to estimating the relative orientation between two views of the object taken simultaneously from disparate cameras.

Estimating the structure and motion of a scene from multiple image views is a classical problem in computer vision, and effective methods have been proposed for its solution [3, 5]. For instance, if the camera intrinsics are known then an efficient solution can be obtained by estimating the essential matrix from point correspondences in the image views, extracting the relative orientation of the cameras, and using dense stereo methods on the result. If the camera intrinsics are not known then they can be estimated via autocalibration, as long as there are are sufficiently many views of the object available and the camera settings are not changed.

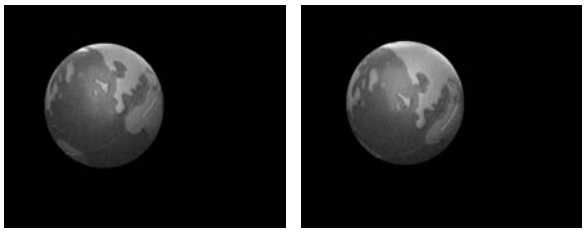
Dense stereo methods are commonly applied to approximately planar scenes. In this case there is a natural coordinate system in which to formulate the problem: each pixel in one of the images (the reference image) is assigned a depth value, indicating the distance to the surface of the object along the ray corresponding to the pixel. This is called a 2.5-D representation, and is limited in that the observed surface depth may only be a single-valued function of the coordinate grid.

In this work we are interested in building 3-D models of small and compact objects. The fact that they are small means that standard corner detectors do not find sufficiently many interest points for a good estimate of the essential matrix to be obtained. Simple methods therefore cannot be used to estimate the relative locations of the cameras imaging the object. We therefore formulate methods that explicitly use image intensity values, rather than geometric features. Secondly, our objects are compact and tend to be approximately spherical rather than planar. Simple depth-based stereo representations can therefore not be used: there is no planar coordinate system that can represent the object to a sufficient degree of accuracy. We therefore make the rather strong assumption that the object in the scene is an ellipsoid.

Under the approximations described, an energy function is constructed that quantifies the inconsistency in the observed images for any given configuration of ellipsoid and camera pair. The formulation resembles a Lukas-Kanade registration strategy [1], with the exception that no explicit functional mapping is found for points between the two camera views. A standard Levenberg-Marquardt optimiser is used to find that configuration of views and object that is most consistent with the observed image data.

An ellipsoidal object model is used for two reasons. Firstly, an ellipsoid has nice mathematical properties under perspective projection, with the image forming an ellipse in the camera view. Secondly, the compact objects that we are interested in often tend to be approximately elliptical, and it is hoped that we may be able to relax the measurement process to make this assumption less constraining. It is envisaged that the formulation presented may constitute an ”ellipsoid-plus-parallax” framework, in contrast to the powerful ”plane-plus-parallax” approaches found in the literature [4]. In many respects this work is most similar to [2].

Section 2 describes the dataset used in this work. A high-level description of the proposed algorithm is presented in



(a) Image 1 (b) Image 2

Figure 1: Two images of a globe used as a dataset

Section 3, with some important details regarding the measurement process and initialisation covered in Sections 4 and 5. The paper then concludes with some results.

2. Dataset

The dataset used in this work is comprised of two 640×480 greyscale images of a small globe of the world, captured at slightly different orientations with the same camera. These globes are hand-segmented into foreground and background to produce the images shown in Figure 1. Note that although the globe is at approximately the same location in each image, this is neither assumed nor required by the algorithm.

The reason for choosing a globe as the subject is that it matches the assumptions made in this work, namely that the object is ellipsoidal. Also, it has sufficient texture for estimates based on greyscale value to be meaningful.

Currently the method that is used requires knowledge of the intrinsic parameters of the camera. Since a good camera (with C-mount optics) was used, it is assumed that the principal point was at the centre of the image plane, with square pixels and zero skew. The camera intrinsics are therefore represented by the matrix

$$\mathbf{K} = \begin{pmatrix} f & 0 & i_w/2 \\ 0 & f & i_h/2 \\ 0 & 0 & 1 \end{pmatrix},$$

where the image has i_w columns and i_h rows. The focal length parameter was estimated from a rough calibration to be approximately $f = 1500$. Experience has shown that this parameter is not critical, as long as it is sufficiently large and both the size of the object in the world and the locations of the cameras are variable.

3. Algorithm outline

This section describes the basic algorithm outline, ignoring various complications that arise in practice. The section that follows describes the exceptions in detail.

We represent the object as a canonical ellipsoid (axes aligned with the coordinate axes) at the origin of the coordinate system, and with size parameters a, b, c representing the radii. There are two cameras in the world, represented by the matrices

$$\mathbf{P}_1 = \mathbf{K}_1 \begin{pmatrix} \mathbf{R}_1 & \mathbf{t}_1 \end{pmatrix} \text{ and } \mathbf{P}_2 = \mathbf{K}_2 \begin{pmatrix} \mathbf{R}_2 & \mathbf{t}_2 \end{pmatrix}.$$

The size of the ellipsoid is unknown, as is the rotation and translation of each camera view. A minimal parameterisation of the problem therefore has 15 degrees of freedom: 3 size parameters for the ellipsoid, and 3 rotation and three translation parameters for each camera. Rotation parameters in 3-D are notoriously badly behaved, so we choose to represent each rotation using 4 quaternion parameters. Thus the overall configuration can be represented by the unknown vector $\mathbf{a} \in \mathbb{R}^{17}$.

One of the benefits of using an ellipsoidal representation of the object is that there is a simple closed form expression for its image under perspective projection. For example, [6] shows that an ellipsoid in the world can be represented by a 4×4 matrix Q (in homogeneous coordinates), with a well-defined and easily calculated projection as an ellipse in any image. For a given camera and world configuration $\mathbf{a}$, we can therefore also easily find the extremities of the projected ellipsoid in any image.

The basic measurement process proceeds as follows. Choose a location $\mathbf{x}$ in the first camera view. For this location, use the assumed camera parameters to back-project the point, and find the 3-D point $\mathbf{X}$ where it intersects the assumed location of the world ellipsoid. Then take this intersection point and project it into the second camera view, also using the assumed camera parameters, to give a point $\mathbf{x}'$ in image 2 corresponding to $\mathbf{x}$. If the intensity in the first image is given by $I_1(\mathbf{x})$ and the intensity in the second image by $I_2(\mathbf{x}')$, then the error corresponding to $\mathbf{x}$ is given by

$$e_1(\mathbf{x}) = I_1(\mathbf{x}) - I_2(\mathbf{x}').$$

Every location $\mathbf{x}$ chosen in the first camera view therefore generates a corresponding greyscale consistency error value.

The set of locations in the image at which errors are calculated is explicitly specified. The option that is used in this work is to use the segmentation map of the object to determine the grid: a uniform (but possibly subsampled) rectangular grid of locations in the image is chosen that covers the rectangular bounding box of the foreground object in the image. This constitutes an image-based coordinate system, although it should be possible to use a grid determined by the current world model if desired. If there are n_1 grid locations, then we have an n_1-dimensional vector $\mathbf{e}_1(\mathbf{a})$ of errors for any given geometrical configuration $\mathbf{a}$.

The total error is taken to be the sum of squared errors over the defined set of image locations: $E_1(\mathbf{a}) =$

$\mathbf{e}_1(\mathbf{a})^T\mathbf{e}_1(\mathbf{a})$. To maximise the consistency of the configuration, we would like to minimise the error $E(\mathbf{a})$ over all $\mathbf{a}$.

It is additionally desirable to have a symmetric error function, so that swapping the images does not change the results. A simple way to do this is to repeat the procedure described with the roles of each image reversed. This will produce an additional set of errors $\mathbf{e}_2(\mathbf{a})$, each component of which corresponds to a grid location in image 2. A combined set of errors can then be formed by concatenation: $\mathbf{e}(\mathbf{a}) = [\mathbf{e}_1(\mathbf{a})^T \ \mathbf{e}_2(\mathbf{a})^T]^T$, with the total error given by $E(\mathbf{a}) = \mathbf{e}(\mathbf{a})^T\mathbf{e}(\mathbf{a})$.

To minimise the error, we perturb the parameters numerically, one at a time, and estimate the derivatives of the error components with respect to the parameters. Parameter updates are then obtained using a Levenberg-Marquardt iteration, where the coefficient matrix to be inverted is of size $p \times p$. Since is our case p is 17, this is easily computable. The process is then repeated until convergence.

The basic measurement process described cannot be implemented, due to inconsistent geometry in an arbitrarily specified set of parameter values. For example, the back-projected ray from a given pixel location may not intersect the world ellipsoid, in which case a corresponding location cannot be found in the second image. The measurement process therefore has to be modified to take into account exceptional circumstances. These modifications are described in the next section.

4. Details of the measurement process

As described in the previous section, the basic measurement process is quite simple. What is more complicated is what to do when inconsistencies arise in the configuration corresponding to a set of parameters.

Consider the case where a point is chosen on the coordinate grid of the first image. If this point lies inside the silhouette of the object there are three exceptions that can occur:

1. The ray back-projected from the point in the first image does not intersect with the model of the 3-D ellipsoid in the world. There is therefore no corresponding point in the second image. In this case we use a geometric measure of inconsistency as the error: the closest consistent point is found in the image (based on the known projection of the ellipsoid), and the distance to this point is taken to be the error.

2. The back-projected ray from the first image intersects the object, but the object itself occludes the intersection point in the second camera view. This does not correspond to an inconsistent geometry, so we take the corresponding error to be zero.

3. The back-projected ray from the first image corresponds to a valid point transfer across the ellipsoid, but the projected point does not lie within the observed silhouette in the second image. In this case the error is taken to be the minimum distance between the projected point and the boundary of the object in the second image. This is found using a distance transform on the segmentation of the second image.

If the point in the first image lies outside of the silhouette of the object, then for a correct configuration the back-projection of this point will not intersect the world ellipsoid. If this is observed to be the case, then a zero error is returned. However, if the ray does intersect the ellipsoid, then the error is taken to be the minimum distance in image space between the point and the projection of the ellipsoid.

The methods described above attempt to avoid discontinuities in the error function, since a large degree of inconsistency results in larger errors than a small degree of inconsistency. However, it must be noted that the errors as defined above are all geometric, in contrast to the intensity errors that result from the case where there is a valid point transfer across the ellipsoid. This is inconsistent, and highlights a severe shortcoming of the method proposed. In practice these two types of errors can be weighted differently in the cost function, but a more principled solution would be desirable.

It may be possible to only use the errors corresponding to valid intensity transfer, and still obtain a cost function that attains a minimum only for the correct parameter values. Initial attempts to do this have failed, but the possibility has not been excluded outright. This issue is further complicated by the need to numerically estimate derivatives with respect to parameters — when perturbing the parameters it may happen that the perturbation causes errors corresponding to valid transfer to become invalid and therefore unavailable. If this is the case, then a possible solution is to do *all* the perturbations, and only to include errors at those locations that are common to all parameters.

Figure 2 depicts important components of the measurement process for one geometric configuration. Shown on the left is the actual image data obtained from the first camera. The second image contains pixel values extracted from image 2, but at locations determined by grid points of view 1 determined by the current geometry. Alternatively, one can think of the second image as being the data from view 2 referred to view 1, where the implied warp is via 3-D transfer across the world ellipsoid. The third image contains the errors for the configuration: for points with valid transfer the error is just the greyscale difference between the previous two images; for pixels with no valid transfer, errors are the geometric distance

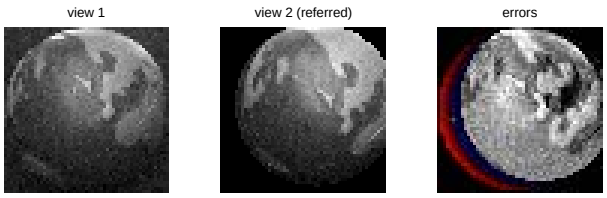


Figure 2: Image, warped image, and error map for some geometric configuration.

transform values described earlier. A zero pixel value transfer error in this image is indicated by a mid-range grey level, so that both positive and negative errors can be seen.

5. Initialisation

Since the method proposed in this work is based on optimisation, an initial specification of both the ellipsoid and the camera locations with respect to the world coordinate system is required.

Two assumptions are used for initialisation. Firstly, the object is initially taken to be a sphere in the world. It is hoped that this is a good enough starting point for the subsequent optimisation to succeed. Secondly, the two views of the object are assumed to be taken from nearby locations to one another. In this case it is reasonable to initialise the second camera at the same position and orientation in the world as the first. Note that the accuracy of this assumption can be improved by capturing image data at a higher frame rate.

Under the assumptions suggested, all that is required is an initial specification of the radius of the spherical object in the world, and the position of the first camera used to image this sphere. Consider the case where the camera is canonical and at the origin, and we want to position the object at an appropriate position in the world. A simple sphere with centre X_c and radius r_w (coordinate axes aligned with the world coordinate frame) can be represented by the quadric equation

$$\mathbf{X}^T\mathbf{Q}\mathbf{X} = \mathbf{X}^T \begin{pmatrix} \mathbf{I} & -\mathbf{X}_c \\ -\mathbf{X}_c^T & \mathbf{X}_c^T\mathbf{X}_c - r_w^2 \end{pmatrix} \mathbf{X} = 0.$$

Suppose we observe that the image point $\mathbf{x}_0$ lies on the edge of the projected ellipse. The corresponding ray in the world can be written as $\mathbf{X}(\alpha) = \alpha\mathbf{x}_0$, which we know must touch the sphere at exactly one point. This point of contact must satisfy

$$\alpha = -\frac{c}{\mathbf{b}^T\mathbf{x}_0} = \frac{(\mathbf{X}_c^T\mathbf{X}_c - r_w^2)}{\mathbf{X}_c^T\mathbf{x}_0}$$

to be a valid tangent, so the corresponding point in the world is

$$\mathbf{X}_0 = \frac{(\mathbf{X}_c^T\mathbf{X}_c - r_w^2)}{\mathbf{X}_c^T\mathbf{x}_0}\mathbf{x}_0.$$

Since the sphere has radius r_w^2 the distance between this point and its centre is fixed:

$$\left|\mathbf{X}_c - \frac{(\mathbf{X}_c^T\mathbf{X}_c - r_w^2)}{\mathbf{X}_c^T\mathbf{x}_0}\mathbf{x}_0\right|^2 = r_w^2.$$

This provides one constraint on $\mathbf{X}_c$.

The central projection of a sphere is in general an ellipse. For practical cameras it seems a good approximation to assume a circular projection, with the centre of the world ellipsoid projecting to the centre of the circle in the image. If this centre appears at $\mathbf{x}_c$ in the image, then we know that the centre of the sphere must lie on the ray $\mathbf{X}(\beta) = \beta\mathbf{x}_c$. Thus we must have $\mathbf{X}_c = \beta\mathbf{x}_c$ for some value of β. We can substitute this into the equation above and solve for β.

The resulting equation factors as

$$(\beta^2\mathbf{x}_c^T\mathbf{x}_c - r_w^2)[(\beta^2\mathbf{x}_c^T\mathbf{x}_c - r_w^2)\mathbf{x}_0^T\mathbf{x}_0 - \beta^2(\mathbf{x}_c^T\mathbf{x}_0)^2] = 0,$$

so two sets of solutions are obtained:

$$\beta^2 = \frac{r_w^2}{\mathbf{x}_c^T\mathbf{x}_c} \quad \text{or} \quad \beta^2 = \frac{r_w^2(\mathbf{x}_0^T\mathbf{x}_0)}{(\mathbf{x}_c^T\mathbf{x}_c)(\mathbf{x}_0^T\mathbf{x}_0) - (\mathbf{x}_c^T\mathbf{x}_0)^2}.$$

Solutions to the first equation result in the tangent point in the world being at $\mathbf{X}_0 = \mathbf{0}$, which in this context is a degenerate solution. The other solutions are the ones required: as expected there is a positive one and a negative one, corresponding to points in front of and behind the camera respectively. We choose the positive solution.

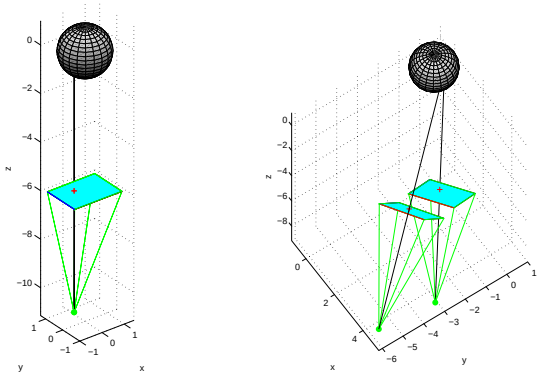
Repositioning the coordinate frame so that the sphere is at the origin simply involves a translation. It is easy to show that a suitable choice for the transformed camera is $\mathbf{P}' = [\mathbf{I}\ \ \mathbf{X}_c]$, and the resulting quadric is

$$\mathbf{Q}' = \begin{pmatrix} \mathbf{I} & \mathbf{0} \\ \mathbf{0}^T & -r_w^2 \end{pmatrix}.$$

6. Results

The method described in this paper was implemented, and applied to the images shown in Figure 1. The symmetric error described in Section 3 was used, on a regular grid of pixel positions covering the bounding boxes of the foreground image regions. The points of this grid were taken at spacings of 5 pixels in both the vertical and horizontal directions, mainly to reduce the amount of computation that has to be done.

A simple multiresolution approach was included into the optimisation, whereby the the Levenberg-Marquardt iterations described were in fact applied to blurred versions



(a) Initial (b) Final

Figure 3: Geometrical configuration for ellipsoid and two cameras.

of the images. After convergence at a high level of blur, this level was decreased and the iterations restarted. The sequence of blurring kernel sizes was $31, 15, 7, 5, 3, 1$, with 10 optimisation iterations run at each level. The total processing time was of the order of a few minutes. No attempts have yet been made to investigate the extent to which all of these steps are truly required — it is almost certain that a good solution can be obtained with far less computation.

Figure 3 shows the geometry of the initial and final configurations of cameras and object for the implemented solution. Initially both cameras are positioned at the same point in the world, so only one is visible.

Figure 4 shows camera views, referred camera views, and error images for both the initial solution and final solution. For the six images corresponding to the initial configuration, the first three relate to the transfer error for the reference frame defined by the first camera view, and the second three to that of the second camera view. Similarly for the images corresponding to the final configuration. At convergence the error map is seen to be approximately homogeneous (at all points except those of high texture gradient), indicating a near-zero error in most of the image.

7. Conclusion

A method of estimating the relative viewpoints for a pair of images of a compact object has been proposed. It uses the assumption that the object in the world can be approximated by an ellipsoid, and minimises a criterion based on direct appearance of the object in the camera views. The method works quite well, although in the current implementation the convergence is quite slow.

Future work will look to relaxing the structure requirement that the object be an ellipsoid. It is envisaged that the method may be useful for initialising a direct structure and motion estimation procedure for small objects with limited feature points. Within the framework it is also possible to compensate for lighting variation and highlights, which could improve the performance significantly.

8. References

[1] Simon Baker and Iain Matthews. Lukas-Kanade 20 years on: a unifying framework. Part I: the quantity approximated, the warp update rule, and the gradient descent approximation. *International Journal of Computer Vision*, 56(3):221–255, 2004.

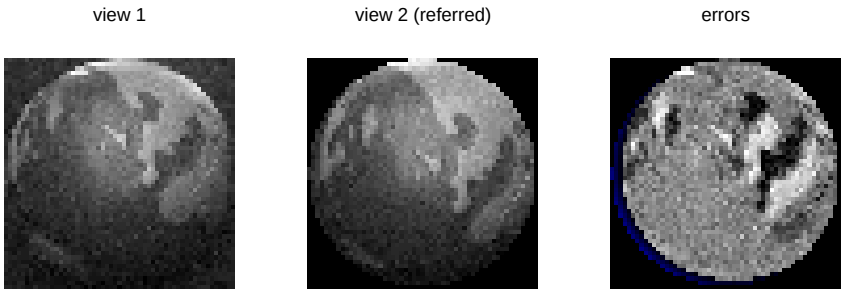
[2] K.J. Hanna and N.E. Okamoto. Combining stereo and motion analysis for direct estimation of scene structure. pages 357–365, 1993.

[3] Richard Hartley and Andrew Zisserman. *Multiple view geometry in computer vision*. Cambridge University Press, second edition, 2003.

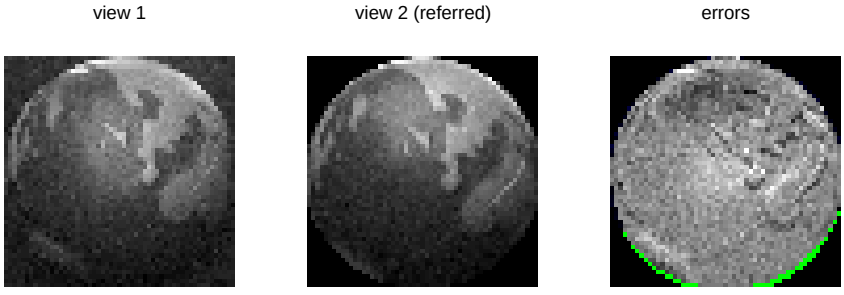
[4] R. Kumar, P. Anandan, and K. Hanna. Shape recovery from multiple views: a parallax based approach. In *DARPA Image Understanding Workshop, Monterey, California*, 1994.

[5] Marc Pollefeys, Luc Van Gool, Maarten Vergauwen, Frank Verbiest, Kurt Cornelis, Jan Tops, and Reinhard Koch. Visual modeling with a hand-held camera. *International Journal of Computer Vision*, 59(3):207–232, September 2004.

[6] B. Stenger, P. R. S. Mendonça, and R. Cipolla. Model-based 3D tracking of an articulated hand. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume II, pages 310–315, December 2001.



(a) Initial



(b) Final

Figure 4: Images, referred images, and errors for configuration, both initial and final.