Windtunnel model position and orientation

R. Stoffer, C. Stolk, S.W. Rienstra, J.K.M. Jansen

Abstract

In this contribution the determination of the position of moving and deforming objects in windtunnels from CCD camera information is studied. An analytical approach is discussed which solves the problem directly from manipulating nonlinear distance formulae. Also a least-squares approach is given, which is most convenient to implement from a numerical point of view.

Keywords

Windtunnel, CCD-camera, Position data-analysis.

1 Introduction

An object, e.g. an aircraft, is placed in a wind tunnel on supports. When there is no air flow in the tunnel, its position and orientation are well-defined. However, aerodynamic forces due to the flow of air through the tunnel may cause rotations or translations of the aircraft. In addition, the object is not completely rigid, but it may deform, e.g. its wings may bend up and down or twist. In order to be able to process the pressure and velocity data, one needs to know the position and orientation of the object as accurately as possible. For this purpose, a black and white CCD camera is available.

For the determination of the position and orientation of the object, well-defined reference points on the object are necessary. One might think of wing tips and the tail of an aircraft. However, we will assume that markers have been placed on the object, whose three-dimensional position on the model is known. We will also assume that we know which marker on the camera picture corresponds to which marker on the object. In this paper, we will describe two ways to determine the position and orientation of the model: an analytical method and a numerical least-squares approximation. Also we make some remarks about related questions.

2 Problem definition

A camera is placed at the origin. Its viewing direction is the positive y axis. All coordinates are made dimensionless on the coordinate of the camera plane. The object in the wind tunnel is rotated and translated. So, a 3-dimensional point in the model reference frame is first rotated along the angles α, β and ω, also known as pitch, yaw and roll which correspond to rotations along the y, z, x axes, respectively. It is then translated by the vector $\mathbf{t} = (x_t, y_t, z_t)$. The total transformation from a vector $\mathbf{v}^o$ in object space to $\mathbf{v}^c$ in camera space thus becomes

$$\mathbf{v}^c = \mathbf{T}(\mathbf{v}^o) = \mathbf{t} + R_\omega R_\beta R_\alpha \mathbf{v}^o, \tag{1}$$

where the R-matrices [?] are rotation matrices for the angles α, β, ω, given by

$$R_\alpha = \begin{pmatrix} \cos(\alpha) & 0 & -\sin(\alpha) \\ 0 & 1 & 0 \\ \sin(\alpha) & 0 & \cos(\alpha) \end{pmatrix},$$

$$R_\beta = \begin{pmatrix} \cos(\beta) & -\sin(\beta) & 0 \\ \sin(\beta) & \cos(\alpha) & 0 \\ 0 & 0 & 1 \end{pmatrix},$$

$$R_\omega = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos(\omega) & -\sin(\omega) \\ 0 & \sin(\omega) & \cos(\omega) \end{pmatrix}. \tag{2}$$

A point $\mathbf{v}^c = (x, y, z)$ is projected on the $y = 1$ plane (equivalent to the camera space) according to

$$\mathbf{v}^p = \mathbf{P}(\mathbf{v}^c) = (p, q) = (\frac{x}{y}, \frac{z}{y}), \tag{3}$$

so the total projection operator becomes

$$\mathbf{v}^p = \mathbf{P}(\mathbf{T}(\mathbf{v}^o)). \tag{4}$$

A picture of the different types of coordinates is given in Figure **??**. For a set of points on the same

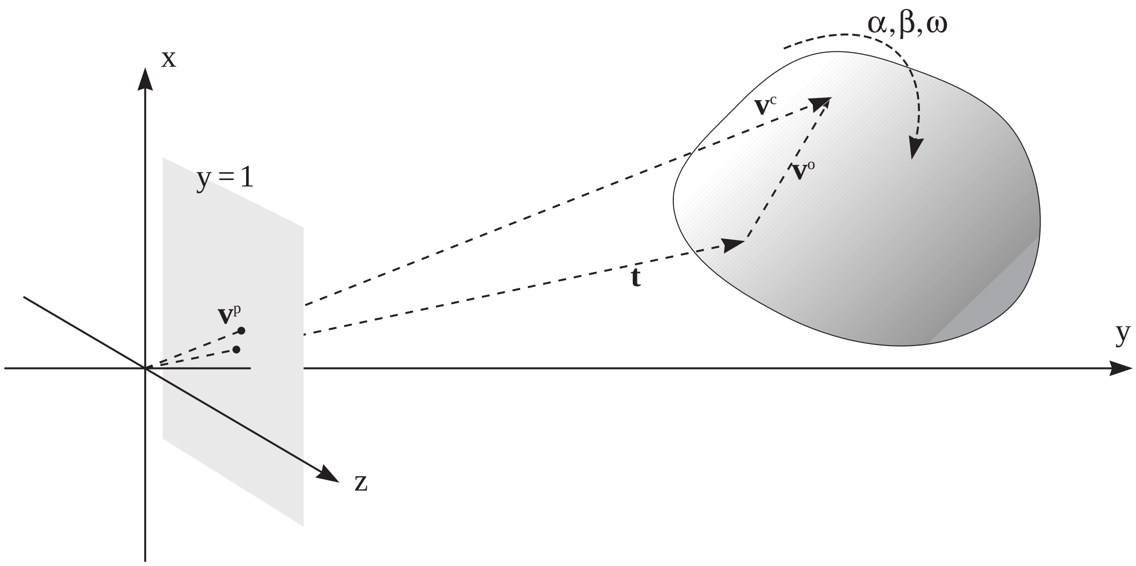

Figure 1: The different coordinates

object the angles α, β, ω and the translation vector $\mathbf{t}$ are the same (if the body is undeformed) or strictly related (if the body is allowed to deform in a particular way). Therefore, α, β, ω and $\mathbf{t}$ can be found from the projection information of a large enough number of points.

3 Analytical approach

The total number of unknowns is 6, so a measurement of the three points $(p_1, q_1), (p_2, q_2)$ and (p_3, q_3) should give a solution. However, this solution should be real and unique. We will investigate whether such a solution exists.

Since nothing is known a priori about the position and orientation of the object, the only information about the points is the distance between them in object space. Since the rotation and translation transformations are unitary, these distances remain the same in camera space. So we will use the distances to determine the 3-dimensional position of the points in camera space, after which the rotation and translation parameters can easily be deduced.

From the observation of the position (p_i, q_i) on the camera of a point, it is clear that, in camera space, the point must lie on a line parameterised by y_i

$$\mathbf{v}_i^c = y_i(p_i, 1, q_i). \tag{5}$$

For convenience, we will introduce the dimensionless parameters λ and μ

$$\lambda := \frac{y_2}{y_1}, \qquad \mu := \frac{y_3}{y_1}. \tag{6}$$

Then the distances between the points can be expressed in y_1, λ and μ

$$\begin{aligned} d_{12} &= y_1\sqrt{(p_1 - \lambda p_2)^2 + (1-\lambda)^2 + (q_1 - \lambda q_2)^2} \\ d_{13} &= y_1\sqrt{(p_1 - \mu p_3)^2 + (1-\mu)^2 + (q_1 - \mu q_3)^2} \\ d_{23} &= y_1\sqrt{(\lambda p_2 - \mu p_3)^2 + (\lambda-\mu)^2 + (\lambda q_2 - \mu q_3)^2} \end{aligned} \tag{7}$$

By eliminating y_1 this can be reduced to the following two quadratic equations in λ and μ

$$\begin{aligned} &\frac{p_2^2+q_2^2+1}{d_{12}^2}\lambda^2 - \frac{p_3^2+q_3^2+1}{d_{13}^2}\mu^2 + \frac{p_1p_2+q_1q_2+1}{d_{12}^2}\lambda + \cdots \\ &\qquad + \frac{p_1p_3+q_1q_3+1}{d_{13}^2}\mu = -(p_1^2+q_1^2+1)\left(\frac{1}{d_{12}^2}+\frac{1}{d_{13}^2}\right) \end{aligned} \tag{8}$$

$$\begin{aligned} &\frac{p_2^2+q_2^2+1}{d_{23}^2}\lambda^2 - \frac{p_2p_3+q_2q_3+1}{d_{23}^2}\lambda\mu + \frac{p_3^2+q_3^2+1}{d_{23}^2}\left(1-\frac{d_{23}^2}{d_{13}^2}\right)\mu^2 + \cdots \\ &\qquad + \frac{p_1p_3+q_1q_3+1}{d_{13}^2}\mu = \frac{p_1^2+q_1^2+1}{d_{13}^2}. \end{aligned} \tag{9}$$

These equations describe curves in the λ-μ plane. The shape of these curves depends on the parameters: for a general quadratic equation of the form $ax^2 + bxy + cy^2 + dx + ey = f$, the sign of the quantity $D = b^2 - 4ac$ determines whether the curve is hyperbolic, parabolic or elliptic in nature. If D is positive, the curve is a hyperbola, if D is zero, it is a parabola and if D is negative, the curve becomes an ellipse. It is clear that equation (**??**) always describes a hyperbola; for equation (**??**), this depends on the parameters, although we can say with certainty that it will be a hyperbola when $d_{23} \geq d_{13}$.

In general, these two equations (**??**) and (**??**) yield four (possibly degenerate) solutions. If there are complex solutions, they will always be in pairs of complex conjugates, since all coefficients of the equations are real. A real-world solution is real; there may be 0, 2 or 4 real solutions. This means that the solution of this problem with three markers is not unique.

We will clarify this with an example: Take a triangle with its corner points at $\mathbf{v}_1^c = (-1,4,-1), \mathbf{v}_2^c = (-1,4,1)$ and $\mathbf{v}_3^c = (1,4,0)$. These will project at $\mathbf{v}_1^p = (-1/4,-1/4), \mathbf{v}_2^p = (-\frac{1}{4},\frac{1}{4})$ and $\mathbf{v}_3^p = (\frac{1}{4},0)$. However, if $\mathbf{v}_3^c$ were located at $(\frac{13}{17},\frac{52}{17},0)$, all sides of the triangle would still have the same lengths and $\mathbf{v}_3^p$ would still be the same. These solutions are two of the four possible solutions with this triangle and these projections.

In order to uniquely determine the solution, a fourth marker is necessary. Adding a fourth marker results in an over-determined problem. In the absence of errors this uniquely fixes the solution (when errors are present one has to minimize some measure of the total error, see the next section). The fourth marker should not be on an edge of the original triangle; then, the same problems can occur. In nearly all other cases, the fourth marker uniquely determines the solution. We will consider the following four markers in object space: $\mathbf{v}_1^o = (-1, 0.7, 0.3), \mathbf{v}_2^o = (0.5, -0.5, 0.6), \mathbf{v}_3^o = (0, 0.1, 0.6)$ and $\mathbf{v}_4^o = (-1, 0.3, 1.0)$. They are rotated along $\alpha = -0.3, \beta = 0.2$ and $\omega = 0.5$ and translated along $(0, 2, 0)$ into camera space. Two triangles are constructed from this; $\triangle_1$ consists of $\mathbf{v}_1^o, \mathbf{v}_2^o$ and $\mathbf{v}_3^o$, and $\triangle_2$ consists of $\mathbf{v}_1^o, \mathbf{v}_2^o$ and $\mathbf{v}_4^o$. The procedure described above is applied to determine the curves described by equations (7) and (8) for each triangle. For both triangles, the curves are hyperbolas. Figures **??** and **??** show the curves for respectively $\triangle_1$ and $\triangle_2$. Both triangles yield 2 solutions (cross-sections of the curves).

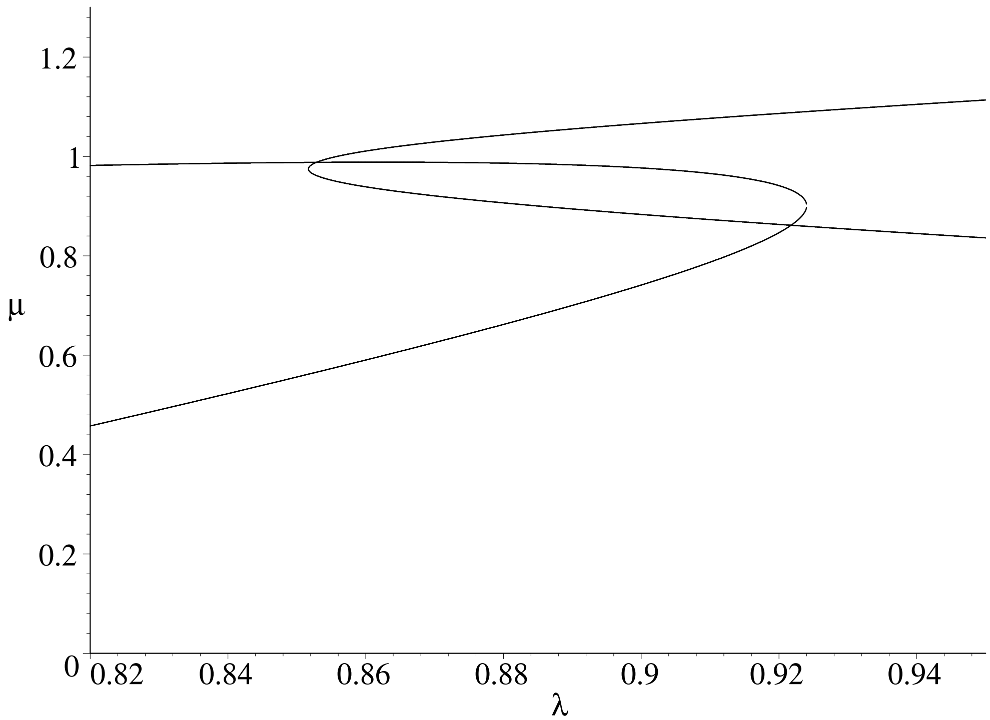

Figure 2: Relevant portion of curves of $\triangle_1$

Triangle $\triangle_1$ yields real solutions at $(\lambda, \mu) = (0.86925, 0.76122)$ and at $(0.77882, 0.98656)$; the solutions for $\triangle_2$ are at $(\lambda, \mu) = (0.77882, 1.13132)$ and $(1.12031, 0.52719)$. In both cases, λ is defined as y_2/y_1, so the solutions for which λ is equal are the solutions in which the calculated $\mathbf{v}_2^c$ is at the same position in both triangles. The calculated $\mathbf{v}_3^c$ is also the same in both triangles.

When the positions of the markers in camera space are known, the corresponding rotation matrix and translation can be easily calculated, and the rotation angles α, β, ω and the translation vector $\mathbf{t}$ are equal to the real values (up to 10^{-8}).

4 The method of least squares

The method of (nonlinear) least squares [**?**] can be used to reconstruct from observed data one or more parameters, in particular when there is more data than parameters. Let F be a map that maps the parameters $(\mathbf{t}, \alpha, \beta, \omega)$, and the marker positions in object coordinates $\mathbf{v}_i$ to the camera

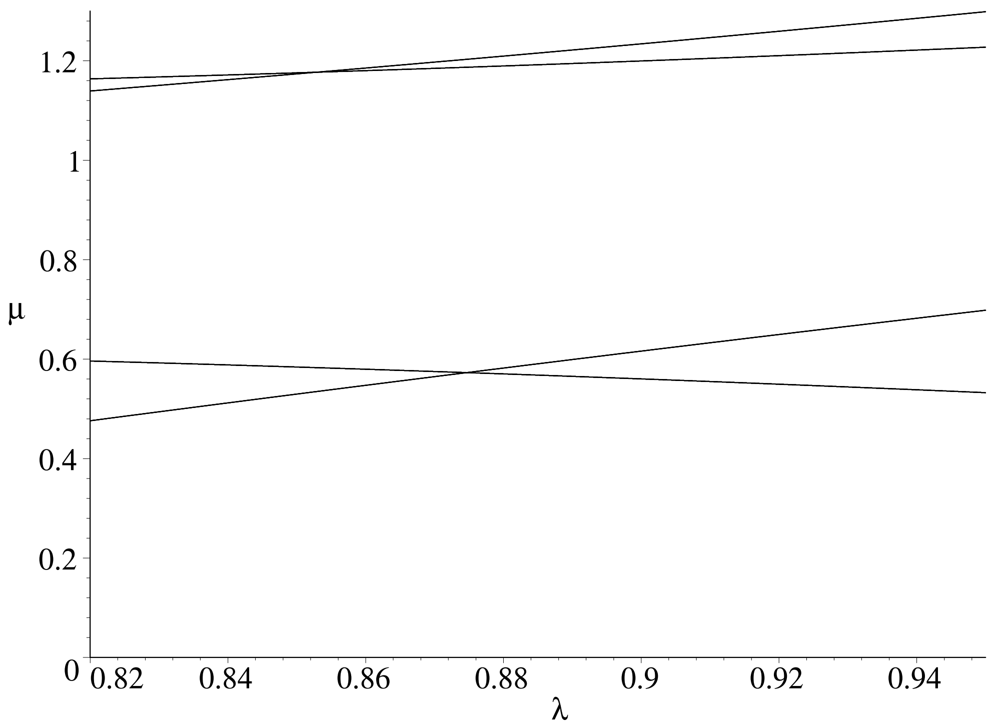

Figure 3: Relevant portion of curves of $\triangle_2$

points (p_i, q_i)

$$\begin{aligned}(p_i, q_i) &= F(\mathbf{t}, \alpha, \beta, \omega; \mathbf{v}_i) + \text{error} \\ &:= P(\mathbf{t} + R_\omega R_\beta R_\alpha \mathbf{v}_i) + \text{error}. \end{aligned} \tag{10}$$

Suppose there are k markers. Let the vector $\mathbf{d} = \{(p_1, q_1), \ldots, (p_k, q_k)\}$ be the data. We now try to find $(\mathbf{t}, \alpha, \beta, \omega)$ that minimize the squared error, i.e. the squared difference of observed and modelled data given by

$$\begin{aligned} E((p_1, q_1), \ldots, (p_k, q_k); \mathbf{t}, \alpha, \beta, \omega; \mathbf{v}_1, \ldots, \mathbf{v}_k) \\ := \sum_{i=1}^{k} [(p_i - p(\mathbf{t}, \alpha, \beta, \omega; \mathbf{v}_i))^2 + (q_i - q(\mathbf{t}, \alpha, \beta, \omega; \mathbf{v}_i))^2]. \end{aligned} \tag{11}$$

Note that the least squares method treats all data on equal footing, unlike the method of the previous section where we solved using three data points and then used the fourth point to determine which of the solutions was the correct solution. Therefore the least squares method is probably a better way of dealing with the data.

The method was invented by the famous German mathematician C.F. Gauss who used it to do a geodesic survey with a large precision. He knew that doing just enough measurements would result in an error that was far too large, so he had many measurements done and using the method least squares the individual errors would average out.

In order to minimize this error function E, one selects a suitable starting point and follows the "path of steepest descent" down to a minimum. In general such a function may have many

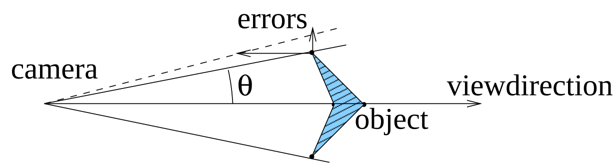

Figure 4: The ratio of the errors depends on the view angle θ.

minima, and this process will converge to a minimum that is close to the starting point. If we have a good starting point it is a very efficient method. In our case we take many images per second, and the parameters from the previous image will most likely be a good starting point. Therefore we think it is very important to use the information from the previous image.

Using the mathematical theory one should investigate when the least squares method converges to the correct minimum in a reasonable time. This depends on the behavior of the function E. It is also possible to obtain an estimate for the errors in the resulting position and orientation. We have not worked this out.

Using standard mathematical software such as Mathematica, Maple or Matlab it is not difficult to implement the method. We used Mathematica to calculate a few examples. We first did experiments with 3 markers. In this case the number of data equals the number of unknowns (so this is not really least squares, but it is equation solving and the minimum value of the error should be zero). We have seen above that there are in general four possible solutions in this case. If these are well separated, and the starting point is sufficiently close to the correct solution then the algorithm converges indeed to the correct solution. On the other hand, when the starting point is too far away from the correct solution, we obtain a wrong answer. Also there were cases where the convergence was relatively slow, which probably corresponded to either the situation of two minima close to each other, or a degenerate minimum.

After that we did experiments with four markers, where the fourth marker was not in the plane determined by the other three. In the examples we did, the algorithm converged to the correct solution.

It is possible to estimate the error in the result due to error in the observed data, simply by perturbing the correct data with some Gaussian error, and then comparing the parameters obtained from this data with the correct parameters. In fact we may take a set of perturbed data, and then look at the set of parameters that is obtained, and the shape of this set indicates what directions are sensitive to errors.

We used this procedure to compare the error in the camera direction with the error in the directions perpendicular to the camera direction. We expect that the ratio of error in camera direction and perpendicular to it is approximately the tangent of the view angle θ (see Figure **??**) To test this we put a triangle with coordinates $(-1,0,0),(0,0,2),(1,0,0)$ at position $(0,2,0)$ resp. $(0,20,0)$. The error clouds due to a Gaussian perturbation of the data with $\sigma = 0.02$ resp. $\sigma = 0.002$ are given in Figure **??**. In the first picture the ratio of errors in x, y coordinates is approximately 1, in the second it is approximately 10, in both cases it is approximately $2\tan\theta$.

5 Extensions of this work

When defining the problem above there are a number of aspects that we disregarded.

When the object is deformed it is not clear what should be called the position and orientation of the object. This introduces an error. Also we may want to determine the amount of deformation of the object. We suggest the following approach to this problem. The marker position in the object frame will now depend on the deformation. We assume that the position of the marker can be parametrized by some deformation parameter λ,

$$\mathbf{v}^o = \mathbf{v}^o(\lambda),$$

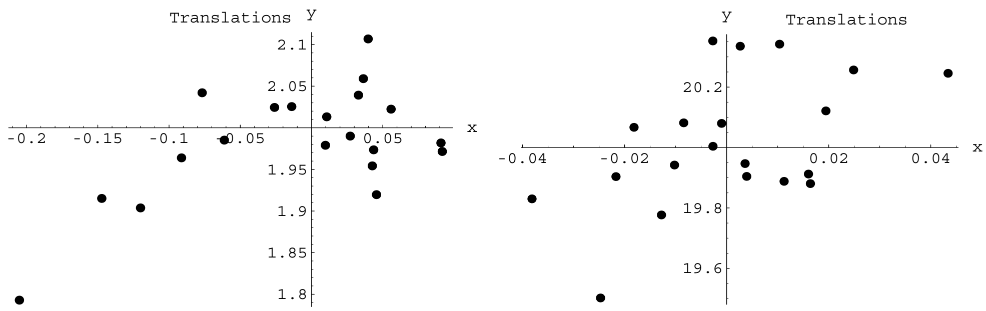

Figure 5: Reconstructed x, y translation parameters with Gaussian errors in the data.

In general when several types of deformation are possible we need m parameters $\lambda = (\lambda_1, \ldots, \lambda_m)$. Now the locations in the camera view depend also on the parameter λ

$$(p_i, q_i) = F(\mathbf{t}, \alpha, \beta, \omega; v_i(\lambda)).$$

Using the least squares algorithm we can also determine λ. This idea has been tested using our Mathematica notebook, in the situation of one point being displaced in z-direction. In the examples with four marker points, one of which was displaced due to assumed deformation, the displacement could be reconstructed, as long as it is not in camera direction (as one would expect).

Another problem is to match the observed markers with the markers on the object. This is complicated by the fact that sometimes markers may be out of the view of the camera, for instance because they are on the backside of the object or because they are hidden by an other part of the object. In addition, there may be spurious or undetected markers. We suggest to use the information of the previous image to solve this problem. From the previous image the location of the different markers is approximately known, and from this information one can obtain information about the matching of the markers. Also one can determine which markers are likely to disappear out of the view.

6 Other suggestions

Finally we suggest a few things that could be important in extending this work. One possible extension is the use of multiple cameras.

Secondly, we mention the detection of edges and corners. If that could be automated then there would be no need to attach any markers to the object.

A third thing to do would be investigating the literature for results on similar problems. We could think of geodesy, remote sensing, motion capturing.

References

[1] O. Bottema, *Theoretische Mechanica*, Epsilon Uitgaven, Utrecht, 1985.

[2] J. Stoer and R. Bulirsch, *Numerical Analysis*, Springer, New York, 1980.