Fitting 3D Data with a Cylinder

David Eberly
Magic Software, Inc.
http://www.magic-software.com

Created: February 25, 2003

This document describes an iterative algorithm for fitting a 3D point set with a cylinder. The assumption is that the underlying data is modeled by a cylinder and that errors have caused the points not to be exactly on the cylinder. You could very well try to fit a random collection of points, but no guarantees of convergence of the iterative algorithm to a meaningful solution.

1 Representation of a Cylinder

An *infinite cylinder* is specified by an axis containing the point $\mathcal{C}$ and whose unit–length direction is $\mathbf{W}$. The radius of the cylinder is $r > 0$. Two more unit–length vectors $\mathbf{U}$ and $\mathbf{V}$ may be defined so that the set of vectors $\{\mathbf{U}, \mathbf{V}, \mathbf{W}\}$ is a right–handed orthonormal set. That is, all vectors are unit–length, mutually perpendicular, with $\mathbf{U} \times \mathbf{V} = \mathbf{W}$, $\mathbf{V} \times \mathbf{W} = \mathbf{U}$, and $\mathbf{W} \times \mathbf{U} = \mathbf{V}$. Any point $\mathcal{X}$ may be written uniquely as

$$\mathcal{X} = \mathcal{C} + y_1\mathbf{U} + y_2\mathbf{V} + y_3\mathbf{W} = \mathcal{C} + R\mathbf{Y}$$

where R is a rotation matrix whose columns are $\mathbf{U}$, $\mathbf{V}$, and $\mathbf{W}$, in that order, and where $\mathbf{Y}$ is a column vector whose rows are y_1, y_2, and y_3, in that order. In this coordinate system a point $\mathcal{X}$ on the cylinder must satisfy

$$\begin{aligned} r^2 &= y_1^2 + y_2^2 \\ &= (\mathbf{U} \cdot (\mathcal{X} - \mathcal{C}))^2 + (\mathbf{V} \cdot (\mathcal{X} - \mathcal{C}))^2 \\ &= (\mathcal{X} - \mathcal{C})^{\mathrm{T}}(\mathbf{U}\mathbf{U}^{\mathrm{T}} + \mathbf{V}\mathbf{V}^{\mathrm{T}})(\mathcal{X} - \mathcal{C}) \\ &= (\mathcal{X} - \mathcal{C})^{\mathrm{T}}(I - \mathbf{W}\mathbf{W}^{\mathrm{T}})(\mathcal{X} - \mathcal{C}) \end{aligned}$$

where I is the identity matrix. A *finite cylinder* has center point $\mathcal{C}$, axis direction $\mathbf{W}$, radius r, and is bounded in the axis direction by $|y_3| \leq h/2$ where $h > 0$ is the height of the cylinder. The equation defining cylinder points $\mathcal{X}$ is a quadratic equation and is written in standard form as

$$(\mathcal{X} - \mathcal{C})^{\mathrm{T}} \frac{(I - \mathbf{W}\mathbf{W}^{\mathrm{T}})}{r^2} (\mathcal{X} - \mathcal{C}) = 1 \tag{1}$$

If the cylinder is finite, the bounding condition is $|\mathbf{W} \cdot (\mathcal{X} - \mathcal{C})| \leq h/2$.

2 A Fitting Least–Squares Formulation

In general terms, a 3D point set $\{\mathcal{X}_i\}_{i=1}^{n}$ is assumed to be on a surface whose structure is dependent on a set of m parameters. The surface is represented implicitly by $F(\mathcal{X}; \mathbf{p}) = 0$ for some function F where

$\mathbf{p}$ denotes the parameters listed as an m–tuple. Although we expect $\mathcal{X}_i$ to be exactly on the surface, that is $F(\mathcal{X}_i;\mathbf{p}) = 0$, for whatever reasons the experiment that determined $\mathcal{X}_i$ introduces measurement errors. Consequently, the error is

$$\varepsilon_i = F(\mathcal{X}_i;\mathbf{p})$$

and in practice is rarely zero. In many experiments, the parameters $\mathbf{p}$ themselves are unknown. We wish to select those parameters so that the error terms are as small as possible. Many approaches may be used to minimize the errors. For example, you can try to find $\mathbf{p}$ for which the maximum error term is minimized. For a selected $\mathbf{p}$, the maximum error term is

$$E(\mathbf{p}) = \max_{1\leq i\leq n} |\varepsilon_i(\mathbf{p})|$$

where the error terms are shown to depend on the parameters. We want to find the parameter vector $\hat{\mathbf{p}}$ such that

$$E(\hat{\mathbf{p}}) \leq E(\mathbf{p})$$

for all possible parameter vectors $\mathbf{p}$. If the points actually all lie on the surface with no experiment error (or floating point error when calculating with a computer), the error terms are all zero and $E(\hat{\mathbf{p}}) = 0$. However, the reality of the situation is that errors are inherent in the construction, so we hope that $E(\hat{\mathbf{p}})$ is as small as possible.

The minimization of the maximum error does not easily lend itself to an algorithm to compute $\hat{\mathbf{p}}$. An alternative that does is a *least–squares algorithm.* The idea is to minimize the *total squared error*,

$$E(\mathbf{p}) = \sum_{i=1}^{n} (F(\mathcal{X}_i;\mathbf{p}))^2 \geq 0$$

That is, we seek a parameter vector $\mathbf{p}$ that gives us the *least squared error.* Assuming F is differentiable with respect to the parameter components, the methods of calculus may assist us in locating the global minimum of $E(\mathbf{p})$. Let $\mathbf{p} = (p_1, \ldots, p_m)$. The global minimum occurs either at a point where all the partial derivatives are zero, $\partial E/\partial p_j = 0$ for $1 \leq j \leq m$, or at a boundary point of the set of all parameters. The derivatives are

$$\frac{\partial E(\mathbf{p})}{\partial p_j} = 2\sum_{i=1}^{n} F(\mathcal{X}_i;\mathbf{p})\frac{\partial F(\mathcal{X}_i;\mathbf{p})}{\partial p_j},\ \ 1 \leq j \leq m$$

Setting all these to zero produces a system of m nonlinear equations in m unknowns p_j. In rare cases you might be able to solve the system with algebraic methods. In most cases you will need root–finding methods in multiple dimensions.

A different attack on the problem is to minimize $E(\mathbf{p})$ using an iterative scheme that searches the multidimensional space of parameters. An initial parameter vector is chosen. In many cases the initial vector is based information you know about your specific application. The construction of the initial guess itself might be a complex process, itself the result of a least–squares algorithm. The cylinder fitting algorithm is of this nature, as we will see. A line is chosen that contains the initial parameter vector and a search is for the minimum of E along the line. The parameter vector at which the minimum occurs is used as the starting point for another linear search using a line of different direction than the first. A few general methods are applicable. *Powell's direction set method* is one that searches along an initial set of m lines whose directions are selected to be the coordinate axis directions. After m lines have been searched, the next line is constructed from the previous ones in a manner designed to give you your best chances of quickly getting close to a minimum point. The method does not require derivative evaluation, making it an attractive algorithm when the function to be minimized is not in a form that is amenable to differentiation. The

Conjugate Gradient method is a sophisticated method that also attempts to be smart about choice of line direction. This method does require derivatives and is usually the minimizer of choice in most applications.

A simpler method is the *method of steepest descent*. The idea is to choose the search line in the direction associated with the largest decrease in the function value. This method is regarded to be inferior in many situations, though, because you can spend a lot of time zig–zagging about the parameter space and not get close to a minimum in a reasonable amount of time. Powell's direction set method and the Conjugate Gradient method tend to choose better directions to follow. That said, least–squares fitting of a cylinder to a data set that is approximately a cylinder appears to work reasonably well when using the method of steepest descent. As the input data set becomes more randomized, not even visually appearing to be on a cylinder, you should expect that the method of steepest descent will have problems and a more sophisticated minimizer is called for.

The steepest descent method is summarized here. An initial parameter vector is chosen, call it $\mathbf{p}_0$. The direction of largest decrease of $E(\mathbf{p})$ starting at the initial parameter is the negative of the gradient,

$$\mathbf{D} = -\boldsymbol{\nabla} E(\mathbf{p}_0) = -2\sum_{i=1}^{n} F(\mathcal{X}_i; \mathbf{p}_0)\boldsymbol{\nabla} F(\mathcal{X}_i; \mathbf{p}_0)$$

where $\boldsymbol{\nabla} F$ is the m–tuple of partial derivatives of F with respect to the components of $\mathbf{p}$. The line to search is $\mathbf{p}(t) = \mathbf{p}_0 + t\mathbf{D}$ for $t \geq 0$. Along that line the error function is

$$f(t) = E(\mathbf{p}(t)) = E(\mathbf{p}_0 + t\mathbf{D}) = \sum_{i=1}^{n}(F(\mathcal{X}_i; \mathbf{p}_0 + t\mathbf{D}))^2 \geq 0$$

for $t \geq 0$. The minimum of $f(t)$ occurs at $\bar{t}$ for which $f'(\bar{t}) = 0$ or at $\bar{t} = 0$ in which case $\mathbf{D} = \mathbf{0}$ and you have to decide if $f(\bar{t})$ is the global minimum for E or if you have reached a flat spot (local minimum) and need to further analyze the problem. The derivative is

$$f'(t) = 2\sum_{i=1}^{n} F(\mathcal{X}_i; \mathbf{p}_0 + t\mathbf{D})(\mathbf{D} \cdot \boldsymbol{\nabla} F(\mathcal{X}_i; \mathbf{p}_0 + t\mathbf{D}))$$

Notice that $f'(0) = -|\mathbf{D}|^2 \leq 0$. We know that $f(t) > 0$ by definition. Assuming $f'(0) < 0$, the minimum of $f(t)$ must exist for some $\bar{t}$ at which $f'(\bar{t}) = 0$. The case of $f(t)$ monotonically decreasing and reaching a minimum only in the limit as $t \to \infty$ is ignored at the moment. In practice the types of fitting problems you encounter will not exhibit this pathological behavior. The update on the parameter is $\mathbf{p}_1 = \mathbf{p}_0 + \bar{t}\mathbf{D}$. The process is repeated starting at $\mathbf{p}_1$ along the line whose direction is the negative gradient of E at that point. The iteration is terminated once you have confidence that E will no longer significantly decrease with further iterations.

3 Fitting with a Cylinder

Let $\{\mathcal{X}_i\}_{i=1}^{n}$ be the input point set. An error function for a cylinder fit is

$$E(\mathcal{C}, \mathbf{W}, r) = \sum_{i=1}^{n}\left((\mathcal{X}_i - \mathcal{C})^{\mathrm{T}}\frac{(I - \mathbf{W}\mathbf{W}^{\mathrm{T}})}{r^2}(\mathcal{X}_i - \mathcal{C}) - 1\right)^2$$

The division by r^2 and the fact that $\mathbf{W}$ is unit–length becomes an issue in attempting to minimize E along lines in the space of the parameter vector $(\mathcal{C}, \mathbf{W}, r)$. Instead we will allow non–unit–length axis vectors $\mathbf{V}$ and define $s = 1/(r|\mathbf{V}|)^2$. The error function we use is

$$E(\mathcal{C}, \mathbf{V}, s) = \sum_{i=1}^{n} \left(s(\mathcal{X}_i - \mathcal{C})^{\mathrm{T}}(|\mathbf{V}|^2 I - \mathbf{V}\mathbf{V}^{\mathrm{T}})(\mathcal{X}_i - \mathcal{C}) - 1 \right)^2$$

Observe that $E(\mathcal{C}, \lambda\mathbf{V}, s/\lambda^2) = E(\mathcal{C}, \mathbf{V}, s)$ for any $\lambda > 0$. If we were to update $\mathbf{V}$ by a minimization along a line, and the update is not unit–length, we can normalize the vector, set $\lambda = 1/|\mathbf{V}|$ and modify s to s/λ^2, thereby not changing the error function and allowing us to always use unit–length axis vectors.