PROJECTED MEAN CURVATURE SMOOTHING FOR VECTOR-VALUED IMAGERY

Anthony Yezzi, Jr.

Department of Electrical Engineering
University of Minnesota
Minneapolis, MN 55455
email: ayezzi@ee.umn.edu

ABSTRACT

In this note, we formulate a general modified mean curvature based equation for image smoothing and enhancement. The key idea is to consider the image as a graph in some $\mathbf{R}^n$, and apply a mean curvature type motion to the graph. We will consider some special cases relevant to greyscale and color images.

I. INTRODUCTION

Recently, there have been a number of researchers who have considered the use of nonlinear curvature based equations for various problems in computer vision and image processing. An excellent reference is the volume edited by Bart Romeny [7] to which we refer the interested reader for a large list of references.

In this paper, we consider a twist on the idea of mean curvature smoothing of an image in that we treat the image as the manifold defined by the graph of a function embedded in some Euclidean space. For example, a greyscale 2D image $I : \mathbf{R}^2 \rightarrow \mathbf{R}$ may be regarded as the surface $(x, y, I(x, y)) \subset \mathbf{R}^3$. A 2D color image similarly may be regarded as a surface in $\mathbf{R}^5$. We consider therefore mean curvature motion of these graphs as our underlying model for image smoothing and enhancement. A very attractive feature is that this gives a natural geometric way to treat vector-valued imagery. See [9] and the references therein for other approaches. We should also add that in [8], the authors also consider a similar approach. See [10] for complete details of the results in this paper.

The utility of our methods will be demonstrated on medical and military imagery. The author would like to thank Professor Allen Tannenbaum for a number of very helpful conversations about the material in this paper.

This work was supported in part by grants from the National Science Foundation ECS-9700588 and NSF-LIS, by the Air Force Office of Scientific Research F49620-94-1-0461 and AF/F49620-98-1-0168, by the Army Research Office DAAH04-94-G-0054 and DAAH04-93-G-0332, and by a MURI Grant.

II. 2D GREYSCALE IMAGE SMOOTHING

Before considering the general case below, we believe it is instructive to consider the key case of a two dimensional greyscale image. Throughout this paper, we will freely use the basic facts of differential geometry from [2]. Accordingly, we will consider such an image as the graph of a surface in $\mathbf{R}^3$.

From an inital image $I(x, y)$, construct an initial parameterized surface $S(x, y) = (x, y, I(x, y))$. The unit normal of this surface is given by

$$N(x, y) = \frac{S_x \times S_y}{\|S_x \times S_y\|} = \frac{(-I_x, -I_y, 1)}{\sqrt{1 + I_x^2 + I_y^2}}$$

and the mean curvature by

$$H(x, y) = \frac{I_{xx}(1 + I_y^2) - 2I_x I_y I_{xy} + I_{yy}(1 + I_x^2)}{2(1 + I_x^2 + I_y^2)^{3/2}}.$$

Since S is a graph, it will, under mean curvature motion $S_t = HN$, evolve into a plane without developing singularities [3]. As S evolves in this manner, small scale features of high curvature induced by noise in the image are very quickly removed. However, an undesirable phenonmenon occurs from the point of view of image processing, namely, edges become blurred. These effects will be illustrated in the following example.

Consider a 2D greyscale image which is constant along the y direction but which is black on the left half and white on the right half so that any horizontal cross section along the x direction yields a common step function. Now add small oscillations to simulate noise in the image and "round off" the corners of the step edge so that our function becomes differentiable. Finally, for the sake of illustration, assume that this modeled noise is constant along the y direction so that

0-7803-4876-1/98/$10.00 © 1998 IEEE

mean curvature motion of the surface ($S_t = HN$) causes the cross sectional curve, C, to evolve according to its curvature ($C_t = \kappa N$). This type of motion will have the desirable effect of flattening out the oscillations but will also have the undesireable effect of widening the step edge.

However, suppose we now constrain the evolution of each point of S to be purely vertical by projecting the mean curvature vectors onto the vertical direction (z-axis) so that no sideways motion can occur. Therefore, instead of $S_t = HN$ we consider $S_t = (HN \cdot Z)Z$ which causes C to evolve according to $C_t = (\kappa N \cdot Z)Z$ where Z represents the unit vector in the vertical direction. Clearly, this completely vertical motion will still eliminate the unwanted oscillations but will no longer pull the corners of the step edge further apart. Thus, by vertically projecting regular mean curvature motion of S we obtain an edge preserving, noise removing evolution given by

$$S_t = ((HN) \cdot Z)Z.$$

Notice that the evolving surface under this modified form of mean curvature motion takes the form of $S(x,y,t) = (x,y,f(x,y,t))$ and so $I(x,y,t)$ is easily extracted from $S(x,y,t)$ by setting $I(x,y,t) = f(x,y,t)$. This allows us to dispense with S altogether and simply write down the following edge perserving anisotropic filter for I:

$$I_t = \frac{I_{xx}(1+I_y^2) - 2I_xI_yI_{xy} + I_{yy}(1+I_x^2)}{2(1+I_x^2+I_y^2)^2}.$$

III. SCALING PARAMETER

The filter we have presented can be extended into an entire family of filters by scaling the height of the image. More precisely, if we make the substitution $I \to kI$ into the above equation for some positive constant k we obtain the more general filter

$$I_t = \frac{\triangle I + k^2(I_x^2I_{yy} - 2I_xI_yI_{xy} + I_y^2I_{xx})}{(1+k^2\|\nabla I\|^2)^2}.$$

Scaling I by a large value of k amplifies edges in the image and will yield a filter with very strong edge preserving properties. However, choosing a smaller value of k will yield a faster diffusion. This tradeoff between speed and edge preservation will be illustrated on real images in Sec. 7 but can been seen mathematically by observing the limiting cases shown below.

$$\begin{aligned} k \to 0: &\quad I_t \to \triangle I, \\ k \to \infty: &\quad I_t \to \tfrac{1}{k^2\|\nabla I\|^2}\nabla \cdot \left(\tfrac{\nabla I}{\|\nabla I\|}\right)\|\nabla I\|. \end{aligned}$$

As k becomes very small the aniosotropic diffusion approaches the isotropic heat equation which implements a rather fast diffusion but does a poor job of preserving edges. As k becomes very large we approach a damped geometric heat equation which does a far superior job of preserving edges. Futhermore, the damping term applied to the limiting geometric heat equation helps to prevent the distortion of shapes caused by the pure geometric heat equation. However, this damping term also makes the diffusion much slower.

IV. PRELIMINARY LEMMAS

In this section, we will derive two results which will be useful in analyzing the general formula for projected mean curvature motion.

Lemma 1: If $a_1, \ldots, a_m \in \mathbf{R}^n$ then

$$\pi_N(s) = s^{n-m}\pi_M(s) \quad \text{where} \quad \begin{cases} N = \sum a_i a_i^T \\ M = [a_i \cdot a_j]_{ij} \end{cases}$$

where π_N, π_M denote the characteristic polynomials of M, N.

Lemma 2: If $a_1, \ldots, a_m \in \mathbf{R}^n$ and $k \in \mathbf{R}$ then

$$\det(kI_n + \sum a_i a_i^T) = k^{n-m}\det(kI_m + [a_i \cdot a_j]_{ij})$$

where I_m, I_n denote the $m \times m$ and $n \times n$ identity matrices.

V. THE GENERAL CASE

We will now derive the general formula for the vertically projected mean curvature motion of a graph of arbitrary dimension and codimension. In what follows below we will use the symbol $\mathcal{I}_p$ for the $p \times p$ identity matrix.

In general, the mean curvature vector H of an m-dimensional surface S in $\mathbf{R}^{m+n}$ (co-dimension n) with coordinates $x_1, \ldots, x_m$ is given by

$$H = \mathrm{Tr}[G^{-1}\mathrm{Proj}(\nabla^2 S)], \quad \begin{cases} G = (S_{x_i} \cdot S_{x_j})_{ij} \\ \mathrm{Proj}(\nabla^2 S) = (PS_{x_ix_j})_{ij} \end{cases}$$

where P is the orthogonal projection map which annihilates the component of $S_{x_ix_j}$ in the tangent space of S (span of $S_{x_1}, \ldots, S_{x_m}$). Actually, the H defined here is m times the true mean curvature vector, but by abuse of notation we will continue to refer to H as the mean curvature vector. Since P is linear and $\mathrm{Tr}[G^{-1}\mathrm{Proj}(\nabla^2 S)]$ yields a linear combination of the elements of $\mathrm{Proj}(\nabla^2 S)$ we can pull P outside and write

$$H = P\mathrm{Tr}(G^{-1}\nabla^2 S)$$

Note that the matrices P, G, and $\nabla^2 S$ are all functions of $x = (x_1, \ldots, x_m)$. Consider the case where S is a graph of the form $S(x) = (x, I(x))$, $I : \mathbf{R}^m \to \mathbf{R}^n$. The elements g_{ij} of G, which are just the coefficients of the first fundamental form S, are then $S_{x_i} \cdot S_{x_j} = \delta_{ij} + I_{x_i} \cdot I_{x_j}$, and so $G = \mathcal{I}_m + J(I)^T J(I)$ where $J(I)$ denotes the $n \times m$ Jacobian matrix of I. Also, note that the elements S_{ij} of $\nabla^2 S$ are given by $S_{ij} = (0, I_{x_i x_j})$ where 0 denotes the m-dimensional zero vector and $I_{x_i x_j}$ are the n-dimensional elements of $\nabla^2 I$. Now represent P in block form as

$$P = \begin{bmatrix} A & B \\ C & \hat{P} \end{bmatrix}$$

where A is $m \times m$, B is $m \times n$, C is $n \times m$ and $\hat{P}$ is $n \times n$. Since P annihilates any tangent vector S_{x_i} we have

$$PS_{x_i} = \begin{bmatrix} A & B \\ C & \hat{P} \end{bmatrix} \begin{bmatrix} e_i \\ I_{x_i} \end{bmatrix} = 0$$

where $e_1, \ldots, e_m$ denote the standard orthonormal basis for $\mathbf{R}^m$. From this expression we see that $Ce_i = -\hat{P} I_{x_i} = 0$ for each $i = 1, \ldots, m$ and so

$$C = \begin{bmatrix} -\hat{P} I_{x_1} & \cdots & -\hat{P} I_{x_m} \end{bmatrix} = -\hat{P} J(I),$$

Next, since P preserves any normal vector $N = (u, v)^T$ of S where $u \in \mathbf{R}^m$ and $v \in \mathbf{R}^n$ then,

$$PN = \begin{bmatrix} A & B \\ C & \hat{P} \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} Au + Bv \\ Cu + \hat{P}v \end{bmatrix} = \begin{bmatrix} u \\ v \end{bmatrix} = N,$$

and so $v = Cu + \hat{P}v$. However, note that $N \cdot S_{x_i} = (u, v) \cdot (e_i, I_{x_i}) = u \cdot e_i + v \cdot I_{x_i} = 0$ for each $i = 1, \ldots, m$ since N is orthogonal to each S_{x_i}. Therefore $u = (-v \cdot I_{x_1}, \ldots, -v \cdot I_{x_m})^T = -J^T(I)v$. Substituting for u and C yields $v = [-\hat{P} J(I)][-J^T(I)v] + \hat{P}v = \hat{P}[J(I)J^T(I) + \mathcal{I}_n]v$. From this expression, it is clear that $\hat{P}[J(I)J^T(I) + \mathcal{I}_n]$ must be the identity map, and so $\hat{P} = [J(I)J^T(I) + \mathcal{I}_n]^{-1}$.

Now consider using projected mean curvature motion of S as a way to smooth the m-dimensional n-vector valued image $I : \mathbf{R}^m \to \mathbf{R}^n$. As before, we project the mean curvature vector H of S onto the n-dimensional subspace of $\mathbf{R}^{m+n}$ orthogonal to the m-dimensional domain of I under the inclusion map via the matrix

$$V = \begin{bmatrix} 0 & \\ & \mathcal{I}_n \end{bmatrix}$$

where 0 represents the $m \times m$ zero matrix, and $\mathcal{I}_n$ the $n \times n$ identity matrix. Since the resulting evolution

$$S_t = VH = (VP)\mathrm{Tr}(G^{-1}\nabla^2 S) = \begin{bmatrix} 0 & 0 \\ C & \hat{P} \end{bmatrix} \mathrm{Tr}(G^{-1}\nabla^2 S)$$

leaves the first m components of S unchanged and since the last n components of S evolve according to $\hat{P}\mathrm{Tr}(G^{-1}\nabla^2 I)$ (this is because the first m components of the elements of $\nabla^2 S$ are zero and the last n components form an element of $\nabla^2 I$), we may dispense with S and evolve the image directly via $I_t = \hat{P}\mathrm{Tr}(G^{-1}\nabla^2 I)$. Substituting the values of G and $\hat{P}$ just computed yields

$$I_t = [\mathcal{I}_n + J(I)J^T(I)]^{-1}\mathrm{Tr}\{[\mathcal{I}_m + J^T(I)J(I)]^{-1}\nabla^2 I\}.$$

If we make the substitution $I \to kI$ to account for arbitrary scalings of I we obtain the more general equation

$$I_t = [\mathcal{I}_n + k^2 J(I)J^T(I)]^{-1}\mathrm{Tr}\{[\mathcal{I}_m + k^2 J^T(I)J(I)]^{-1}\nabla^2 I\}.$$

Note, as seen already in the 2D grey-scale case, that as k goes to zero, the diffusion approaches the linear heat equation ($I_t \to \mathrm{Tr}\nabla^2 I$).

If $m > n$, then it may be easier to compute the determinant of the $n \times n$ first fundamental form matrix than it is the $m \times m$ projection matrix in the above equation. We can then use the result of Lemma 2 to avoid the computation of the more difficult determinant. Call Λ the following expression:

$$\mathrm{Adj}[k^{-2}\mathcal{I}_n + J(I)J^T(I)]\mathrm{Tr}\{\mathrm{Adj}[k^{-2}\mathcal{I}_m + J^T(I)J(I)]\nabla^2 I\}$$

Pulling out a factor of k^{-4} and applying Lemma 2 yields

$$I_t = k^{2(n-m)-4}\frac{\Lambda}{\det^2[k^{-2}\mathcal{I}_n + J^T(I)J(I)]},$$

or in case $n > m$ it may be easer to use

$$I_t = k^{2(m-n)-4}\frac{\Lambda}{\det^2[k^{-2}\mathcal{I}_m + J(I)J^T(I)]}.$$

VI. 2D COLOR AND 3D GREY SCALE IMAGERY

In this section, we compute from the general equation, the projected mean curvature diffusions for the important special cases of 2D color and 3D grey scale images.

A 2D color image amounts to a surface in $\mathbf{R}^5$, given by $S(x, y) = (x, y, I(x, y))$. Solving the general equation for $m = 2$ and $n = 3$ with the scaling factor k yields

$$I_t = k^{-2}\mathrm{Adj}(k^{-2}\mathcal{I}_3 + I_x I_x^T + I_y I_y^T)\Omega$$

where Ω is given by the expression

$$\frac{(k^{-2} + I_y \cdot I_y)I_{xx} - 2(I_x \cdot I_y)I_{xy} + (k^{-2} + I_x \cdot I_x)I_{yy}}{[(k^{-2} + I_x \cdot I_x)(k^{-2} + I_y \cdot I_y) - (I_x \cdot I_y)^2]^2}$$

In fact, by merely replacing $\mathcal{I}_3$ in the above equation with $\mathcal{I}_n$, we obtain the projected mean curvature diffusion equation for an n-vector valued 2D image.

A 3D grey scale image amounts to a 3 dimensional hypersurface in $\mathbf{R}^4$ given by $S(x,y,z) = (x,y,z,I(x,y,z))$. Solving the general equation for $m = 3$ and $n = 1$ with the scaling factor k yields

$$I_t = \frac{\Omega_1 + \Omega_2}{(k^{-2} + \|\nabla I\|^2)^2}$$

where

$$\Omega_1 := k^{-2}\nabla^2 I + (I_y^2 + I_z^2)I_{xx} + (I_x^2 + I_z^2)I_{yy}$$

and

$$\Omega_2 := (I_x^2 + I_y^2)I_{zz} - 2(I_x I_y I_{xy} + I_x I_z I_{xz} + I_y I_z I_{yz}).$$

VII. NUMERICAL EXPERIMENTS

Figures 1 and 2 illustrate the use of projected mean curvature smoothing on two-dimensional greyscale and color data. The color image Figure 2 is reproduced here in greyscale. The original images may be found on my website at www.ece.umn.edu/users/ayezzi. A time step of 0.1 was consistently applied in each example so that the number of iterations performed in different cases could be compared in a meaningful manner. In both figures, the uppermost image displays the original unfiltered data and the lower four images display the results of our filter under two different scaling factors. In Fig. 1, a greyscale MRI image of the spinal cord lumbar region, the two images of the left exhibit the effect of 50 and 100 iterations using the scaling factor $k = 0.2$. Alongside these images, on the righthand side, are the results of 250 and 500 iterations using a larger scaling factor $k = 1.0$. The larger scaling factor has done a superior job of preserving edges; the images on the lefthand side are clearly more blurred. However, this improvement in performance has come at the cost of a slower diffusion as seen by comparing the number of iterations required in these two cases. Fig. 2, a rather grainy aircraft image, reveals this same type of behavior on a color image. The bottom two images show the results of 10 and 20 iterations using a scale factor $k = 0.05$ while the preceding two images show the results of 4 and 8 iterations using a scale factor $k = 0.01$. Although a larger number of iterations were required in the bottom two images to attain an equivalent level of smoothing, the improvement in edge preservation can be seen very clearly by looking at the letters "YF-23" on the tail of the airplane in the two sets of images.

VIII. REFERENCES

[1] L. Alvarez, F. Guichard, P. L. Lions, and J. M. Morel, "Axioms and fundamental equations of image processing," *Archive of Rational Mechanics and Analysis* **123**, 1993.

[2] M. P. Do Carmo, *Differential Geometry of Curves and Surfaces*, Prentice-Hall, Inc., New Jersey, 1976.

[3] K. Ecker and G. Huisken, "Mean curvature flow of entire graphs," *Annals of Math.* **130**, pp. 453-471, 1989.

[4] A. El-Fallah and G. Ford, "The evolution of mean curvature in image filtering," *Proceedings of ICIP*, pages 298-303, 1994.

[5] P. Perona and J. Malik, "Scale-space and edge detection using anisotropic diffusion," *IEEE Trans. Pattern Anal. Machine Intell.* **12**, pp. 629-639, 1990.

[6] G. Sapiro and D. Ringbach, "Anisotropic diffusion of multivalued images with applications to color filtering," *IEEE Trans. Image Processing* **5**, pp. 1582-1585, 1996.

[7] *Geometry-Driven Diffusion in Computer Vision*, edited by Bart ter har Romeny, Kluwer, 1994.

[8] N. Sochen, R. Kimmel, and R. Malladi, "From high energy physics to low level vision," LBNL Technical Report #39243, Berkeley, CA, August 1996. This paper appears in this issue of *IEEE Transactions of Image Processing.*

[9] R. Whitaker and G. Gerig, "Vector-valued diffusion," in *Geometry-Driven Diffusion in Computer Vision* (edited by Bart ter har Romeny), pages 93-133, 1994.

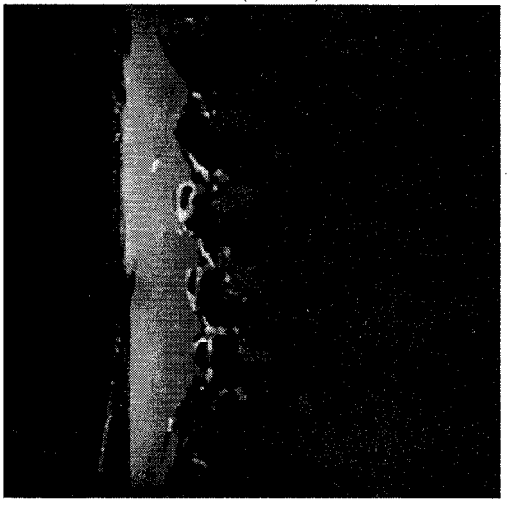
[10] A. Yezzi, "Modified mean curvature motion for smoothing and enhancement," to appear in *IEEE Trans. Image Processing*, 1998.

Figure 1: Projected mean curvature smoothing on a grey scale MRI image

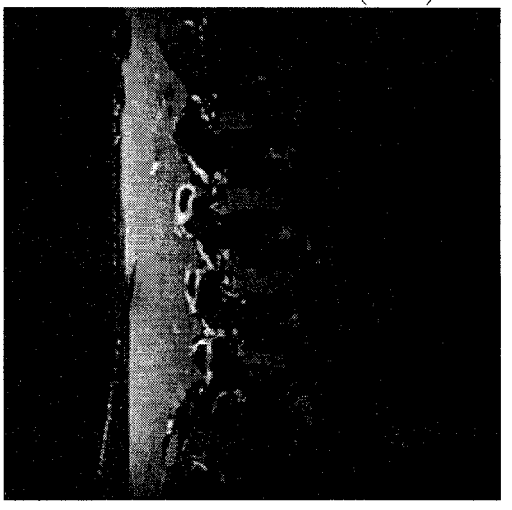
Original Image

50 iterations (k=.2)

250 iterations (k=1)



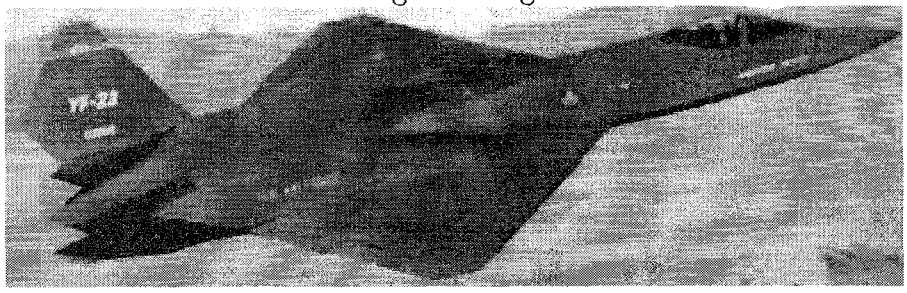
100 iterations (k=.2)



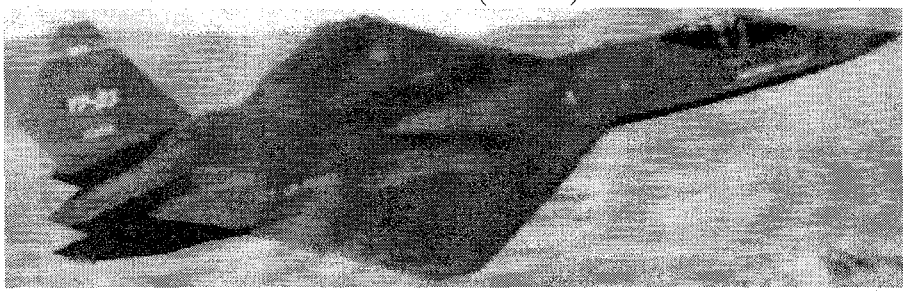
500 iterations (k=1)

Figure 2: Projected mean curvature smoothing on a color aircraft image

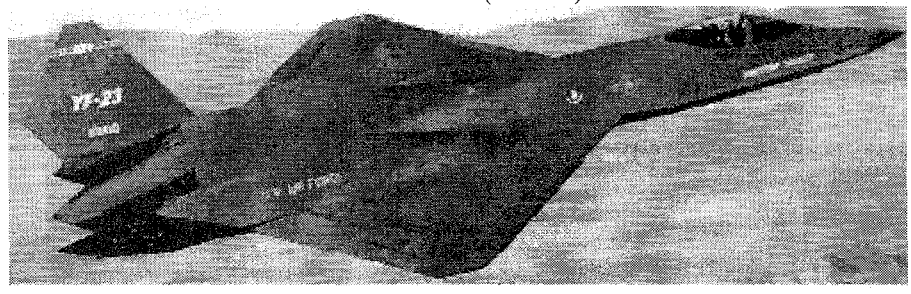
Original Image



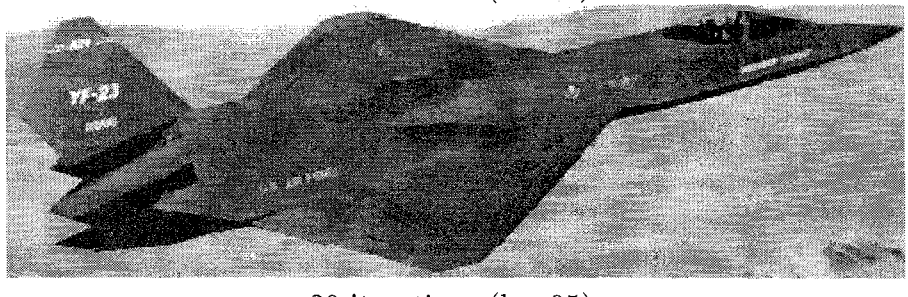
4 iterations (k=.01)



8 iterations (k=.01)



10 iterations (k=.05)



20 iterations (k=.05)