FROM 2-D TO 3-D: ALGORITHMS TO RECREATE A REAL-WORLD SCENE FROM FLAT PHOTOGRAPHS*

JESSICA CONWAY†

Abstract. The goal of this paper is to provide a simple and efficient algorithm for the The recovery of a three-dimensional scene from two-dimensional images of the same object or scene. To this end, we present an outline of an approach to extracting depth information from two-dimensional images, and then a direct featureless method to recover the 15 parameters of the exact projective coordinate transformation between two images. When we say exact, we are operating under the assumptions of static scene and no parallax, although we suspect and hope in the future to show that our methods are robust under deviations from these assumptions. Future work includes numerical experiments, and comparisons with real data.

Key words. Image processing, least-squares estimation

AMS subject classifications.

1. Introduction and examples. It would often be useful to be able to generate a three dimensional image from a series of two dimensional images of the same object. One could extract more detailed information from the scene being photographed, for example. To do this, we would need some approach to uncover depth information and then find the coordinate transformations between images.

Our goal is to accomplish this featurelessly, that is, without some foreknowledge of objects in the image to create automatic depth or coordinate transformations. To this end, we split our task into two steps: generating the three-dimensional from two-dimensional scenes, and then relating the three-dimensional images to each other through coordinate transformations.

The algorithm to make the three-dimensional images have yet to be properly developed. We present here a first approach to the problem, together with a more concrete but unfortunately featured method for extracting depth information.

The featureless motion estimation to find the coordinate transformation is more specifically geared towards finding the 15-parameter projective coordinate transformation. Our technique utilizes the projective flow transformation and the optical flow equations, taking as input two frames and providing as output the 15-parameters of the exact model of motion. This approach is largely a three-dimensional extension of Mann and Picard's work on featureless estimation of parameters,[1].

The approach presented here operates under the assumptions of static scene (the image brightness stays constant between shots) and no parallax. We hope, with time, to show the methods are robust to within some deviations of these assumptions, as that would render them more practical.

2. Extracting Depth Information from 2-Dimensional Images:. Transforming a scene from a series of 2-dimensional images to a full-blown 3-dimensional scene is tricky. How does one extract depth information out of a photograph consistently without some precious knowledge of the scene?

Our approach is to attempt to find a transformation from the three-dimensional coordinate $\vec{x} = (x, y, z)^T$ to the two-dimensional coordinate $\vec{x}' = (x', y', z_0)^T$ where z_0 is a fixed plane, and then invert the transformation.

*Undergraduate Honors project work, in progress, under supervision by J. Cooperstock and N. Nigam.

†Department of Mathematics and Statistics, McGill U.(delareuth@hotmail.com).

We assume that the original point $\vec{x}$ (coordinates relative to some fixed origin) is related to the image point (x', y', z_0) (in the same reference frame) via a projection:

$$\begin{pmatrix} x' \\ y' \\ z_0 \end{pmatrix} = \frac{1}{(c_1, c_2, 0)\vec{x} + 1} \left\{ \begin{pmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ 0 & 0 & k_0/z \end{pmatrix} \vec{x} + \begin{pmatrix} b_1 \\ b_2 \\ 0 \end{pmatrix} \right\},$$

where $k_0 = ((c_1, c_2, 0)\vec{x} + 1)z_0$. We seek to estimate the parameters $a_{11}, a_{12}, ..., c_2$. Notice that both **b** and **c** here have a third component of zero. This may seem strange, but it is automatically corrected for in 2-dimensional images in the x and y coordinate transformation. So we have

$$\begin{pmatrix} x' \\ y' \end{pmatrix} = \frac{1}{(c_1, c_2, 0)\vec{x} + 1} \left\{ \begin{pmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} b_1 + a_{13}z \\ b_2 + a_{23}z \end{pmatrix} \right\}$$

Since there are 12 unknowns, we need 4 points from a frame whose z-coordinate is z_0 to make any progress at all. One attempt to solve for the parameters is to take points located in a pictures- say $\begin{pmatrix} x' \\ y' \\ z_1 \end{pmatrix}$ and $\begin{pmatrix} x'' \\ y'' \\ z_2 \end{pmatrix}$, apply the transformation, and compare the results. We thus get equations of the form

$$\begin{pmatrix} x'' \\ y'' \\ z_2 \end{pmatrix} - \begin{pmatrix} x' \\ y' \\ z_1 \end{pmatrix} = \frac{1}{(c_1 x'' + c_2 y'') + 1} \left\{ \begin{pmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \end{pmatrix} \begin{pmatrix} x'' \\ y'' \end{pmatrix} + \begin{pmatrix} b_1 + a_{13}z_2 \\ b_2 + a_{23}z_2 \end{pmatrix} \right\}$$
$$- \frac{1}{(c_1 x' + c_2 y') + 1} \left\{ \begin{pmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \end{pmatrix} \begin{pmatrix} x' \\ y' \end{pmatrix} + \begin{pmatrix} b_1 + a_{13}z_1 \\ b_2 + a_{23}z_1 \end{pmatrix} \right\}$$

which we can solve for the parameters so that $(x', y', z_1)^T$ and $(x'', y'', z_2)^T$) both map to some (x, y, z_0). Another, more featured approach to finding the transform before inverting it is to find a box of known depth configuration in the three-dimensional image (say, the side of a building). An analysis of how this box shifts into a parallelopiped may yield an initial transform $T\vec{x} \to \vec{x}'$, where $\vec{x}' = (x', y', z_0)^T$, and a simple inversion of T would give the transformation we require for the whole picture.

3. Relating 3-dimensional images to each other.

3.1. Coordinate transformation. The desired coordinate transformation maps the image coordinates $\vec{x} = (x, y, z)^T$ to a new set of coordinates $\vec{x}' = (x', y', z')^T$. The transformation parameters encompass camera rotation, zoom, pan, tilt and translation. We use the exact projective coordinate transformation

$$\vec{x}' = \frac{A\vec{x} + \mathbf{b}}{\mathbf{c}^T\vec{x} + \delta},$$

where A describes a rotation, **b** a translation, and $\mathbf{c}, \delta$ describe "stretching". As in most engineering applications $\delta \neq 0$, we divide through by δ, resulting in the transformation

$$\vec{x}' = \frac{A\vec{x} + \mathbf{b}}{\mathbf{c}^T\vec{x} + 1}.$$

Our aim is to estimate the motion parameters. To do so, we define for $\vec{x}$ in frame t going to $\vec{x}' = \vec{x} + \triangle\vec{x}$ in a frame $t + \triangle t$ the "image brightness", $E(\vec{x}, t)$. From our static scene assumption,

$$E(\vec{x}, t) = E(\vec{x} + \triangle\vec{x}, t + \triangle t).$$

We expand the right-hand side in a Taylor series about $(\vec{x}, t)$, and drop the higher order terms to get

$$E(\vec{x}, t) = E(\vec{x}, t) + \triangle\vec{x}E_x + \triangle t\boldsymbol{E}_t \Rightarrow \frac{\triangle\vec{x}}{\triangle t}E_x + E_t = 0.$$

Here we denote $\nabla E(\vec{x}, t) = (\frac{\partial E}{x}, \frac{\partial E}{\partial y}, \frac{\partial E}{\partial z}) = \boldsymbol{E}_x$. The quantity $\frac{\triangle\vec{x}}{\triangle t}$ is called the *flow velocity* $\mathbf{u}_f$. We want the flow velocity to match the model velocity $\mathbf{u}_m = \vec{x}' - \vec{x} = \frac{A\vec{x}+b}{c^T\vec{x}+1} - \vec{x}$. We therefore want to minimize

$$\epsilon = \sum_{\vec{x}\,\text{in frame}} \|\mathbf{u}_m - \mathbf{u}_f\|^2.$$

To this end, let

$$A := \begin{pmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{pmatrix}, \quad \mathbf{b} = \begin{pmatrix} b_1 \\ b_2 \\ b_3 \end{pmatrix}, \quad \mathbf{c} = \begin{pmatrix} c_1 \\ c_2 \\ c_3 \end{pmatrix}, \quad \boldsymbol{E}_x = \begin{pmatrix} E_x \\ E_y \\ E_z \end{pmatrix}, etc$$

and the cost function be given by

$$\mathcal{E}_w = \sum_{\vec{x}\,\text{in frame}} \{(A\vec{x} + \mathbf{b} - (\mathbf{c}\cdot\vec{x} + 1)\vec{x}) \cdot \boldsymbol{E}_x + (\mathbf{c}\cdot\vec{x} + 1)E_t\}^2 \tag{3.1}$$

Note that $\mathcal{E}_w$ is a *weighted* error, whose minimum is computed via a least squares best fit. (The weight is $(\mathbf{c}^T\vec{x} + 1)\boldsymbol{E}_x$). To complete this process, we require at least 5 points. Note that

$$(A\vec{x}) \cdot \boldsymbol{E}_x = \begin{pmatrix} a_{11}xE_x & a_{12}yE_x & a_{13}zE_x \\ a_{21}xE_y & a_{22}yE_y & a_{23}zE_y \\ a_{31}xE_z & a_{32}yE_z & a_{33}zE_z \end{pmatrix}$$

$$\mathbf{b} \cdot \boldsymbol{E}_x = b_1E_x + b_2E_y + b_3E_z$$

$$\begin{aligned}(\mathbf{c}\cdot\vec{x} + 1)\vec{x} \cdot \boldsymbol{E}_x = {} & c_1(x^2E_x + xyE_y + xzE_z) \\ & + c_2(xyE_x + y^2E_y + yzE_z) \\ & + c_3(xzE_x + yzE_y + z^2E_z) - \vec{x} \cdot \boldsymbol{E}_x\end{aligned}$$

and

$$(c\cdot\vec{x} + 1)E_t = c_1xE_t + c_2yE_t + c_3zE_t.$$

To compute the minimum, we differentiate $\mathcal{E}_w$ w.r.t. the free parameters $A, \mathbf{b}, \mathbf{c}$ and set the result to zero. This results in the linear system

$$(\sum \phi\phi^T)\,[a_{11}\,a_{12}\,a_{13}\,b_1\,a_{21}\,a_{22}\,a_{23}\,b_2,\;a_{31}\,a_{32}\,a_{33}\,b_3]^T = \sum(\vec{x}^T\boldsymbol{E}_x - E_t)\phi,$$

where

$$\phi = \left[E_x(x,y,z,1), E_y(x,y,z,1), E_z(x,y,z,1), (E_t - \vec{x}^T\boldsymbol{E}_x)(x,y,z)\right]^T$$

3.2. The approximate model. To obtain the approximate model for the transformation parameters, we expand $\vec{x}' = \frac{A\vec{x}+b}{c^T\vec{x}+1}$ in a Taylor series about $\vec{x}$. Approximately constraining parameters engendered by the Taylor series to 3rd or 4th order, a variety of 15-parameter models can be obtained. Call these parameters $\mathbf{q}$. These parameters are used to create a model velocity $\mathbf{u}_m$, which is then inserted into the flow criteria. This insertion yields a simple set of fifteen nonlinear equations:

$$(\sum \phi^T\phi^)\mathbf{q} = -\sum E_t\phi$$

where ϕ is a vector function of $x, y, z, \boldsymbol{E}, E_t$ depending on the approximate model chosen.

The goodness of fit of the approximate model to the exact model can be evaluated by first relating the parameters of the approximate model to the exact model, and then finding the MSE between the reference image and the comparison image after application of the coordinate transformation to the exact model. This is done because our true goal is to assess how the exact model describes the coordinate transformation. A method of finding, given the parameters of the approximate model, those of the exact model, is given next.

3.3. An '8-point method' for relating an approximate model to an exact model. As the Taylor series method described above yields 15 nonlinear equations, the parameters of the exact model cannot be easily related to those of the approximate model directly. Instead, we put forward as featureless "8-point" method based on the 4-point method of [1].

1. Take 8 ordered points (e.g., the 8 corners bounding the entire 3-dimensional image, or the 8 corners of a box bounding a region under analysis). For argument's sake, suppose the box is the unit cube

$$\mathbf{s} = [s_1, s_2, s_3, s_4, s_5, s_6, s_7, s_8] = [(0,0,0)^T, (1,0,0)^T, ..., (0,1,1)^T]$$

2. Apply to these points the coordinate transformation using the Taylor series for the approximate model to each of these points, obtaining $r_m = u_m(\mathbf{s})$.
3. Treat the correspondences between r and $\mathbf{s}$ as features, which results in 8 easier-to-solve linear equations

$$\begin{pmatrix} x'_k \\ y'_k \\ z'_k \end{pmatrix} = M_1\vec{m}$$

with

$$M_1 = \begin{pmatrix} x_k & y_k & z_k & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -x_kx'_k & -y_kx'_k & -z_kx'_k \\ 0 & 0 & 0 & 0 & x_k & y_k & z_k & 1 & 0 & 0 & 0 & 0 & -x_ky'_k & -y_ky'_k & -z_ky'_k \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & x_k & y_k & z_k & 1 & -x_kz'_k & -y_kz'_k & -z_kz'_k \end{pmatrix}$$

and

$$\vec{m} = \begin{pmatrix} a_{x'x} \\ a_{x'y} \\ a_{x'z} \\ b_x \\ a_{y'x} \\ a_{y'y} \\ a_{y'z} \\ b_y \\ a_{z'x} \\ a_{z'y} \\ a_{z'z} \\ b_z \\ c_x \\ c_y \\ c_z \end{pmatrix}.$$

This results in fifteen exact parameters, which we denote by $\mathbf{p}$.

3.4. Multiscale iterative implementation. In practice, multi-scale algorithms are preferable, allowing the recovery of the transformation between images to a desired accuracy. This process of iterative refinement is rapid. Here, we would like to retrieve the "exact" motion model from approximate ones. Given two images (a reference image and a transformed one), we begin with a coarse-level description of an approximate transformation between the two. Once we have this approximate model, we retrieve a (coarse-level) exact model, and check this for "goodness" of fit. If required, we iterative this procedure.

In an attempt to iteratively converge to the exact motion model, one must first note that the projective model forms a group, and that thus we can use the algebraic law of composition.

The following algorithm utilizes both this idea and the 8-point method to yield the exact coordinate transform between two images. (Again, this is a 3-d version of Mann's algorithm).

1. Initialize: set g the reference image, $\mathbf{h}$ the shifted image, $\mathbf{h_o} = \mathbf{q}$, $\mathbf{p_o}, \mathbf{T_o}$ the identity.
2. Somehow estimate the 15 or more terms of the approximate model between two image frames g and h_{i-1} which results in the approximate model parameters q_i.
3. Use the 8-point method to relate the approximate parameters q_i to the exact parameters, denotes T_i.
4. Resample : set $p_i = T_i \odot T_{i-1}, \mathbf{h_i} = \mathbf{p_i} \odot \mathbf{h}$

This process is repeated until the error between h_i and g falls below some tolerance, or until a maximum number of iterations is achieved. One should note that this is based on the featureless 8-point method, and is hence also featureless.

REFERENCES

[1] S. Mann and W. Picard, *Video orbits of the projective group: a simple approach to featureless estimation of parameters.*

[2] S. Maybank, *Theory of reconstruction from image motion,* Springer Verlag series in Information Science, Springer-Verlag 1993.

[3] B. Girod, D. Kuo*Direct estimation of displacement histograms*, OSA meeting on Image understanding and machine vision, June 1989.

[4] P. Eisert, E. Steinbach, B. Girod*Automatic reconstruction of stationary 3-D objects from multiple uncalibrated camera views*, IEEE. TRANS. CIRCUITS & SYS. for VIDEO TECH., V. 10 no. 2 March 2000.