VR Models from Epipolar Images: An Approach to Minimize Errors in Synthesized Images

Mikio Shinya, Takafumi Saito, Takeaki Mori and Noriyoshi Osumi

NTT Human Interface Laboratories

Abstract. A new paradigm, the minimization of errors in synthesized images, is introduced to organically combine Computer Vision and Computer Graphics for Virtual Reality applications. Based on it, a powerful algorithm, called *the strip DP algorithm*, is proposed for epipolar image analysis. The algorithm reconstructs VR models from epipolar images so that the error in the images synthesized from the extracted model is minimized while geometrical consistency is maintained. The dynamic programming technique, adopted as the optimization engine, yields complete optimization at reasonable computation cost in a robust way. The strip DP algorithm is a multi-pass solution to occlusion problems, and, in each pass, it extracts connecting feature lines that are not occluded by undetermined feature lines. Experiments demonstrate its feasibility.

1 Introduction

The recent strong progress in Computer Graphics technologies has brought Computer Graphics (CG) and Virtual Reality (VR) applications into daily life. This results in increasing demands to create complex CG/VR models for fancy image generation. Unfortunately, geometric modeling technologies have advanced only in limited areas in the last decade, and most modeling still requires skilled creators performing tedious, precise, and time-consuming work. Quite naturally, people started to consider VR model generation from real images in order to by-pass the geometric modeling process and to provide easier VR model construction tools [6,5,10,8,4,1].

This paper presents a new paradigm in CG applications of CV: *minimization of errors in synthesized images*. The conventional application scheme is a simple concatenation of a CV system and a CG system, where the CV part extracts surface geometries and textures (VR models), and the CG part renders images from them. Although the CV part, namely Shape-from-X, may involve optimization in itself, the total system is an open-loop in terms of image synthesis. The error minimization paradigm can introduce a closed loop into the over-all system that tries to optimize the extraction of VR models for image synthesis. This feedback is essential to the performance and robustness of the system unifying CV and CG.

In CG applications, good models are those from which good images can be synthesized. What are good images? A simple idea is that, if the same

images as the input images from which a model is derived are synthesized, they are good images. This criterion is reliable when we have a massive set of input images taken from many different view points because we can expect satisfactory image synthesis under a variety of viewing conditions similar to those of the input data.

In this paper, we particularly focus on the CG application of epipolar image analysis, and realize an error minimization paradigm. It is shown that the optimization can be achieved efficiently and robustly by the dynamic programming technique (DP), which calculates the optimum solution at $O(n^2)$ cost for n-variable optimization. The process is implemented and examined with several data sets. Preliminary results show the potential of our approach.

2 Epipolar image analysis

Epipolar analysis [3] is one of the 'shape from motion' techniques, and reconstructs 3D shape from dense image sequences. An epipolar image is a slice of an image sequence that satisfies an epipolar constraint. When the camera movement is parallel to the scan-line direction (hence perpendicular to the viewing direction), epipolar image $I_0(x,t)$ is a simple time slice of input image sequence $i(x,y;t)$, described by

$$I_0(x,t) = i(x,y_0;t),$$

where t is time, and the x-axis is in the scan-line direction. (For more general cases, see [2].) In this simple case, the trace of the image points projected from a 3D point is a straight line and its slope is proportional to the depth from the camera (Z), described as

$$dx/dt = \alpha Z,$$

where α is a constant determined by the camera parameters and the camera velocity. Therefore, extracting the traces of feature points provides their 3D depth values.

For CG applications, however, topological connections among these points are necessary to execute hidden surface removal and texture mapping in the rendering process. This reconstruction is a hard problem to solve. If the hidden parts of all the traced lines are correctly extracted, logically possible connections can be searched through symbolic operations [9]. However, it is generally difficult to detect the end/start points of edges near intersections, and small mistakes in extraction can lead to fatal errors due to the logical reasoning features.

Here, we introduce the error minimization paradigm to robustly extract geometry and textures so that the 'best' images can be synthesised in terms of errors. We do not assume that the hidden parts of feature lines are known. We also allow incorrect feature lines to be extracted. Both the selection of feature

lines and the decision of occlusions are made in the optimization process. This yields robust computation by suppressing the influence of wrong results from the pre-processing stage. In short, the optimization problem addressed here can be stated in the following way: *given n extracted feature lines, select* $m \leq n$ *lines and determine their connections so that the difference between the synthesized images and input images is minimized.*

3 Error estimation and optimization for occlusion-less scenes

This section presents analyses on occlusion-less cases. Occlusions are then discussed in the next sections.

3.1 Errors in synthesized images

First, let us define errors when connecting two feature lines $a_1(t)$ and $a_2(t)$. By using a norm of pixel colors $||.||$, the error h can be defined by the difference between the input epipolar image $I_0(x,t)$ and the synthesized image $I_{syn}(x,t)$, as:

$$h(a_1, a2) = \int_t \int_{a_1(t)}^{a_2(t)} ||I_0(x,t) - I_{syn}(x,t)|| dx dt. \tag{1}$$

The norm can be, for example, the square sum of rgb-values,

$$||I|| = I^2 = r^2 + g^2 + b^2 \tag{2}$$

for $I = (r, g, b)$. The synthesized image I_{syn}, and hence the error function as well, depends on the rendering algorithm used. In most VR applications, simple texture mapping with the diffusive reflection model is used. In this case, I_{syn} is represented by using texture $f(s)$ defined on $0 \leq s \leq 1$, as

$$I_{syn}(x,t) = f((x - a_1(t))/(a_2(t) - a_1(t))). \tag{3}$$

Next, let us determine the texture $f(s)$ that minimizes the error h. By using Eqs. 2 and 3 and setting the deviation $\partial h/\partial f = 0$, we have the optimum texture:

$$f(s) = \int_t (a_2 - a_1) I_0((a_2 - a_1)s + a_1, t) dt / \int_t (a_2 - a_1) dt. \tag{4}$$

This equation means that the texture should be the weighted average of the input image along the flow. The error function $h(a_1, a_2)$ is then calculated as the variance of the image, represented by

$$h(a_1, a_2) = \int_0^1 [\int (a_2 - a_1) I_0^2 dt - (\int (a_2 - a_1) I_0 dt)^2] ds \tag{5}$$

3.2 Optimization with Dynamic Programming

The most important point in the previous discussion is that the error between two lines depends only on these two lines in the occlusion-less case. This feature allows us to apply the dynamic programming technique (DP), which is a powerful optimization tool, widely used in many areas, e.g., [7].

Assume that n-lines $\{l_1, ..., l_n\} = L$ are extracted as candidate feature lines. The task is to select m-lines $\{\lambda_1, ..., \lambda_m\} = \Lambda$ from L so as to minimize the total error $H(\Lambda)$. The total error H is the sum of the errors between adjacent selected lines, represented by

$$H(\lambda_1, ..., \lambda_m) = \sum_{i=0}^{m} h(\lambda_i, \lambda_{i+1}), \tag{6}$$

where $\lambda_0 = l_0$ and $\lambda_{m+1} = l_e$ are the left and right boundaries of the epipolar image. Although a naive optimization of $H(\Lambda)$ involves *n-dimensional* search, it is possible to achieve it as an $O(n^2)$ process by DP (see [7]).

4 Occlusions

Although the analysis of the occlusion-less case is straightforward, occlusion poses serious problems to DP optimization. Unlike the occlusion-less case, candidate lines may intersect each other when one occludes another. This suggests that error h associated with two lines (l_1-l_2) also depends on all crossing lines,

$$h(l_1, l_2) = h(l_1, l_2; l_3, l_4, ...).$$

Thus, DP cannot be applied in this form.

Fortunately, we found a multi-pass solution using DP that avoids NP complexity. This solution is based on the fact that the visible area of an occluding span is never affected by the occluded spans. This suggests the possibility of successively appling DP and determining the optimal connections from near to far in terms of depth.

This section presents the basic idea of dealing with occlusions using simple examples consisting of a two-layered scene (foreground and background). Although we have theoretical form for general situations, they are omitted here due to the tight space requirement.

4.1 Without self-occlusion

There are two types of occlusion to be considered: self-occlusion, in which a surface is hidden by itself or its adjacent faces, and 'circumstantial' occlusion, in which a surface is occluded by other surfaces. We first discuss the circumstantial case, and then extend the analysis to general cases. Our strategy is a multi-pass approach. In each pass, we extract spans that are occluded only by the already extracted spans. When there is no self-occlusion, this extraction can be performed in the following way.

1 Select candidate lines that cannot be occluded. A line is selected if its slope is smaller than those of all lines intersecting it. In the example in Figure 1-a, candidate lines l_4 and l_5 are selected. Let us call these lines *primal candidate lines*, or PCLs in short. Note that PCLs do not intersect each other.

2 For each pair of PCLs (p_i, p_j), evaluate the error $h_{close}(p_i, p_j)$ according to Eq. 5. The value h_{close} represents the error imposed when p_i and p_j are connected.

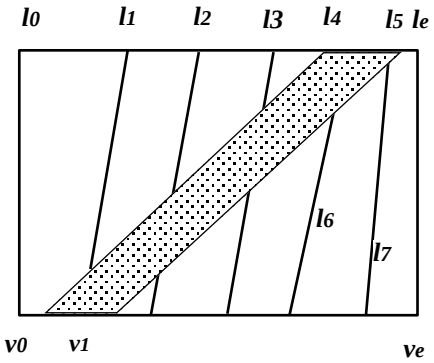
3 For each pair of PCLs, evaluate $h_{open}(p_i, p_j)$ that represents the minimum error imposed on the region bounded by p_i and p_j when they are assumed to be disconnected. For example, $h_{open}(l_0, l_4)$ can be given by applying DP described in Section 3.2 to lines l_0, l_1, l_2, l_3 and l_4 in the region left to l_4. More generally, h_{open} can be calculated by recursively applying Steps 1-4 to the region as in a divide-and-conquer fashion.

4 Apply DP to get

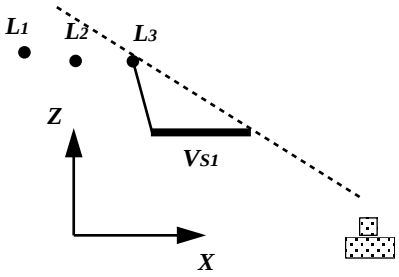
$$\min \sum h_i(\lambda_i, \lambda_{i+1}),$$

where h_i is either h_{open} or h_{close}. We set the constraint that h_i and h_{i+1} can not be h_{open} at the same time for any i. This is because, if both h_i and h_{i+1} are open, the candidate line λ_{i+1} does not really exist, in which case $h_{open}(\lambda_i, \lambda_{i+2})$ should be taken instead. Let us call this connected span a *visible span (VS)*.

5 Remove connecting PCLs and denied PCLs. Also trim VS regions from the image area to process. Repeat the process from Step 1.



(a) Visible spans and candidate lines. Spans *v0* and *ve* are the boundaries, treated as a special kind of visible spans.



(b) The corresponding X−Z space. Span *VS1−L3* occludes *L1 and L2* when viewing from the right side.

Figure 1: Occlusion in epipolar images.

4.2 General occlusion

Self-occlusion makes the situation more complicated. Let us consider the situation in Figure 1, where visible span v_1 (l_4-l_5) is already extracted and all remaining lines p_i are primal candidate lines. When, for example, l_3 is connected to v_1, primal candidate lines left to l_3 are all occluded by span l_3-v_1 when viewing from the right side of v_1 (see Figure 1-b). Similarly, when l_6 connects to v_1, span v_1-l_6 occludes all PCLs right to l_6 when viewing from the left-side of v_1.

Therefore, the error $h_v(v_0, v_1)$ associated with the region between v_0 and v_1 depends only on the connectivities of the VSs and candidate lines between v_0 and v_1, represented by

$$h_v(v_0, v_1) = h_v(v_0, v_1; \mu_1, \{\lambda_i\}),$$
$$\lambda_i \in \{l_1, l_2, l_3\},$$

where μ_1 is assumed to connect to v_1. Let H_v denote the minimum value of h_v, defined as

$$H_v(v_0, v_1; \mu_1) = \min_{\lambda_i} h_v(v_1, v_2; \mu_1, \lambda_1, .., \lambda_m). \tag{7}$$

This minimization can be performed by DP described in Section 3.2, since all l_i are primal in this example. Let us define a function H as the summation of H_v:

$$H(V_0, .., V_m; \mu_0, .., \mu_m) = \sum_i H_v(v_i, v_{i+1}) \tag{8}$$

H represents the minimum error imposed when we choose m connections V_i-μ_i. Since Eq. 8 is in the form of DP with respect to connection (V_i, μ_i), it is possible to minimize the total error H by DP. The connections (V_i, μ_i) that provide the minimum error can be regarded as new visible spans. Therefore, by incorporating this optimization with the multi-pass approach previously discussed, it is possible to extract visible spans in general occluding environments. Since the process extracts and removes visible spans in each pass, we call it *the strip DP algorithm*.

5 Experiment and Discussion

Some preliminary experiments were made to confirm the feasibility of the strip DP algorithm. Figure 2-a shows a test data sample that was photographed by an uncalibrated motion-controlled camera. The strip DP method first extracts un-occluded visible spans, and other spans were extracted in the front-to-back order in each pass. There initially were 85 candidate feature lines(2-b), and 59 visible spans consisting of 65 feature lines were extracted in total, discarding 20 feature lines were discarded. The required computation time was 36.9 seconds in total on an SGI workstation with an R4400 at

250MHz. The image synthesized from the extracted spans is almost visually identical to the input image, and the error-power ratio was 0.018 (Figure 2-d and -e).

6 Conclusion

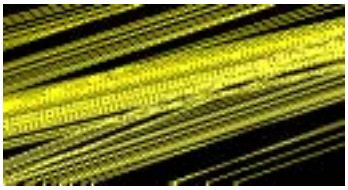
A new algorithm, called *the strip DP algorithm*, was proposed for epipolar image analysis. The algorithm reconstructs VR models from epipolar images so that the error in the images synthesized from the extracted model becomes minimal while geometric consistency is maintained. We adopted the dynamic programming technique as the optimization engine, because it can perform complete optimization at reasonable computation cost. The strip DP algorithm is a multi-pass solution to occlusion problems, and, in each pass, it extracts connecting feature lines that are not occluded by undetermined feature lines. Preliminary experiments demonstrated its feasibility.

References

1. Azarbayejani, A., Galyean, T., Horowitz, B., and Pentland, A "Recursive estimation for CAD model recovery", *proc. 2nd CAD-Based Vision Workshop*, (1994).
2. Baker, H. H., and Balles, R., "Generalizing epipolar-plane image analysis on the spatiotemporal surface", *International Journal of Computer Vision*, vol. 3, No.1, pp. 33-49 (1989).
3. Balles, R. C., Baker, H. H., and Marimont, D. H., "Epipolar-Plane Image Analysis: An Approach to Determining Structure from Motion", *Int. J. Computer Vision*, Vol.1, No.1, pp. 7–55 (1987).
4. Debevec, P., Taylor, C., and Malik, J., "Modeling and rendering architecture from photographs: a hybrid geometry- and image-based approach", *proc. Siggraph'96*, pp.11-20 (1996).
5. Gortler, S., Grzeszczuk, R., Szeliski, R., and Cohen, M., "The Lumigraph", *proc. Siggraph'96*, pp.43-54 (1996).
6. Levoy, M., and Hanrahan, P., "Light field rendering", *proc. Siggraph'96*, pp.31-42 (1996).
7. Ohta, Y., and Kanade, T., "Stereo by Intra- and Inter-Scanline Search Using Dynamic Programming", *IEEE PAMI*, Vol.7, No.2, pp. 139–154 (1985).
8. Seits, S., and Dyer, C., "View Morphing", *proc. Siggraph'96*, pp.21-30 (1996).
9. Yasuno, T., and Suzuki, S., "Occlusion analysis of spatiotemporal images for surface reconstruction", *proc. 4th British Machine Vision Conference* pp. 549-558 (1993).
10. Werner, T., Hersch, R, and Hlavac, V., "Rendering real-world objects using view interpolation", *proc. ICCV'95*, pp.957-962 (1995).

(a) An example of input images.

(b) An input epipolar image.



(c) Candidate feature lines.

(d) Synthesized epipolar image.

(e) Difference between input and synthesized images.

Figure 2: Experimental results.