

CaMor: Screw Interpolation between Perspective Projections of Partial Views of Rectangular Images

Gokul Raghuraman, Nicholas Barrash, Jarek Rossignac

College of Computing, Georgia Institute of Technology, USA

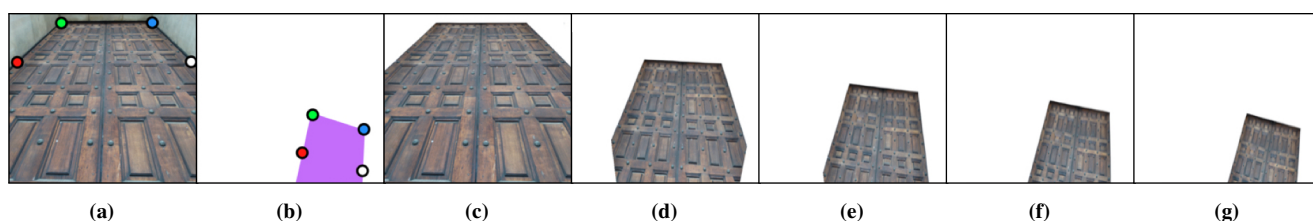


Figure 1: Initial photograph (a) and 4 user-specified points (red and white on edges). The final (target) pose (b) with four corresponding points displaced by the user. Views from successive frames (c) through (g) of a camera motion computed by CaMor.

Abstract

CaMor is a tool for generating an animation from a single drawing or photograph that represents a partial view of a perspective projection of a planar shape or image that contains portions of only 3 edges of an unknown rectangle. The user identifies these portions and indicates where the corresponding lines should be at the end of the animation. CaMor produces a non-affine animation of the entire plane by combining (1) a new rectification procedure that identifies the orientation in 3D of a rectangle from the partial image of its perspective projection, (2) a depth adjustment that ensures that the two rectified rectangles are congruent in 3D, (3) a screw motion that interpolates in 3D between the two congruent shapes, and (4) at each frame, a perspective projection of a user-selected portion of the original image. The animation may be modified interactively by adjusting the final positions of the lines or the focal length. We suggest applications to the animation of hand-drawn scenes, to the morph between two photographs, and to the intuitive design of camera motions for indoor and street scenes.

Categories and Subject Descriptors (according to ACM CCS): I.3.7 [Computer Graphics]: Three-Dimensional Graphics and Realism—Animation, I.4.1 [Image Processing and Computer Vision]: I.4.8 Scene Analysis—Motion

1. Introduction

The objective of this work is to support the automatic generation of animations of the perspective projection of a rigid planar shape or image, M . The animation should appear as a natural 3D motion.

The inputs to our system, which we called CaMor (short for Camera Morph), are: (1) a subset, P , of a photograph or drawing that represents the perspective projection of a planar, but rigid, feature or image (such as a building façade), M , (2) a set of three initial lines, I_1 , I_2 , and I_3 , in the image, each containing the perspective projection of an edge of an unknown rectangle, $Rect$, on M at the initial pose, and (3) the corresponding set of three target (i.e., final) lines, T_1 , T_2 , and T_3 , each containing the perspective projection of the corresponding edge of $Rect$ at the final pose of M . These lines

may be extracted automatically through image processing or specified interactively by the user.

Options for interactive specification of initial lines include:

4 points: Point **A** on the first edge, **B** at the corner where first and second edge join, **C** where second and third edge join, and **D** further along the third edge (Fig. 1a).

2 points and 2 tangents: Points **B** and **C** as defined above and vectors \overrightarrow{BA} and \overrightarrow{CD} (Fig. 3 bottom).

3 pairs of points: S_1 and E_1 along the first edge, S_2 and E_2 along the second edge, and S_3 and E_3 along the third edge (Fig. 4).

The same options exist for specifying the target lines. The first two options assume that two consecutive corners of $Rect$ are visible. The last option does not. CaMor does not require that any portion of the fourth edge of $Rect$ be visible. This is useful when $Rect$

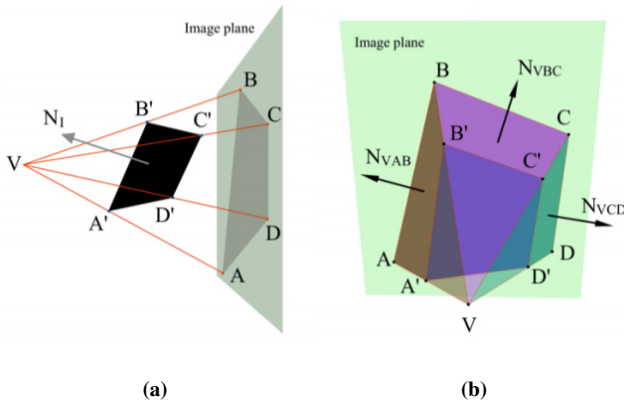


Figure 2: (a) Points A' , B' , C' , D' on the border of a rectangle, $Rect$, in π with normal \vec{N}_I . B' and C' are two adjacent corners of the rectangle. Their respective perspective projections are A , B , C , and D . (b) \vec{N}_{VAB} is the normal to the plane through viewpoint V and I_1 . That plane contains L_1 . Similarly for \vec{N}_{VBC} , and \vec{N}_{VCD} .

is the floor or wall of a room that cannot be captured in its entirety in a single photograph. For simplicity, and without loss of generality, we use the **4 points** option when explaining the details of our solution (Fig. 2a). It can be computed from either of the other two options.

CaMor assumes that P is a portion of the perspective projection of a planar image or shape, M , that lies in some unknown plane π in 3D and that there are 3 lines, L_1 , L_2 , and L_3 , in π , such that: (1) L_i projects onto I_i , for $i \in \{1, 2, 3\}$, (2) L_2 is orthogonal to both L_1 and L_3 , (3) there exists a rigid body transformation R such that $R(L_i)$ projects onto T_i , for $i \in \{1, 2, 3\}$. We want to compute π .

CaMor computes the orientation (normal \vec{N}_I) of π in 3D from I_1 , I_2 , I_3 , and from a given value of the focal length f of the camera or perspective projection of the drawing. It then computes the rectified image M in 3D of P (that is produced by applying the inverse of the perspective projection assumed to have produced P).

Similarly, CaMor computes the orientation $\vec{N}_P = R(\vec{N}_I)$ of $R(\pi)$ from T_1 , T_2 , and T_3 . Then it computes a relative displacement in depth (i.e., along the viewing direction) that matches the length of the edge, E of $Rect$ that projects onto edge (B, C) to the length of $R(E)$. Then, CaMor computes the rigid body transformation R .

Finally, CaMor computes the interpolating screw motion, R_t , such that R_0 is the identity and R_1 is R . For each time value t in $[0, 1]$, it produces a new frame of M in 3D and projects the desired subset onto the canvas. Note that the portion of M selected by the user does not need to be inside the rectangle.

The output produced by CaMor is either a 2D animation or a discrete set of image frames from that animation. Each image is the perspective projection of the image portion, P , that lies on an instance of M at a specific position and orientation along a 3D screw motion computed by CaMor (see Fig. 1c–g). It is parametrized by the focal length f , which either is precisely defined by the camera used or can be set by the artist.

Anticipated applications include: (1) animating a drawing of a planar portion of a 3D scene, (2) designing a natural camera motion in 3D, and (3) constructing natural morphs between photographs.

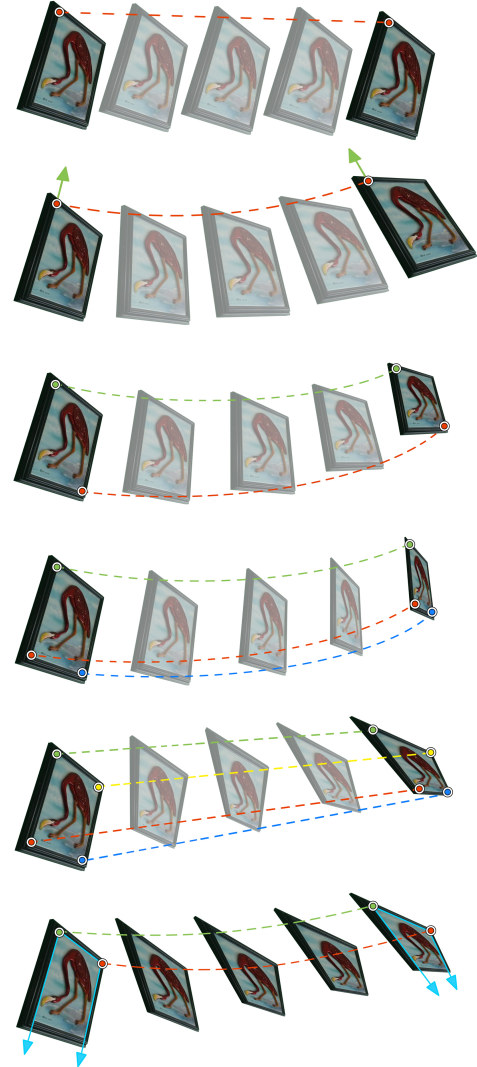


Figure 3: Color-coded initial points (left), corresponding targets (right), and 3 intermediate frames, for (top-to-bottom) a translation, a rotation, a logarithmic spiral, a SAM, a LERP, and CaMor, defined using the 2 point and 2 tangents option (cyan arrows).

2. Prior Work

Techniques for computing the motion of an image from initial and target point-pairs include: (1) **translation** specified by one pair of points (initial point and its target), (2) **rotation** specified by such a pair of points and a pair of (initial and target) directions, (3) **logarithmic spiral** specified by two pairs of points [WNS*10], (4) **SAM** (Steady Affine Morph) [RV11] specified by three pairs of points, and (5) a vertex **LERP** specified by four pairs of points and combined with a non-affine bi-linear map. Examples of these

solutions are shown in Fig. 3. None of these produces a realistic perspective projection of the 3D animations of a rigid motion. The CaMor solution, proposed here and shown at the bottom of Fig. 3, does.

CaMor does not require registration, arbitrary decompositions of matrices, or other optimization processes, that are used in more general solutions [XRS02]. Prior morphing techniques described in [BN92] and [CW93] provide 3D visual effects, while using only 2D transformations. The technique described in [SD96] assumes vertical alignment of two views. [TE99] use corresponding loops in two images to morph their interiors via compatible triangulation and texture mapping. In *Tour Into the Picture*, [HAA97] generate animations from a single photograph, but their approach can handle only images with one vanishing point, and is therefore not suitable for more general perspective views of rectangles. [ZwH04] show how to compute the camera parameters (focal length, aspect ratio, and rigid transformation) from the perspective image of an unknown rectangle, but require that the entire rectangle or at least 2 points on each edge be visible. Variations of the technique described in [ZwH04] that require that the focal length be known were proposed in [Har89] and [TSB95]. A generalization of the technique in [Har89] that can recover the orientation of any arbitrary quadrilateral is presented in [Pen91]. However it is required for the dimensions of the quadrilateral to be known in advance. [WSB05] use parallelepipeds to discover the focal length and orientation and position of surfaces, and to create 3D models from a single image. An interactive technique is discussed in [CZS*13] to recover the orientation of a square given the perspective projection of two of its edges. The computation of the 3D orientation of a circle from its projection is discussed in [Phi97]. [GMMB00] and [CRZ00] extract 3D information from a scene using two distinct vanishing points without knowledge of the focal length. As noted in [TSB95] and [WNR88], vanishing point methods can become unstable when recovering the orientation of planes that are nearly parallel to the image surface, but work well in street views, as demonstrated in [DLN07].

The CaMor solution to rectification presented here is different from the above solutions and has two advantages: (1) it uses a previously unknown, yet simple and elegant, closed-form solution and (2) it requires only the identification of lines that contain the projections of 3 edges of a rectangle. Hence, CaMor is useful when we are only given two perspective images (with the same field of view) of the same rectangle where only 3 of the four edges have a visible portion in each image.

Its main drawback (that the focal length must be known) is acceptable in applications where the focal length is fixed (as in a cell phone), when it is dictated by the viewing configuration (relative distance between the viewer and the screen), or when the artist wishes to control it.

3. Rectification

We are given screen locations \mathbf{A} , \mathbf{B} , \mathbf{C} , and \mathbf{D} and focal length f . We assume the notation shown in Fig. 3.

We compute the normal $\vec{\mathbf{N}}_I$ of π , in the camera frame with origin at the center \mathbf{O} of the image plane and view point $\mathbf{V} = (0, 0, -f)$.

The details of the computation are presented below.

$$\vec{\mathbf{N}}_{VAB} = \vec{\mathbf{V}}\mathbf{B} \times \vec{\mathbf{V}}\mathbf{A}, \quad \vec{\mathbf{N}}_{VBC} = \vec{\mathbf{V}}\mathbf{C} \times \vec{\mathbf{V}}\mathbf{B}, \quad \vec{\mathbf{N}}_{VCD} = \vec{\mathbf{V}}\mathbf{D} \times \vec{\mathbf{V}}\mathbf{C}. \quad (1)$$

The key observation is that edges $(\mathbf{A}', \mathbf{B}')$ and $(\mathbf{D}', \mathbf{C}')$ of the unknown 3D rectangle are parallel to vector

$$\vec{\mathbf{T}}_1 = \vec{\mathbf{N}}_{VAB} \times \vec{\mathbf{N}}_{VCD}, \quad (2)$$

because $\vec{\mathbf{N}}_{VAB}$ and $\vec{\mathbf{N}}_{VCD}$ are orthogonal to these edges, regardless of the position of \mathbf{V} . Edge $(\mathbf{A}', \mathbf{B}')$ is parallel to

$$\vec{\mathbf{T}}_2 = \vec{\mathbf{T}}_1 \times \vec{\mathbf{N}}_{VBC} \quad (3)$$

because it is orthogonal to $\vec{\mathbf{T}}_1$ and to $\vec{\mathbf{N}}_{VBC}$. Hence,

$$\vec{\mathbf{N}}_I = \vec{\mathbf{T}}_1 \times \vec{\mathbf{T}}_2 \quad (4)$$

Note that this formulation does not work for the singular cases where \mathbf{O} , \mathbf{B} and \mathbf{C} are collinear. In this case, $\vec{\mathbf{N}}_{VBC}$ is parallel to $\vec{\mathbf{T}}_1$. As a result, $\vec{\mathbf{T}}_2$ and therefore also $\vec{\mathbf{N}}_I$ are null.

Our solution simplifies when \mathbf{A} and \mathbf{D} are the vertices of a rectangle. In this case, $\vec{\mathbf{N}}_I$ can be computed as follows:

$$\begin{aligned} \vec{\mathbf{N}}_{VDA} &= \mathbf{VA} \times \vec{\mathbf{V}}\mathbf{D} \\ \vec{\mathbf{T}}_1 &= \vec{\mathbf{N}}_{VAB} \times \vec{\mathbf{N}}_{VCD} \\ \vec{\mathbf{T}}_2 &= \vec{\mathbf{N}}_{VBC} \times \vec{\mathbf{N}}_{VDA} \\ \vec{\mathbf{N}}_I &= \vec{\mathbf{T}}_2 \times \vec{\mathbf{T}}_1 \end{aligned}$$

We use corresponding expressions to compute the normal vector $\vec{\mathbf{N}}_T$ to $R(\pi)$ from the target points. Note that, as in all of the prior art cited above, all 3D locations recovered through rectification are defined up to a common scale factor. We fix that scale factor here by fixing a distance d and defining

$$\mathbf{B}' = \mathbf{V} + d\hat{\mathbf{V}}\mathbf{B} \quad (5)$$

where $\hat{\mathbf{V}}\mathbf{B}$ denotes the normalized (unit) version of vector $\vec{\mathbf{V}}\mathbf{B}$. (Note that the choice of d is arbitrary and does not affect the resulting screw motion.) Hence plane π is defined by point \mathbf{B}' and normal $\vec{\mathbf{N}}_I$, and similarly, plane $R(\pi)$ is defined by point

$$\mathbf{F}' = \mathbf{V} + h\hat{\mathbf{V}}\mathbf{F} \quad (6)$$

and normal $\vec{\mathbf{N}}_T$. We use \mathbf{E}' , \mathbf{F}' , \mathbf{G}' , and \mathbf{H}' to denote the 3D points that project on the corresponding target points.

4. Reverse Projection

Points \mathbf{X}' on π satisfy $\vec{\mathbf{X}}'\mathbf{B}' \cdot \vec{\mathbf{N}}_I = 0$. Hence, the 3D location \mathbf{X}' of an arbitrary point on π can be recovered from its perspective projection \mathbf{X} by substituting $\mathbf{F} + s\vec{\mathbf{F}}\mathbf{X}$ for \mathbf{X}' in $\vec{\mathbf{X}}'\mathbf{B}' \cdot \vec{\mathbf{N}}_I = 0$. Solving for s yields

$$s = \frac{\vec{\mathbf{F}}\mathbf{B}' \cdot \vec{\mathbf{N}}_I}{\vec{\mathbf{F}}\mathbf{X} \cdot \vec{\mathbf{N}}_I} \quad (7)$$

5. Depth Adjustment

Since R is an isometry, to fix the scale factor for plane $R(\pi)$, we compute h such that $|\mathbf{B}'\mathbf{C}'| = |\mathbf{F}'\mathbf{G}'|$.

Hence, h is computed by solving: $d|\vec{\mathbf{V}}\mathbf{C} - \vec{\mathbf{V}}\mathbf{B}| = h|\vec{\mathbf{V}}\mathbf{G} - \vec{\mathbf{V}}\mathbf{F}|$. We now use d to compute \mathbf{B}' using (5). We then use \mathbf{B}' and $\vec{\mathbf{N}}_I$

to compute s using (7). We can then compute \mathbf{A}' , \mathbf{C}' , and \mathbf{D}' . We perform similar computations with h and $\vec{\mathbf{N}}_{\mathbf{T}}$ using (5) and (7) to compute \mathbf{F}' , \mathbf{E}' , \mathbf{G}' , and \mathbf{H}' .

6. Screw Motion

We advocate the screw motion, because (1) it is uniquely defined by the two frames, (2) it is affine invariant, (3) it minimizes rotation angle and translation distance, (4) it is symmetric (Swapping the two frames yields the same motion backwards), and it feels natural. We define the initial pose $\{\mathbf{O}_0, \hat{\mathbf{I}}_0, \hat{\mathbf{J}}_0, \hat{\mathbf{K}}_0\}$ on π with $\mathbf{O}_0 = \mathbf{B}'$, $\hat{\mathbf{I}}_0 = \mathbf{B}'\mathbf{C}'$, $\hat{\mathbf{K}}_0 = \vec{\mathbf{N}}_{\mathbf{T}}$, and $\hat{\mathbf{J}}_0 = \hat{\mathbf{K}}_0 \times \hat{\mathbf{I}}_0$. Similarly, we define the target pose $\{\mathbf{O}_1, \hat{\mathbf{I}}_1, \hat{\mathbf{J}}_1, \hat{\mathbf{K}}_1\}$ for $\mathbf{R}(\pi)$. We compute and animate the minimal screw motion using the simple formulae provided in [RK01]. The pose $\{\mathbf{O}_t, \hat{\mathbf{I}}_t, \hat{\mathbf{J}}_t, \hat{\mathbf{K}}_t\}$ at time t is $\mathbf{R}^t(\{\mathbf{O}_0, \hat{\mathbf{I}}_0, \hat{\mathbf{J}}_0, \hat{\mathbf{K}}_0\})$.

7. Animation

To animate a vector drawing or geometric model, we animate each vertex \mathbf{X} and redraw the model at each frame. To do so, we compute \mathbf{X}' as explained in Section 4, produce $\mathbf{X}_t' = \mathbf{R}^t(\mathbf{X}')$, and compute the current location \mathbf{X}_t on the screen through perspective projection. To animate an image, we rely on the artist to draw a polygonal contour around the desired portion of the image on the screen and we record the screen locations (x_i, y_i) of its vertices \mathbf{X}_i as texture coordinates (u_i, v_i) . Then, we animate the points \mathbf{X}_i as explained in the previous paragraph, but render the intermediate frames using a user-specified portion of the original image as texture map. Fig. 4 illustrates an animation produced by CaMor using two input photographs and six pairs of input points (initial and target).

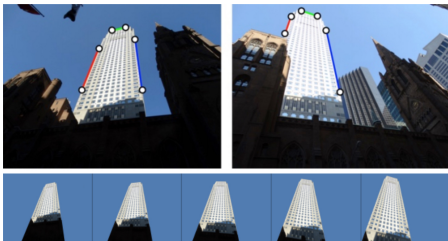


Figure 4: CaMor animation from two images. Top: The user inputs 6 points in each image to define the initial (left) and target (right) poses. Bottom: Frames from the resulting animation.

8. Applications

We discuss 3 potential applications for CaMor. Its advantage over prior art is limited to situations where one of the edges of the rectangle is entirely hidden in the photographs or drawings. CaMor demonstrates that a physically plausible, 3D camera motion may be derived from two partial images and used to produce a natural and pleasing interpolation

An animator or architect may wish to create an animation that suggests a 3D motion of a planar shape or image that has been drawn by hand (in a perspective view) or cut out from a photograph. CaMor facilitates the process. The artist defines the input edges

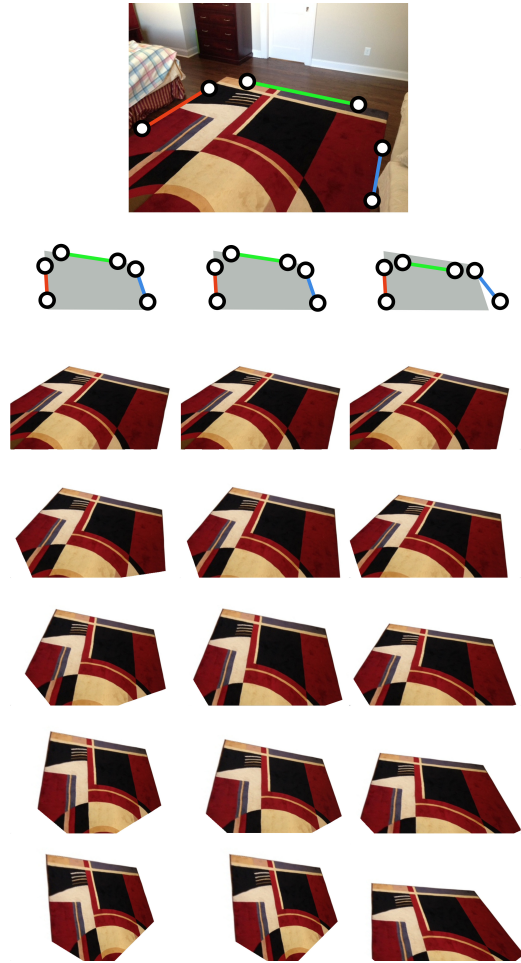


Figure 5: Generating 3 different motions from a single photograph. The original photograph is shown at the top. The user specifies the initial pose using six points (2 per edge). Each target pose is defined by six corresponding points. Left: The animation for a focal length $f = 4.6\text{mm}$. Center: The animation for the same target pose as before, but using $f = 3.2\text{mm}$. Right: The animation using $f = 4.6\text{mm}$, but with the target points slightly displaced.

(using 4 or 6 points as discussed in Section 1) and simply displaces these points to new (target) locations. The animation may be tuned by adjusting the target points or the focal length f . Fig. 5 shows the process on a photograph of a designer carpet.

It is often difficult to design pleasing camera motions from two-dimensional references such as photographs or drawings. CaMor allows the artist to easily explore a variety of natural camera motions. The user simply clicks and drags the locations of the target points. Further, the user can vary the focal length f to produce more dramatic effects.

An interesting variant of this approach is when the user has two photographs of a scene, each showing a different portion of the same planar shape or image, but from a different perspective (Fig. 6). We can combine the forward morph produced by CaMor

as explained above with the time reversed backward morph from the final to the initial configurations. A cross-fade of the two textures may be used to provide higher resolution in places where the two textures overlap. This solution works because a screw motion equals the time reverse of its inverse. Furthermore, when a portion of the moving image or planar is not visible in the initial photograph, it may be visible in the final one and vice versa.

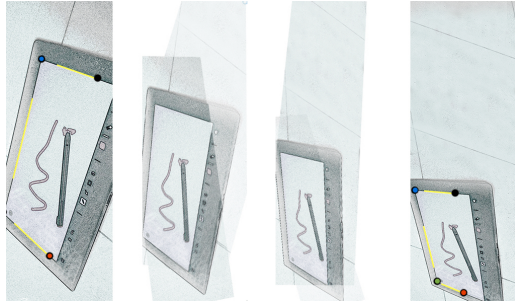


Figure 6: Cross-fading a forward (left-to-right) CaMor animation between two images with its inverse.

9. Accuracy of results

We explored how errors in the graphically selected locations of the input points alter the animation. To simplify sensitivity analysis, we performed the following experiment: (1) we start with the vertices A' , B' , C' and D' of a known rectangle in 3D, (2) we compute their perspective projections, A , B , C , and D , (3) we perturb the location of these projections using a displacement of a controlled magnitude but a random direction, (4) we perform the rectification, and (5) we report the error as the angle between the normal vectors of the original and rectified planes. To allow this comparison, we adjust the depth of the rectified points so as to match the actual distance from F to A' in the input model. We run this test on 9 configurations with different orientations. We chose edge lengths of about 500 pixels. For each configuration, we repeated the test 500 times, each time with a randomly generated set of various perturbations measured in pixels.

The results are reported in Fig. 7 for cases where edge (A' , B') has half the length of (C' , D') and for cases with the same orientations but where edge (A' , B') has the same length as (C' , D'). The rectified normal was computed using the steps from Section 3.

10. Conclusion

CaMor computes a 3D animation of a planar rectangle or image from an input picture of its perspective projection and from 3 pairs of edges drawn on the screen. Each edge in a pair traces a portion of the initial and final perspective projections of one edge of the rectangle. Our solution is novel and improves over prior art by its simplicity, numeric robustness, and by the fact that one of the edges of the rectangle may be totally hidden in the input picture. We discuss applications to 3D animation, camera motion planning, and virtual fly through in architectural design.

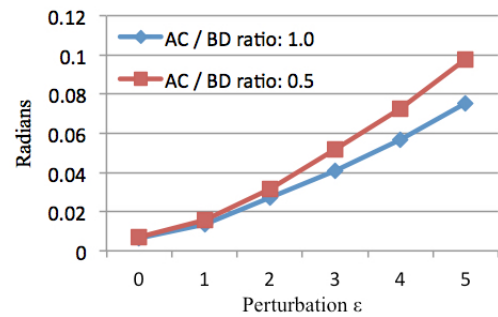


Figure 7: The average angle between the actual normal and the normal computed using randomly perturbed perspective projection points for different ratios between the lengths of two of the sides of the reference object.

References

- [BN92] BEIER T., NEELY S.: Feature-based image metamorphosis. *SIGGRAPH Comput. Graph.* 26, 2 (July 1992), 35–42. URL: <http://doi.acm.org/10.1145/142920.134003>, doi:10.1145/142920.134003. 3
- [CRZ00] CRIMINISI A., REID I., ZISSERMAN A.: Single view metrology. *International Journal of Computer Vision* 40, 2 (2000), 123–148. URL: <http://dx.doi.org/10.1023/A:1026598000963>, doi:10.1023/A:1026598000963. 3
- [CW93] CHEN S. E., WILLIAMS L.: View interpolation for image synthesis. In *Proceedings of the 20th Annual Conference on Computer Graphics and Interactive Techniques* (New York, NY, USA, 1993), SIGGRAPH '93, ACM, pp. 279–288. URL: <http://doi.acm.org/10.1145/166117.166153>, doi:10.1145/166117.166153. 3
- [CZS*13] CHEN T., ZHU Z., SHAMIR A., HU S.-M., COHEN-OR D.: 3-sweep: Extracting editable objects from a single photo. *ACM Trans. Graph.* 32, 6 (Nov. 2013), 195:1–195:10. URL: <http://doi.acm.org/10.1145/2508363.2508378>, doi:10.1145/2508363.2508378. 3
- [DLN07] DELAGE E., LEE H., NG A. Y.: *Automatic Single-Image 3d Reconstructions of Indoor Manhattan World Scenes*. Springer Berlin Heidelberg, Berlin, Heidelberg, 2007, pp. 305–321. URL: http://dx.doi.org/10.1007/978-3-540-48113-3_28, doi:10.1007/978-3-540-48113-3_28. 3
- [GMMB00] GUILLOU E., MENEVEAUX D., MAISEL E., BOUATOUCH K.: Using vanishing points for camera calibration and coarse 3d reconstruction from a single image. *The Visual Computer* 16, 7 (2000), 396–410. URL: <http://dx.doi.org/10.1007/PL00013394>, doi:10.1007/PL00013394. 3
- [HAA97] HORRY Y., ANJYO K.-I., ARAI K.: Tour into the picture: Using a spidery mesh interface to make animation from a single image. In *Proceedings of the 24th Annual Conference on Computer Graphics and Interactive Techniques* (New York, NY, USA, 1997), SIGGRAPH '97, ACM Press/Addison-Wesley Publishing Co., pp. 225–232. URL: <http://dx.doi.org/10.1145/258734.258854>, doi:10.1145/258734.258854. 3
- [Har89] HARALICK R. M.: Determining camera parameters from the perspective projection of a rectangle. *Pattern Recogn.* 22, 3 (Feb. 1989), 225–230. URL: [http://dx.doi.org/10.1016/0031-3203\(89\)90071-X](http://dx.doi.org/10.1016/0031-3203(89)90071-X), doi:10.1016/0031-3203(89)90071-X. 3
- [Pen91] PENNA M. A.: Determining camera parameters from the perspective projection of a quadrilateral. *Pattern Recogn.* 24, 6 (Apr. 1991), 533–541. URL: [http://doi.org/10.1016/0031-3203\(89\)90071-X](http://doi.org/10.1016/0031-3203(89)90071-X). 3

- [https://dx.doi.org/10.1016/0031-3203\(91\)90019-2](https://dx.doi.org/10.1016/0031-3203(91)90019-2),
doi:10.1016/0031-3203(91)90019-2. 3
- [Phi97] PHILIP J.: *An Algorithm for Determining the Position of a Circle in 3D from Its Perspective 2D Projection*. TRITA / MAT / MA: TRITA. Royal Institute of Technology, 1997. URL: <https://books.google.com/books?id=SRPMOQAACAAJ>. 3
- [RK01] ROSSIGNAC J., KIM J.: Computing and visualizing pose-interpolating 3d motions. *Computer-Aided Design* 33, 4 (2001), 279 – 291. URL: <http://www.sciencedirect.com/science/article/pii/S0010448500000865>, doi:[https://dx.doi.org/10.1016/S0010-4485\(00\)00086-5](https://dx.doi.org/10.1016/S0010-4485(00)00086-5). 4
- [RV11] ROSSIGNAC J., VINACUA A.: Steady affine motions and morphs. *ACM Trans. Graph.* 30, 5 (Oct. 2011), 116:1–116:16. URL: <http://doi.acm.org/10.1145/2019627.2019635>, doi:10.1145/2019627.2019635. 2
- [SD96] SEITZ S. M., DYER C. R.: View morphing. In *Proceedings of the 23rd Annual Conference on Computer Graphics and Interactive Techniques* (New York, NY, USA, 1996), SIGGRAPH '96, ACM, pp. 21–30. URL: <http://doi.acm.org/10.1145/237170.237196>, doi:10.1145/237170.237196. 3
- [TE99] TAL A., ELBER G.: Image morphing with feature preserving texture. *Computer Graphics Forum* 18, 3 (1999), 339–348. URL: <http://dx.doi.org/10.1111/1467-8659.00354>, doi:10.1111/1467-8659.00354. 3
- [TSB95] TAN T. N., SULLIVAN G. D., BAKER K. D.: Recovery of intrinsic and extrinsic camera parameters using perspective views of rectangles. In *Proceedings of the 1995 British Conference on Machine Vision (Vol. 1)* (Surrey, UK, UK, 1995), BMVC '95, BMVA Press, pp. 177–186. URL: <http://dl.acm.org/citation.cfm?id=236190.236207>. 3
- [WNR88] WEISS R. S., NAKATANI H., RISEMAN E. M.: An error analysis for surface orientation from vanishing points. vol. 0974, pp. 187–195. URL: <http://dx.doi.org/10.1117/12.948458>, doi:10.1117/12.948458. 3
- [WNS*10] WHITED B., NORIS G., SIMMONS M., SUMNER R., GROSS M., ROSSIGNAC J.: Betweenit: An interactive tool for tight inbetweening. *Comput. Graphics Forum (Proc. Eurographics)* 29, 2 (2010), 605–614. URL: <https://cgl.ethz.ch/publications/papers/paperWhi10.php>. 2
- [WSB05] WILCZKOWIAK M., STURM P., BOYER E.: Using geometric constraints through parallelepipeds for calibration and 3d modeling. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 27, 2 (Feb 2005), 194–207. doi:10.1109/TPAMI.2005.40. 3
- [XRS02] XIAO J., RAO C., SHAH M.: View Interpolation for Dynamic Scenes. In *Eurographics 2002 - Short Presentations* (2002), Eurographics Association. doi:10.2312/egs.20021005. 3
- [ZwH04] ZHANG Z., WEI HE L.: Note-taking with a camera: whiteboard scanning and image enhancement. In *2004 IEEE International Conference on Acoustics, Speech, and Signal Processing* (May 2004), vol. 3, pp. iii–533–6 vol.3. doi:10.1109/ICASSP.2004.1326599. 3