Planospheric camera model

Todd E. Litwin*

Jet Propulsion Laboratory, California Institute of Technology, Pasadena, California,
United States

Abstract. Realtime stereo processing, particularly in computationally constrained environments such as the Mars 2020 Perseverance rover, requires image rectification to enable efficient one-dimensional correlation. A camera model has been designed to perform better image rectification for fish-eye lenses with a particular emphasis on the engineering cameras on Perseverance. The primary innovation is the use of a unit projection sphere rather than an image plane as the rectifying surface. For stereo rectification, the rows (for a horizontal pair) or the columns (for a vertical pair) of both cameras must lie on the same epipolar plane. Thus, we use a pair of moving planes to define the rows and columns. Each plane rotates about a dedicated static axis passing through the sphere center. Virtual pixels are located where the planes intersect with each other and the unit sphere. © The Authors. Published by SPIE under a Creative Commons Attribution 4.0 International License. Distribution or reproduction of this work in whole or in part requires full attribution of the original publication, including its DOI. [DOI: [10.1117/1.JRS.15.034517](https://doi.org/10.1117/1.JRS.15.034517)]

Keywords: camera model; epipolar alignment; fish-eye lens; Mars 2020; stereo vision; visual odometry.

Paper 210218 received Apr. 28, 2021; accepted for publication Aug. 31, 2021; published online Sep. 16, 2021.

1 Introduction

Researchers at NASA's Jet Propulsion Laboratory (JPL) have been developing and working with geometric camera models for decades. In the late 1970s, the Yakimovsky and Cunningham linear, perspective-projection model was developed,¹ followed in the 1990s by Gennery's update of that model to include nonlinear distortion,² and then in the 2000s his further update supporting both perspective-projection and fish-eye optics in a single hybrid model.³

In the 1990s, the linear model was used to perform stereo correlation in near realtime for a horizontal stereo pair. To enable this with limited computational power, the disparity search was restricted to the horizontal dimension. For this to work, corresponding rows must be made to line up in the left and right images, i.e., they must be brought into epipolar alignment. No real-world stereo camera pair is perfectly epipolar-aligned. Early systems approximated epipolar alignment by mechanically aligning the cameras. Given the low-resolution systems of the time, the residual misalignment was often not apparent. However, the process was tedious and unreliable.

To achieve epipolar alignment for more demanding cases, a technique was developed in which virtual, epipolar-aligned, co-located, camera models were generated synthetically from the true camera models. The real images were then resampled as if by the virtual cameras into corresponding, epipolar-aligned images. This resulted in images well suited to performing one-dimensional stereo correlation.⁴

The technique graduated from research to a real application in 2003 with NASA's Mars Exploration Rover (MER) mission's rovers Spirit and Opportunity to support obstacle detection while driving.⁵ It was used again in 2011 on the Mars Science Laboratory (MSL) mission's Curiosity rover.⁶ These missions had nonlinear camera optics, with both perspective-projection lenses that included radial-like distortion in some cases (Navcams) and fish-eye-like optics in others (Hazcams).^{7,8} The rectification technique worked well for the former, but less well for the latter.

This is not surprising given that the virtual, rectified models were represented as Yakimovsky and Cunningham linear models.⁹ This choice was reasonable for optics that were fundamentally

*Address all correspondence to Todd E. Litwin, telitwin@jpl.nasa.gov

Fig. 1 (a) MER Hazcam stereo pair of indoor test facility. (b) Images rectified with linear camera models. (c) Images rectified with new planospheric camera models.

perspective-projection, even with some (modest) radial distortion. Rectifying fish-eye images to linear models, however, significantly distorts the pixel data.

In Fig. 1, the top row shows a stereo pair of images taken by MER Hazcams mounted on a test rover with a view of an indoor sandbox. The middle row shows those same images rectified using linear models. Note in particular that the pixels are clearly stretched around the perimeter of the images. See Fig. 2 from MSL Hazcams on the surface of Mars for a view that shows more clearly how the pixels in the middle of the images are compressed.

Fig. 2 (a) MSL Hazcam stereo pair of the surface of Mars. (b) Images rectified with linear camera models. (c) Images rectified with new planospheric camera models.

For the earlier missions, the disadvantages of linear rectification were mitigated in part by using the perspective-projection Navcams preferentially. On the new Mars 2020 mission's Perseverance rover, that is not possible since all the engineering cameras have fish-eye optics.¹⁰

The bottom rows of Figs. 1 and 2 show new images created using the planospheric model described in this paper. The model's better fit to fish-eye geometry is evidenced by less distortion at the pixel level throughout the images.

Work on the JPL planospheric camera model began in 2002 when the deficiencies of the existing approach for the MER Hazcams became evident. Since then similar concepts, albeit with different representations and less specialized objectives, have been pursued independently in the research community.^{11,12}

2 Planospheric Camera Model

As with the earlier rectification approach, a virtual camera model is generated coincident with the model for the real camera. Consider a unit sphere whose center coincides with the center of projection of the virtual lens. The image passes through the center and inverts before falling on the back inner surface of the sphere. This is the essence of the model's conceptual geometry for a new type of imaging surface. As with perspective projection, it is more intuitive to picture oneself looking out at the world from the sphere center, and seeing the image as appearing on the forward half of the sphere, upright. All descriptions below will follow this convention.

2.1 Model Definition

Define the image coordinates as

x = column, horizontal coordinate, increasing to the right,

y = row, vertical coordinate, increasing downward. Then let the model parameters be

$\mathbf{c}$ = 3D position of the sphere center,

$\mathbf{a}_x$ = unit column-plane rotation axis, passing through the sphere center, typically vertical and pointing down so that positive rotations (by the right-hand rule) will rotate the forward half of the plane in the (rightward) direction of increasing column (as projected on the forward hemisphere),

$\mathbf{a}_y$ = unit row-plane rotation axis, passing through the sphere center, typically horizontal and pointing left so that positive rotations (by the right-hand rule) will rotate the forward half of the plane in the (downward) direction of increasing row (as projected on the forward hemisphere),

$\mathbf{n}_x$ = unit normal vector to the column plane when x equals zero, pointing in the same direction as the cross product of $\mathbf{a}_x$ with an outward-pointing vector that also lies in the plane,

$\mathbf{n}_y$ = unit normal vector to the row plane when y equals zero, pointing in the same direction as the cross product of $\mathbf{a}_y$ with an outward-pointing vector that also lies in the plane,

s_x = column scale factor to convert between x coordinate and rotation around $\mathbf{a}_x$, expressed in radians/pixel,

s_y = row scale factor to convert between y coordinate and rotation around $\mathbf{a}_y$, expressed in radians/pixel.

The external coordinate system in which the vectors $\mathbf{c}$, $\mathbf{a}_x$, $\mathbf{a}_y$, $\mathbf{n}_x$, and $\mathbf{n}_y$ are expressed is arbitrary. Later, it will be called the 3D "world" coordinate system. It is any convenient coordinate system, in which the camera may be posed arbitrarily. For the Mars rover missions, the vehicle body's coordinate system is chosen.

The scalars s_x and s_y can be the same, but this is not required. They would be identical if one models square pixels, where the rows and columns are equally spaced. Where they are not, these scalars will differ.

See Figs. 3 and 4 for renderings of a horizontal stereo pair of planospheric models from both the front and back. The line containing the row-plane rotation axis vectors passes through both the left and right projection centers, and the column-plane rotation axis vectors are parallel. The virtual pixels are located where the planes intersect with each other and with the forward half of the unit sphere. Note that the row planes in this case are the epipolar planes of the camera system (in the projective-geometry sense) and are independent of any camera-model formulation.

The descriptions of $\mathbf{a}_x$ and $\mathbf{a}_y$ above suggest that they are orthogonal. They typically are, but they need not be. If there is a reason to make those axes not quite orthogonal, the model can support that. This is also true of the earlier models. Consider a trinocular-stereo system where, say, there are two cameras in a common horizontal configuration, and a third camera directly above or below one of the first two. Here, to prevent performing completely independent rectification for each stereo pair, a common geometry for all three can be chosen such that the row plane passes through the

Fig. 3 Stereo planospheric models as viewed from in front of the cameras.

Fig. 4 Stereo planospheric models as viewed from behind the cameras.

horizontal pair and the column plane passes through the vertical pair (see Fig. 5). In any real-world system, these two plane-rotation axes will not be precisely orthogonal.

2.2 Projecting from World to Image

The connection between 2D image coordinates and 3D world coordinates is made by rotating the planes through angles corresponding to the pixel address (x, y) , and then following the line of intersection where the planes cross. Where the line intersects the sphere identifies the image pixel, all 3D world points that fall along the line project onto that pixel.

The projection, therefore, from a 3D world point $\mathbf{p}$ to a 2D image coordinate is

$$x = \frac{1}{s_x} \text{atan2}(\sin \theta_x, \cos \theta_x),$$

$$y = \frac{1}{s_y} \text{atan2}(\sin \theta_y, \cos \theta_y),$$

where

$$\begin{aligned}\sin \theta_x &= (\mathbf{n}_x \times \mathbf{n}'_x) \cdot \mathbf{a}_x, \\ \sin \theta_y &= (\mathbf{n}_y \times \mathbf{n}'_y) \cdot \mathbf{a}_y, \\ \cos \theta_x &= \mathbf{n}_x \cdot \mathbf{n}'_x, \\ \cos \theta_y &= \mathbf{n}_y \cdot \mathbf{n}'_y, \\ \mathbf{n}'_x &= \text{unit}(\mathbf{a}_x \times (\mathbf{p} - \mathbf{c})), \\ \mathbf{n}'_y &= \text{unit}(\mathbf{a}_y \times (\mathbf{p} - \mathbf{c})),\end{aligned}$$

Fig. 5 Trinocular stereo planospheric models as viewed from in front of the cameras.

and $\mathbf{n}'_x$ and $\mathbf{n}'_y$ are the column and row normals for the planes containing $\mathbf{p}$, respectively. The intermediate (and unsolved-for) terms θ_x and θ_y are the angles between those normal vectors and their corresponding zero-based model parameters $\mathbf{n}_x$ and $\mathbf{n}_y$, respectively.

The partial derivatives of the 2D coordinates with respect to the 3D point are

$$\frac{\partial x}{\partial \mathbf{p}} = \frac{[\mathbf{n}'_x]^T}{s_x |\mathbf{u}|}, \quad \frac{\partial y}{\partial \mathbf{p}} = \frac{[\mathbf{n}'_y]^T}{s_y |\mathbf{v}|},$$

where $\mathbf{u}$ and $\mathbf{v}$ are the perpendicular vectors between $\mathbf{p}$ and the rotation axes

$$\begin{aligned} \mathbf{u} &= (\mathbf{p} - \mathbf{c}) - [(\mathbf{p} - \mathbf{c}) \cdot \mathbf{a}_x] \mathbf{a}_x, \\ \mathbf{v} &= (\mathbf{p} - \mathbf{c}) - [(\mathbf{p} - \mathbf{c}) \cdot \mathbf{a}_y] \mathbf{a}_y. \end{aligned}$$

2.3 Projecting from Image to World

Using the same geometric image as before, the projection from a 2D image coordinate (x, y) to an outward-facing 3D unit vector $\mathbf{u}$ in world coordinates, anchored at the sphere's center, is

$$\mathbf{u} = \text{unit}(\mathbf{n}'_x \times \mathbf{n}'_y),$$

where

$$\begin{aligned} \mathbf{n}'_x &= \mathbf{n}_x \text{ rotated around } \mathbf{a}_x \text{ by } s_x x, \\ \mathbf{n}'_y &= \mathbf{n}_y \text{ rotated around } \mathbf{a}_y \text{ by } s_y y, \end{aligned}$$

which, using Rodriguez' rotation formula¹³ simplified for our case of orthogonal unit vectors, is

$$\mathbf{n}'_x = \mathbf{n}_x \cos(s_x x) + (\mathbf{a}_x \times \mathbf{n}_x) \sin(s_x x), \quad \mathbf{n}'_y = \mathbf{n}_y \cos(s_y y) + (\mathbf{a}_y \times \mathbf{n}_y) \sin(s_y y).$$

The partial derivatives of the 3D unit pointing vector with respect to the 2D coordinates x or y are

$$\frac{\partial \mathbf{u}}{\partial x} = \left[\sqrt{\mathbf{v} \cdot \mathbf{v}} \frac{\partial \mathbf{v}}{\partial x} - \left(\frac{\partial \mathbf{v}}{\partial x} \cdot \mathbf{v} \right) \mathbf{u} \right] / (\mathbf{v} \cdot \mathbf{v}),$$

$$\frac{\partial \mathbf{u}}{\partial y} = \left[\sqrt{\mathbf{v} \cdot \mathbf{v}} \frac{\partial \mathbf{v}}{\partial y} - \left(\frac{\partial \mathbf{v}}{\partial y} \cdot \mathbf{v} \right) \mathbf{u} \right] / (\mathbf{v} \cdot \mathbf{v}),$$

where

$$\mathbf{v} = \mathbf{n}'_x \times \mathbf{n}'_y,$$

$$\partial \mathbf{v} / \partial x = s_x (\mathbf{a}_x \times \mathbf{n}'_x) \times \mathbf{n}'_y,$$

$$\partial \mathbf{v} / \partial y = s_y \mathbf{n}'_x \times (\mathbf{a}_y \times \mathbf{n}'_y).$$

To derive these partial derivatives start by noting that

$$\frac{\partial \mathbf{n}'_x}{\partial x} = s_x \mathbf{a}_x \times \mathbf{n}'_x,$$

$$\frac{\partial \mathbf{n}'_y}{\partial y} = s_y \mathbf{a}_y \times \mathbf{n}'_y.$$

Let

$$\mathbf{u} = \text{unit}(\mathbf{v}) = \mathbf{v} / |\mathbf{v}| = \mathbf{v} / \sqrt{\mathbf{v} \cdot \mathbf{v}}.$$

Then the partial derivative of $\mathbf{u}$ is

$$\begin{aligned} \partial \mathbf{u} / \partial x &= \frac{\partial}{\partial x} (\mathbf{v} / \sqrt{\mathbf{v} \cdot \mathbf{v}}) \\ &= \left[\sqrt{\mathbf{v} \cdot \mathbf{v}} \frac{\partial \mathbf{v}}{\partial x} - \mathbf{v} \left(\frac{\partial}{\partial x} \sqrt{\mathbf{v} \cdot \mathbf{v}} \right) \right] / (\mathbf{v} \cdot \mathbf{v}) \\ &= \left\{ \sqrt{\mathbf{v} \cdot \mathbf{v}} \frac{\partial \mathbf{v}}{\partial x} - \mathbf{v} \left[\frac{\partial}{\partial x} (\mathbf{v} \cdot \mathbf{v}) \right] / (2\sqrt{\mathbf{v} \cdot \mathbf{v}}) \right\} / (\mathbf{v} \cdot \mathbf{v}) \\ &= \left[\sqrt{\mathbf{v} \cdot \mathbf{v}} \frac{\partial \mathbf{v}}{\partial x} - \mathbf{v} \left(\frac{\partial \mathbf{v}}{\partial x} \cdot \mathbf{v} + \mathbf{v} \cdot \frac{\partial \mathbf{v}}{\partial x} \right) / (2\sqrt{\mathbf{v} \cdot \mathbf{v}}) \right] / (\mathbf{v} \cdot \mathbf{v}) \\ &= \left[\sqrt{\mathbf{v} \cdot \mathbf{v}} \frac{\partial \mathbf{v}}{\partial x} - \mathbf{v} \left(2 \frac{\partial \mathbf{v}}{\partial x} \cdot \mathbf{v} \right) / (2\sqrt{\mathbf{v} \cdot \mathbf{v}}) \right] / (\mathbf{v} \cdot \mathbf{v}) \\ &= \left[\sqrt{\mathbf{v} \cdot \mathbf{v}} \frac{\partial \mathbf{v}}{\partial x} - \mathbf{v} \left(\frac{\partial \mathbf{v}}{\partial x} \cdot \mathbf{v} \right) / \sqrt{\mathbf{v} \cdot \mathbf{v}} \right] / (\mathbf{v} \cdot \mathbf{v}) \\ &= \left[\sqrt{\mathbf{v} \cdot \mathbf{v}} \frac{\partial \mathbf{v}}{\partial x} - \left(\frac{\partial \mathbf{v}}{\partial x} \cdot \mathbf{v} \right) (\mathbf{v} / \sqrt{\mathbf{v} \cdot \mathbf{v}}) \right] / (\mathbf{v} \cdot \mathbf{v}) \\ &= \left[\sqrt{\mathbf{v} \cdot \mathbf{v}} \frac{\partial \mathbf{v}}{\partial x} - \left(\frac{\partial \mathbf{v}}{\partial x} \cdot \mathbf{v} \right) \mathbf{u} \right] / (\mathbf{v} \cdot \mathbf{v}), \end{aligned}$$

and similarly for $\partial \mathbf{u} / \partial y$. The set of partial derivatives can be used to derive covariance matrices for future sensitivity analysis against model parameters and viewing geometry.

2.4 Computational Complexity

The advantages of the planospheric model come at some cost. Compare the above to the Yakimovsky and Cunningham linear model, which has a very efficient representation. To project from a 3D world coordinate to image row and column coordinates takes only one vector subtraction, three dot products, and two identical scalar divisions. To project from image to world takes two vector subtractions, three scalar–vector multiplications, one cross product, one square root, and one scalar division. It does not use any trigonometric functions.

The difference between the models can be seen when performing the calculations required to prepare for image rectification. This involves projecting from image to world using the virtual camera model, either linear or planospheric, and then from world to image using the calibrated model, for each pixel in the image. When performing this for a pair of 1280×960 images on the Mars 2020 flight processor, it takes approximately 47 s when using the linear model and 53 s when using the planospheric model. For Mars 2020, where these calculations are performed during initialization and not on the fly, this is considered to be an acceptable cost.

3 Preliminary Results

The planospheric camera model has been incorporated into the flight software of the Mars 2020 rover Perseverance. Operators can select using the legacy linear model or the new planospheric model to rectify the images used by stereo correlation and visual odometry. A preliminary comparison of the models has been performed with images taken in JPL's MarsYard.

To evaluate the effects of the different geometries, we can look at visual odometry's use of template matching to find corresponding features from one frame to the next.¹⁴ Such matching is more robust the less feature appearances change as a function of location in the rectified image. A rectification model that results in more uniform appearance across the image should produce better feature-matching behavior. We see this with the planospheric model.

Testing was performed in mid-2020 on a test rover, "Scarecrow," a mobility test platform inherited from MSL, and upgraded to exercise Mars 2020's mobility software. It was outfitted with cameras similar to those on the flight vehicle. Visual odometry using the new model detected and matched approximately 44% more features on average for turns in place of 20 deg and 30 deg than it did with the old model. For circular drives, the advantage was 10% on average (see Table 1).

Table 1 Comparison between the linear and planospheric camera models on the number of visual-odometry inliers found per step when testing on Scarecrow rover in JPL's MarsYard in mid-2020.

Scarecrow: mean number of inliers		
Motion	Linear	Planospheric
Turn in place, 20 deg	227	310
Turn in place, 30 deg	121	192
Circular drive, 1-m radius, 20 deg steps	150	165

Table 2 Comparison between the linear and planospheric camera models on the number of visual-odometry inliers found per step when testing on VSTB rover in JPL's MarsYard in late-2020.

VSTB: mean number of inliers		
Motion	Linear	Planospheric
Turn in place, 10 deg	371	442
Turn in place, 20 deg	188	292
Turn in place, 30 deg	94	140
Arc, 1 m, 0 deg	167	169
Arc, 1 m, 10 deg	150	167
Arc, 1 m, 20 deg	69	86

In late 2020, additional testing was performed on a more flight-like test rover, “VSTB,” a higher-fidelity testbed than Scarecrow, designed specifically for the Mars 2020 mission with many engineering-model and other flight-like components. Visual odometry using the planospheric model found about 34% more features during turns in place of 10 deg, 20 deg, and 30 deg than it did with the linear model. For 1-m drives of 0 deg, 10 deg, and 20 deg arcs, the comparison was about 9% in the new model’s favor (see Table 2).

4 Conclusion

The planospheric camera model described here is potentially a better fit for rectifying fish-eye images than the legacy linear model JPL has been using in all its one-dimensional near-realtime stereo processing to date. As the Mars 2020 mission unfolds, and the operators experiment with the old and new models, we will see how this potential proves out in real operations on the Martian surface.

Acknowledgments

The author would like to thank Mike McHenry and Philip Twu for help testing the model for stereo and visual odometry. He would also like to thank Yalin Xiong, who many years ago stated that he knew how to rectify fish-eye images, although without providing any details. It was on the strength of that statement that the author was encouraged to think enough about the subject to develop the new model. The research was carried out at the Jet Propulsion Laboratory, California Institute of Technology, under a contract with the National Aeronautics and Space Administration (80NM0018D0004). The author has no relevant financial interests in the paper and no other potential conflicts of interest to disclose. Copyright © 2021. California Institute of Technology. Government sponsorship acknowledged.

References

1. Y. Yakimovsky and R. T. Cunningham, “A system for extracting three-dimensional measurements from a stereo pair of TV cameras,” *Comput. Graphics Image Process.* **7**, 195–210 (1978).
2. D. B. Gennery et al., “Sensing and perception research for space telerobotics at JPL,” in *Proc. IEEE Int. Conf. Rob. and Autom.*, Raleigh, North Carolina, Vol. 1898, pp. 311–317 (1987).
3. D. B. Gennery, “Generalized camera calibration including fish-eye lenses,” *Int. J. Comput. Vision* **68**, 239–266 (2006).
4. L. Matthies, T. Litwin, and A. Kelly, “Obstacle detection for unmanned ground vehicles: a progress report,” in *Int. Symp. Rob. Res.*, Munich (1995).
5. S. Goldberg, M. Maimone, and L. Matthies, “Stereo vision and rover navigation software for planetary exploration,” in *IEEE Aerosp. Conf. Proc.*, Big Sky, Montana, Vol. 5, pp. 2025–2036 (2002).
6. A. Rankin et al., “Driving curiosity: Mars rover mobility trends during the first seven years,” in *IEEE Aerosp. Conf.* (2020).
7. J. N. Maki et al., “The Mars Exploration Rover engineering cameras,” *J. Geophys. Res.* **108**(E12), 12-1–12-24 (2003).
8. J. Maki et al., “The Mars Science Laboratory engineering cameras,” *Space Sci. Rev.* **170**, 77–93 (2012).
9. P. Grandjean, “Stereo images warping and triangulation,” Internal JPL tech note (1992).
10. J. N. Maki et al., “The Mars 2020 engineering cameras and microphone on the Perseverance rover: a next-generation imaging system for Mars exploration,” *Space Sci. Rev.* **216**, 137 (2020).
11. S. Li, “Real-time spherical stereo,” in *18th Int. Conf. Pattern Recognit. (ICPR’06)*, Hong Kong, pp. 1046–1049 (2006).

12. D. Wan, J. Zhou, and D. Zhang, "A spherical rectification for dual-PTZ-camera system," in *IEEE Int. Conf. Acoust. Speech and Signal Process. – ICASSP '07* (2007).
13. Wikipedia, "Rodrigues' rotation formula," 2021, https://en.wikipedia.org/wiki/Rodrigues'_rotation_formula (accessed 16 March 2021).
14. A. Howard, "Real-time stereo visual odometry for autonomous ground vehicles," in *Int. Conf. Rob. and Syst. (IROS)* (2008).

Todd E. Litwin has been an engineer at NASA's Jet Propulsion Laboratory for over 40 years, and has worked on all four of its Mars rover missions. He is currently a senior engineer in the Mobility and Robotics Systems section, and has been an author on over 30 publications. He received a BS degree in applied physics from Harvey Mudd College in 1979.