Computer Vision Fundamentals: Cameras, Projection and Epipolar Geometry

Oct 28, 2026 · @Luiz Velho with Claude

1. From light to images

A camera is a device that turns a 3D scene into a 2D image, and almost every problem in geometric computer vision consists of inverting that map. This note develops the mathematics needed to do so: projective geometry, the camera model, calibration, and the geometry relating two views of the same scene.

The most general description of the light in a scene is the **plenoptic function**. It gives the radiance seen from every position (x, y, z), in every direction (θ, ϕ), at every wavelength λ and time t:

$$L = L(x, y, z, \theta, \phi, \lambda, t) \tag{8}$$

In free space radiance is constant along a ray, so for a static scene and a fixed wavelength the plenoptic function reduces to a 4D **light field**: the radiance carried by each oriented line in space. An image is a 2D slice of that light field: it records the rays passing through one point, the camera's centre of projection, and indexes them by where they cross the image plane.

This viewpoint explains why geometry is central. Each pixel u corresponds to a ray, and each ray corresponds to a set of 3D points that project to the same pixel. Depth is lost in the projection. Recovering it requires either knowledge of the camera and the scene, or more than one view. The rest of this note makes these statements precise.

2. Projective geometry essentials

Perspective projection divides by depth, so it is nonlinear in Euclidean coordinates. Homogeneous coordinates turn it into a matrix product, which is why projective geometry is the natural language of vision.

Homogeneous coordinates. A point (x, y) of the plane is represented by any vector (wx, wy, w) with w ≠ 0. Two homogeneous vectors represent the same point when one is a nonzero multiple of the other, written u ~ v. To return to Euclidean coordinates, divide by the last component:

$$\tilde{u} = \begin{pmatrix} u_1 \\ u_2 \\ u_3 \end{pmatrix} \longmapsto \left(\frac{u_1}{u_3}, \frac{u_2}{u_3} \right) \tag{10}$$

The set of all such equivalence classes is the **projective plane** $\mathbb{P}^2$. In the same way, 3D points X = (X, Y, Z) become 4-vectors (X, Y, Z, 1) in **projective space** $\mathbb{P}^3$.

Points at infinity. Vectors with last component 0, such as (x, y, 0), have no Euclidean counterpart. They represent directions, or points at infinity. Parallel lines meet at such a point, which is exactly what we see in a photograph where railway tracks converge at a vanishing point. Projective geometry treats these points like any other, which removes many special cases.

Lines and duality. A line in $\mathbb{P}^2$ is also a 3-vector l = (a, b, c), the set of points with ax + by + c = 0. A point u lies on l when

$$l^\top u = 0 \tag{12}$$

The line through two points is their cross product, l = u × v, and the intersection of two lines is l × m. Points and lines play symmetric roles. This duality is used constantly in epipolar geometry (Section 8).

Projective transformations. Any invertible 3×3 matrix H maps $\mathbb{P}^2$ to itself and preserves collinearity. Such a map is a homography. It has 8 degrees of freedom, since H and λH act identically. Homographies form a hierarchy of increasingly restrictive groups:

Group	Matrix form	Degrees of freedom (2D)	Preserves
Projective	general invertible H	8	collinearity, cross-ratio
Affine	last row (0, 0, 1)	6	parallelism, ratios of lengths on a line
Similarity	rotation × scale + translation	4	angles, ratios of lengths
Euclidean	rotation + translation	3	lengths, areas

A camera projection is a map from $\mathbb{P}^3$ to $\mathbb{P}^2$, represented by a 3×4 matrix. The next two sections build it.

3. The camera model

The standard model is the **pinhole camera**: every ray reaching the image passes through a single point, the centre of projection. It ignores lens blur and distortion, which are handled as corrections on top of it.

Camera parameters. A pinhole camera is fully described by six quantities:

Symbol	Parameter	Role
o	Position	centre of projection in world coordinates
n	View direction	the optical axis, normal to the image plane
up	Up vector	fixes the rotation of the camera about n
f	Focal distance	distance from o to the image plane
c	Image centre	principal point, where the optical axis meets the image
s	Image size	extent of the image, in pixels

The first three describe where the camera is and where it looks. The last three describe the camera's internal geometry. This split reappears as the extrinsic and intrinsic parameters of Section 4.

Perspective projection. Place the camera at the origin looking down the Z axis, with the image plane at Z = f. A point (X, Y, Z) projects along the ray through the origin. By similar triangles,

$$x = f\,\frac{X}{Z}, \qquad y = f\,\frac{Y}{Z} \tag{14}$$

The division by Z makes distant objects smaller. In homogeneous coordinates the same map is linear:

$$\begin{pmatrix} x \\ y \\ 1 \end{pmatrix} \sim \begin{pmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix} \tag{16}$$

Types of projection. The pinhole model is one member of a family of projections from 3D to 2D:

- **Orthographic (affine).** Rays are parallel to the optical axis, so x = X and y = Y. There is no foreshortening with depth. It models a camera infinitely far away with infinite zoom.
- **Weak perspective.** Orthographic projection followed by a uniform scale f / Z_0, where Z_0 is the average depth of the object. It is accurate when the object's depth range is small compared with its distance.
- **Perspective.** The full pinhole model above. Parallel lines in the scene converge at vanishing points.
- **Others.** Wide-angle and fisheye lenses, catadioptric (mirror) cameras and pushbroom satellite sensors follow different models, usually treated as the pinhole model plus a nonlinear mapping.

Affine projections are the case where the last row of the projection matrix is (0, 0, 0, 1). They are easier to work with and are often used to initialise algorithms that later switch to full perspective.

4. The fundamental equation u = Px

All of geometric vision rests on one equation. A 3D point x, in homogeneous coordinates, projects to an image point u through a 3×4 camera matrix P:

$$u \sim P\,x, \qquad u \in \mathbb{P}^2,\ x \in \mathbb{P}^3,\ P \in \mathbb{R}^{3\times 4} \tag{18}$$

The symbol ~ means equality up to scale. In practice one writes λu = Px, where λ is the depth of the point along the viewing ray.

Intrinsic and extrinsic parts. The camera matrix factors into two pieces, written KA on the slides:

$$P = K\,A = K\,[\,R \mid t\,] \tag{20}$$

The **extrinsic** part A = $R|t$ is a rigid (Euclidean, hence affine) transformation. It moves world coordinates into the camera's frame. R is a 3×3 rotation whose rows are the camera's axes, built from the view direction n and the up vector. The translation is t = −R o, where o is the camera position.

The **intrinsic** part K is an upper-triangular 3×3 matrix. It performs the projection and converts metric image coordinates to pixels:

$$K = \begin{pmatrix} f_x & \gamma & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{pmatrix} \tag{22}$$

Here f_x and f_y are the focal distance measured in horizontal and vertical pixel units, (c_x, c_y) is the image centre, and γ is a skew term, zero for almost all modern sensors. When pixels are square, f_x = f_y = f.

Degrees of freedom. K has 5 parameters, R has 3 and t has 3, for 11 in total. This matches the 12 entries of P minus one for the overall scale. Any 3×4 matrix whose left 3×3 block is invertible can be factored this way, as Section 5 shows.

Useful facts about P.

- The camera centre o is the null vector of P: P (o, 1) = 0. It is the only point with no image.
- The left 3×3 block M = KR determines the direction of each viewing ray. The ray through pixel u is o + μ M^{-1} u.
- Points at infinity (d, 0) project to vanishing points K R d. They are independent of the camera position.

The key assumption: correspondence. The equation relates a specific 3D point to a specific pixel. Every algorithm that uses it assumes we know which pixel u belongs to which point x, or which pixels in different images show the same point. Establishing these correspondences, through feature detection and matching, is often the hardest practical step. The geometry in this note takes them as given.

5. Two vision problems: calibration and reconstruction

The equation u = Px has three ingredients. Fixing which ones are known gives the basic problems of vision:

Problem	Known	Unknown
Projection (rendering)	P, x	u
Camera calibration	u, x	P
Shape reconstruction	u, P	x

Projection is the forward problem of computer graphics. Calibration and reconstruction are inverse problems, and both rely on correspondences $u \in I \subset \mathbb{R}^2$, $x \in \mathbb{R}^3$.

5.1 Camera calibration

Calibration estimates P from n known 3D points x_i and their images u_i, typically the corners of a calibration target.

Direct Linear Transform (DLT). Since u_i and Px_i are parallel vectors, their cross product vanishes: $u_i \times Px_i = 0$. Write $p_j^\top$ for the rows of P and $u_i = (u_i, v_i, w_i)$. Each correspondence gives two independent linear equations in the 12 entries of P:

$$\begin{pmatrix} 0^\top & -w_i\, x_i^\top & v_i\, x_i^\top \\ w_i\, x_i^\top & 0^\top & -u_i\, x_i^\top \end{pmatrix} \begin{pmatrix} p_1 \\ p_2 \\ p_3 \end{pmatrix} = 0 \tag{24}$$

Stacking n correspondences gives a 2n×12 system A p = 0. With 11 degrees of freedom, six points in general position suffice. With noisy data, use more points and take p as the right singular vector of A with smallest singular value, which minimises ‖Ap‖ subject to ‖p‖ = 1. The points must not all lie on one plane, or the system is degenerate.

Two practical steps matter a great deal:

1. **Normalise the data first.** Translate and scale the image points (and the 3D points) so they are centred at the origin with average distance √2 (√3 in 3D). Without this, the linear system is badly conditioned and the solution is inaccurate.
2. **Refine nonlinearly.** The DLT minimises an algebraic error with no geometric meaning. Use its result to initialise a Levenberg–Marquardt minimisation of the reprojection error, the sum of squared image distances between u_i and the projection of x_i. Lens distortion parameters are added at this stage.

Recovering K, R and t. Write P = $M|p_4$. Since M = KR with K upper-triangular and R orthogonal, an **RQ decomposition** of M yields both. Fix signs so that K has a positive diagonal. Then t = K^{-1} p_4, and the camera centre is o = $-M^{-1}$ p_4.

In practice, the most common approach is Zhang's method. It uses several images of a planar checkerboard, estimates a homography for each one, and solves for K from the constraints these homographies impose.

5.2 Shape reconstruction (triangulation)

Reconstruction estimates a 3D point x from its images in cameras with known matrices. One view is not enough: a pixel only fixes a ray. Two views give two rays, and x lies where they meet.

For cameras P_0 and P_1 with rows p_j and p'_j, the same cross-product trick gives four equations in x:

$$\begin{pmatrix} u_0\, p_3^\top - p_1^\top \\ v_0\, p_3^\top - p_2^\top \\ u_1\, p_3'^\top - p_1'^\top \\ v_1\, p_3'^\top - p_2'^\top \end{pmatrix} x = 0 \tag{26}$$

Here (u_0, v_0) and (u_1, v_1) are the Euclidean pixel coordinates in each image. Solve again by SVD. With noise the two rays do not intersect exactly. The linear solution is then refined by minimising reprojection error in both images. The accuracy of depth falls as the angle between the rays shrinks, so a short baseline gives poor depth.

6. Variants: one view or many

The basic equation becomes a family of problems once we index points by i and cameras by k.

Single view: many points, one camera.

$$u_i = P\, x_i \tag{28}$$

The correspondence problem here is **pattern recognition**: given a 3D object (a model, or a calibration target) and an image of it, find the pairs (u_i, x_i). Once they are found, the same equation answers two questions. If K is unknown, solve for all of P (calibration, Section 5.1). If K is known, solve only for the 6 pose parameters R and t. This is the **Perspective-n-Point (PnP)** problem, solvable from as few as 3 points (P3P, up to four solutions) and used in augmented reality and robot localisation.

Multiple views: many points, various cameras.

$$u_{ik} = P_k \, x_i \tag{30}$$

Now the 3D points are usually unknown, and the correspondence problem is between images: given K images, find pairs (u_i, u_j) of pixels that show the same scene point, written i ↔ j. This is **feature matching** when the images are taken from distinct viewpoints, and **tracking** when they are consecutive frames of a video. A typical pipeline detects keypoints (corners or blobs), describes each by a local descriptor such as SIFT or ORB, and matches descriptors between images. Some matches will be wrong, which is why the estimation methods of Section 9 must be robust.

The multi-view equation has far more unknowns than the single-view one, but it also has far more equations: each matched point seen in k images contributes 2k equations and only 3 unknowns. That surplus is what makes reconstruction without a calibration object possible.

7. Multi-view metrology

Writing the camera matrices in factored form makes the unknowns explicit:

$$u_{ik} = P_k \, x_i = [\, K_k A_k \,] \, x_i \tag{32}$$

Three classical problems differ in which of K_k, A_k and x_i are known:

Problem	Given	Find
Stereo reconstruction	point images (u_i^0), (u_i^1); both cameras P_0, P_1	the 3D points x_i
Structure from motion (SfM)	correspondences (u_i, u_j), i ↔ j; intrinsics K_k	camera motions A_k and 3D points x_i
Self-calibration	correspondences (u_i, u_j), i ↔ j only	the full cameras P_k = KA_k

Stereo reconstruction. Both cameras are calibrated in advance, usually a rigid stereo rig. Reconstruction reduces to finding correspondences and triangulating (Section 5.2). Epipolar geometry turns the correspondence search into a 1D search along a line (Section 8). For dense stereo, the images are first **rectified**, warped so that epipolar lines become horizontal rows. The match for each pixel is then found on the same row, and its horizontal offset, the disparity d, gives depth Z = f b / d, where b is the baseline.

Structure from motion. A single calibrated camera moves through a scene, or many photos are taken with cameras of known intrinsics. Both the camera motion (R_k, t_k) and the scene structure x_i are unknown. Two views fix the relative pose through the essential matrix (Section 9). Further views are added incrementally by PnP and triangulation. A final **bundle adjustment** jointly refines all cameras and points by minimising the total reprojection error:

$$\min_{\{A_k\},\{x_i\}} \sum_{i,k} \left\| u_{ik} - \pi\left(K_k A_k\, x_i\right) \right\|^2 \tag{34}$$

Here π divides by the last homogeneous coordinate. A fundamental limitation: the reconstruction is defined only up to a global similarity. Scaling the scene and the camera translations by the same factor leaves every image unchanged, so absolute scale needs outside information.

Self-calibration. Nothing about the cameras is known in advance. From correspondences alone one can recover cameras and structure up to an arbitrary **projective** transformation H: replacing P_k by P_k H and x_i by H^{-1} x_i leaves every image unchanged. Self-calibration uses constraints on the intrinsics, for example that K is the same for all views or that pixels are square, to upgrade this projective reconstruction to a metric one. It is what allows reconstruction from uncontrolled photo collections.

The three problems form a progression of decreasing prior knowledge. Each relies on the two-view relations developed next.

8. Epipolar geometry

Two views of the same point are tied together by one geometric fact: the scene point, both camera centres and both image points all lie in a single plane. As a consequence, the match of a pixel in one image must lie on a known line in the other, which reduces the search for correspondences from 2D to 1D.

Consider two cameras with centres C_1 and C_2 observing a scene point M, which appears at m_1 in the first image and m_2 in the second. The construction involves five objects:

- **Baseline.** The line joining the two camera centres C_1 and C_2.
- **Epipoles e_1 and e_2.** The points where the baseline pierces each image plane. Equivalently, e_1 is the image of C_2 in the first camera, and e_2 the image of C_1 in the second.
- **Epipolar plane.** The plane through C_1, C_2 and M. Every scene point defines one, and all of them contain the baseline, rotating around it like the pages of a book.
- **Epipolar lines l_1 and l_2.** The intersection of the epipolar plane with each image. l_2 is the image, in the second camera, of the viewing ray from C_1 through m_1. All epipolar lines in an image pass through its epipole.
- **Epipolar constraint.** The match m_2 of m_1 must lie on l_2, and symmetrically m_1 must lie on l_1.

[embedded content: two-view epipolar geometry · 2 cameras, 1 scene point]

The first camera alone cannot tell M from any other point on its viewing ray, such as M'. All these candidates project onto the same epipolar line l_2 in the second image, here at m_2 and m_2'. A second view therefore resolves the depth, and the search for the match need only run along l_2.

Two configurations are worth recognising. When the cameras are side by side with parallel optical axes, as in a rectified stereo rig, the baseline is parallel to both image planes. The epipoles are then at infinity and the epipolar lines are parallel horizontal rows. When the camera moves straight forward, the epipole is the point the camera is heading towards (the focus of expansion), and the epipolar lines radiate from it.

Section 9 turns this geometric picture into algebra.

9. The fundamental and essential matrices

The epipolar constraint is captured by a single 3×3 matrix. When the intrinsics are unknown it is the **fundamental matrix** F. When they are known it is the **essential matrix** E, a metric object from which the relative camera pose can be read off.

In this section x and x' denote corresponding image points in homogeneous pixel coordinates, following the slides.

9.1 The fundamental matrix (K unknown)

For every pair of corresponding points,

$$x'^{\top} F\, x = 0 \tag{36}$$

F maps a point in the first image to its epipolar line in the second, l' = F x, and $F^{\top}$ does the reverse, l = $F^{\top}$ x'. The constraint above simply states that x' lies on l' (compare $l^{\top}u = 0$ in Section 2).

For cameras P = K$I|0$ and P' = K'$R|t$, the fundamental matrix is

$$F = K'^{-\top}\,[t]_{\times}\,R\,K^{-1}, \qquad [t]_{\times} = \begin{pmatrix} 0 & -t_3 & t_2 \\ t_3 & 0 & -t_1 \\ -t_2 & t_1 & 0 \end{pmatrix} \tag{37}$$

where $t\times$ is the matrix of the cross product, $t\times$ v = t × v. Its key properties:

- **Rank 2.** $t\times$ is singular, so det F = 0.
- **7 degrees of freedom.** 9 entries, minus one for scale, minus one for the rank constraint.
- **Epipoles are its null vectors.** F e = 0 and $F^{\top}$ e' = 0. Every epipolar line passes through the epipole.
- **Depends only on image measurements.** F can be estimated from correspondences alone, with no knowledge of the cameras. It determines the cameras only up to a projective transformation, which is the starting point for self-calibration (Section 7).

9.2 The essential matrix (K known)

When K and K' are known, convert pixels to normalised camera coordinates $\hat{x} = K^{-1} x$. In these coordinates the cameras are $I|0$ and $R|t$, and the constraint becomes

$$\hat{x}'^{\top} E\, \hat{x} = 0, \qquad E = [t]_{\times} R = K'^{\top} F\, K \tag{38}$$

E has only 5 degrees of freedom: 3 for rotation and 2 for the direction of translation, since its scale is undetermined. It is characterised by its singular values, which take the form (σ, σ, 0). Because it is built from R and t, E is a metric object: it encodes the actual relative motion between the two cameras.

Recovering the pose from E. Take the SVD E = U diag(1, 1, 0) $V^{\top}$ and let

$$W = \begin{pmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix}, \qquad R \in \{\, UWV^{\top},\ UW^{\top}V^{\top} \,\}, \qquad t = \pm\, u_3 \tag{39}$$

where u_3 is the last column of U. This gives four candidate poses. Only one places the reconstructed points in front of both cameras (the **cheirality** condition). Triangulating a single correspondence is enough to choose it. The translation is recovered only up to scale, which is the scale ambiguity of SfM noted in Section 7.

9.3 The 8-point algorithm (estimating F)

Expanding $x'^{\mathsf{T}} F x = 0$ for $x = (x, y, 1)$ and $x' = (x', y', 1)$ gives one linear equation per correspondence in the 9 entries f of F:

$$\begin{pmatrix} x'x,\ x'y,\ x',\ y'x,\ y'y,\ y',\ x,\ y,\ 1 \end{pmatrix} \cdot f = 0 \tag{40}$$

1. Normalise the points in each image (centre at the origin, mean distance $\sqrt{2}$). This is Hartley's normalisation and it is essential for accuracy.
2. Stack $n \geq 8$ equations into $A f = 0$ and take f as the singular vector of A with smallest singular value.
3. Enforce rank 2: compute the SVD of the estimate, set its smallest singular value to zero and recompose.
4. Undo the normalisation: $F = T'^{\mathsf{T}} \hat{F} T$, where T and T' are the normalising transforms.

A 7-point variant uses the constraint $\det F = 0$ directly. It solves a cubic and gives one or three solutions. Because F needs no calibration, the 8-point algorithm is the entry point for uncalibrated reconstruction and self-calibration.

9.4 The 5-point algorithm (estimating E)

E has 5 degrees of freedom, so 5 correspondences suffice when K is known. Nistér's 5-point algorithm combines the 5 linear equations with the cubic constraints that define an essential matrix, $2EE^{\mathsf{T}}E - \operatorname{tr}(EE^{\mathsf{T}})E = 0$ and $\det E = 0$. The result is a polynomial of degree 10 with up to 10 real solutions. Each candidate is checked against further correspondences.

The 5-point algorithm is the standard tool for calibrated SfM and visual odometry. It needs fewer points than the 8-point method and remains valid when the scene is planar, where the 8-point method breaks down.

9.5 Robust estimation with RANSAC

Real matches contain outliers, and a single bad match ruins a least-squares estimate. **RANSAC** handles this by repeating three steps: sample a minimal set of s correspondences, fit a model, and count the inliers whose epipolar error (for example the Sampson distance) is below a threshold. The model with most inliers is then refit to all of its inliers. The number of samples needed to find one clean set with probability p, when a fraction w of matches are inliers, is

$$N = \frac{\log(1 - p)}{\log(1 - w^s)} \tag{41}$$

With half the matches correct ($w = 0.5$) and $p = 0.99$, the 8-point algorithm needs about 1,177 samples, the 7-point variant about 587, and the 5-point algorithm about 145. This is the practical reason minimal solvers matter.

Degenerate configurations. If the cameras only rotate (t = 0), there is no baseline and no depth information. The two images are then related by a homography H = K' R K^{-1}, and F is undefined. The same homography relation holds for any planar scene. Robust pipelines test for these cases before estimating F or E.

10. Summary and further reading

The whole note unfolds from u = Px, the factorisation P = K$R|t$, and the two-view constraint x'^{T} F x = 0. The table collects the main objects:

Object	Size	Degrees of freedom	Relates	Minimal estimate
Camera matrix P	3×4	11	3D point ↔ image point	6 points (DLT)
Intrinsics K	3×3, upper-triangular	5	camera frame ↔ pixels	from P by RQ, or Zhang's method
Pose $R\|t$	3×4	6	world frame ↔ camera frame	3 points (P3P), K known
Homography H	3×3	8	plane ↔ image, or two images under pure rotation	4 points
Fundamental matrix F	3×3, rank 2	7	image point ↔ epipolar line, K unknown	7 points (8 linear)
Essential matrix E	3×3, singular values (σ, σ, 0)	5	normalised point ↔ epipolar line, K known	5 points

Exercises.

1. Show that the camera centre o satisfies P (o, 1) = 0, and derive o = $-M^{-1}$ p_4.
2. Prove that the epipoles are the null vectors of F and F^{T}.
3. Derive F = K'^{-T} $t\times$ R K^{-1} from the two camera matrices P = K$I|0$ and P' = K'$R|t$.
4. Explain why an essential matrix has two equal singular values.
5. Using the RANSAC formula, compute how many samples the 8-point and 5-point algorithms need when only 30% of the matches are correct.

Further reading.

- R. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*, 2nd ed., Cambridge University Press, 2004. The standard reference for everything in this note.
- R. Szeliski, *Computer Vision: Algorithms and Applications*, 2nd ed., Springer, 2022. Broad coverage, including feature matching and dense stereo.
- Y. Ma, S. Soatto, J. Košecká and S. Sastry, *An Invitation to 3-D Vision*, Springer, 2004. A careful geometric treatment of the essential matrix and SfM.
- O. Faugeras, *Three-Dimensional Computer Vision: A Geometric Viewpoint*, MIT Press, 1993. The classic projective treatment.

- H. C. Longuet-Higgins, "A computer algorithm for reconstructing a scene from two projections," *Nature*, 1981. Introduces the essential matrix.
- R. Hartley, "In defense of the eight-point algorithm," *IEEE TPAMI*, 1997. Data normalisation.
- Z. Zhang, "A flexible new technique for camera calibration," *IEEE TPAMI*, 2000. Calibration from planar targets.
- D. Nistér, "An efficient solution to the five-point relative pose problem," *IEEE TPAMI*, 2004.
- M. Fischler and R. Bolles, "Random sample consensus," *Communications of the ACM*, 1981. RANSAC.$$