Binocular Stereo

Nov 6, 2026 · @Luiz Velho with Claude

Background reading for the lecture *Binocular Stereo: from correspondence to learned depth*. It follows the slides and adds the derivations, formulas and references that a 50-minute lecture leaves out.

1. The central idea

Depth from two images is a correspondence problem. Once we know which pixel in the right image shows the same scene point as a given pixel in the left image, depth follows from one line of geometry. Finding that match is the hard part, and in general it is ill-posed.

Every method in this note can be read as an answer to one question: **where does the missing information come from?** The answers, roughly in historical order, are:

- **Geometry**: the epipolar constraint reduces the search for a match from a 2D region to a 1D line (Sections 3–4).
- **Projected light**: active sensors paint texture onto the scene so that matching becomes easy (Section 2).
- **Assumed smoothness**: neighbouring pixels usually have similar depth, so evidence can be borrowed from them (Section 7).
- **Learned experience**: networks trained on millions of images learn what scenes look like, and fill in where matching is silent (Section 8).

The note is organised as follows. Sections 2–5 cover sensing and geometry: how a match becomes a depth, and how accurate that depth is. Section 6 explains why matching is hard, Section 7 covers the classical algorithms and Section 8 the learned ones. Section 9 shows how the field measures progress, and Section 10 connects machine stereo to human binocular vision.

Notation. The symbols used throughout:

Symbol	Meaning	Unit
f	focal length	pixels
B	baseline: distance between the two optical centres	metres
xL, xR	horizontal image coordinate of the same point in the left and right images	pixels
d = xL − xR	disparity	pixels
Z	depth along the optical axis	metres
K	intrinsic calibration matrix	—
R, t	rotation and translation from the left to the right camera	—
E, F	essential and fundamental matrices	—

2. The sensing landscape

Every optical depth sensor measures either an **angle** or a **time**. Triangulation needs a second viewpoint, which can be a camera, a projector or a laser, and every such second viewpoint turns depth into a correspondence problem. Time of flight needs no correspondence at all, but it needs extremely precise timing.

	Triangulation: measure an angle	Time of flight: measure a time
Variants	Passive stereo (two cameras); active stereo (cameras + dot projector); structured light (projector + camera); laser triangulation (stripe + camera)	Indirect ToF (phase of modulated light); LiDAR (timed laser pulses)
Needs correspondence?	Yes	No
Depth error vs. distance Z	Grows as Z^2 (Section 5)	Roughly constant

The timing requirement of time of flight is severe. Light travels to the target and back, so one millimetre of range corresponds to 2 mm / c ≈ 6.7 picoseconds of round-trip time.

Active sensors buy correspondence with energy. Instead of relying on whatever texture the world offers, they write a signal onto the scene: random dots (Intel RealSense), coded patterns (Kinect v1, Apple Face ID), a laser stripe, or a timed pulse (Kinect v2, LiDAR).

Sensor	How it finds correspondence	Shines at	Struggles with
Passive stereo	Searches for similar appearance	Cheap, silent, outdoors, any range	Blank, repetitive, shiny surfaces
Active stereo	Projects dots onto blank surfaces	Indoor scenes with plain walls	Sunlight drowns the dots
Structured light	Codes every projector ray	Dense, very accurate	Needs controlled light and a static scene
Time of flight / LiDAR	Needs none: it times the light	Long range, darkness, precision	Sparse or costly; multipath reflections

Both families survive because each trades something different: energy, interference, range and sunlight on one side, ambiguity on the other. **Passive stereo must infer what active sensors write onto the scene.** The rest of this note is about passive stereo, but active sensing returns in Section 9: the ground truth of the Middlebury benchmark is itself measured with structured light.

3. Two-view geometry

The lecture starts from rectified cameras. This section, which is supplementary, explains where rectified cameras come from: a general pair of calibrated cameras, and the epipolar constraint that ties their images together.

3.1 The pinhole camera

A pinhole camera maps a 3D point X = (X, Y, Z) to an image point x = (u, v) by central projection through the optical centre. In homogeneous coordinates the map is linear:

$$\lambda \begin{pmatrix} u \\ v \\ 1 \end{pmatrix} = K \, [\, R \mid t \,] \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix}, \qquad K = \begin{pmatrix} f & s & c_x \\ 0 & f & c_y \\ 0 & 0 & 1 \end{pmatrix}$$

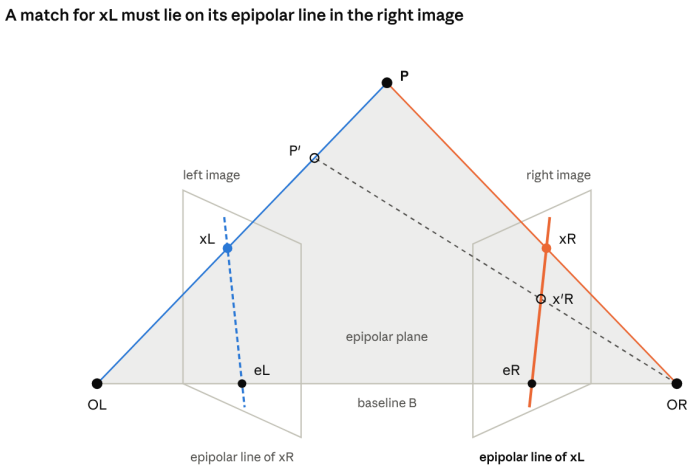
The **intrinsic** matrix K holds the focal length f in pixels, the principal point (cx, cy) and a skew s that is zero for modern sensors. The **extrinsic** parameters R and t place the camera in the world. Real lenses also add radial and tangential distortion, which calibration estimates and removes before any stereo processing.

Calibration is usually done by photographing a planar checkerboard in several poses (Zhang, 2000). For a stereo rig it returns KL, KR, the lens distortions, and the relative pose (R, t) of the right camera with respect to the left. OpenCV's `stereoCalibrate` implements this.

3.2 The epipolar constraint

Take a scene point P seen by both cameras. The two optical centres and P define a plane, the **epipolar plane**. The line joining the two centres is the **baseline**; it pierces each image plane at a point called the **epipole**.

The epipolar plane cuts each image in a line, the **epipolar line**. If P projects to xL in the left image, its match in the right image must lie on the epipolar line of xL, because the whole ray through xL projects onto that line. **The search for a match is therefore one-dimensional**, whatever the camera configuration.



Epipolar geometry · optical centres OL, OR; epipoles eL, eR

Every point on the ray through xL, such as P and P’, projects onto the same epipolar line in the right image. The baseline pierces both images at the epipoles, through which all epipolar lines pass.

3.3 Essential and fundamental matrices

Let $\hat{x} = K^{-1} x$ denote normalised image coordinates, and let a point in left-camera coordinates map to the right camera by $X_R = R X_L + t$. The vectors $\hat{x}_R$, t and $R \hat{x}_L$ are coplanar, which gives the epipolar constraint in its algebraic form:

$$\hat{x}_R^\top E \hat{x}_L = 0, \qquad E = [t]_\times R$$

Here $[t]_\times$ is the matrix of the cross product with t. The **essential matrix** E has five degrees of freedom: three for rotation and two for the direction of translation. The length of t, and so the scale of the scene, cannot be recovered from images alone.

In pixel coordinates the same constraint uses the **fundamental matrix** F, which absorbs the intrinsics:

$$x_R^\top F x_L = 0, \qquad F = K_R^{-\top} E K_L^{-1}, \qquad \ell_R = F x_L$$

F is a 3×3 matrix of rank 2 with seven degrees of freedom. The product $\ell_R = F x_L$ is the epipolar line of x_L in the right image, and the epipoles are the null vectors of F and $F^\top$.

Both matrices can be estimated from point matches alone. The normalised eight-point algorithm (Longuet-Higgins, 1981; Hartley, 1997) estimates F from eight or more matches. Nistér's five-point algorithm (2004) estimates E when K is known. In practice both run inside RANSAC to reject wrong matches.

What this buys. Once calibration gives us (R, t), the 2D search for correspondence collapses to a 1D search along epipolar lines. Rectification, in the next section, makes those lines horizontal so that the search is also easy to implement.

4. Rectified stereo: from disparity to depth

4.1 Rectification

Rectification turns a 2D search into a 1D search along image rows. It warps both images by homographies H_L and H_R that virtually rotate the two cameras, so that their image planes become coplanar and parallel to the baseline. After rectification the epipoles lie at infinity and every epipolar line is a horizontal row shared by both images.

The match of pixel (x, y) in the left image is then at $(x - d, y)$ in the right image. Finding it is a search over a single number, the **disparity** d. Every matching algorithm in Sections 6–8 assumes rectified input.

The rotation that achieves this is not unique; standard choices minimise the distortion of the warped images (Fusiello, Trucco and Verri, 2000; Bouguet's method in OpenCV's `stereoRectify`). Rectification is also why the quality of calibration matters so much. A residual vertical misalignment of even one pixel puts the true match off the row being searched.

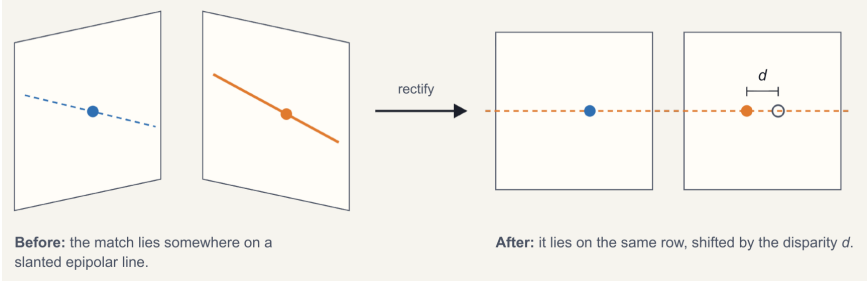


Figure 4.1 (from the lecture slides). Before rectification the match lies somewhere on a slanted epipolar line; after, it lies on the same row, shifted by the disparity d.

4.2 Depth from disparity

Place the left optical centre at the origin and the right one at (B, 0, 0), both with focal length f and image coordinates measured from the principal point. A point P = (X, Y, Z) projects to

$$x_L = \frac{fX}{Z}, \qquad x_R = \frac{f\,(X - B)}{Z}, \qquad d = x_L - x_R = \frac{fB}{Z}$$

Solving for depth gives the central equation of stereo:

$$Z = \frac{fB}{d}$$

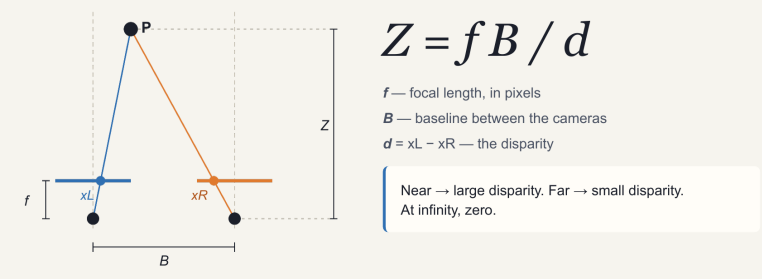


Figure 4.2 (from the lecture slides). The rectified geometry behind Z = fB/d.

Depth is inversely proportional to disparity. Near points have large disparity, far points small disparity, and points at infinity have zero disparity. The other two coordinates follow from the left image: X = xL Z / f and Y = y Z / f. OpenCV packs these three formulas into a 4 × 4 reprojection matrix Q (`reprojectImageTo3D`).

Two practical remarks:

- If the rectified principal points differ horizontally, the disparity carries a constant offset: d = xL − xR − (cx,L − cx,R).
- The nearest depth of interest fixes the **disparity search range**: dmax = f B / Zmin. With f = 700 px and B = 0.12 m, an object at 0.5 m has d = 168 px, while one at 10 m has only 8.4 px.

A quick demonstration of disparity: hold a finger near your nose and alternate eyes, then repeat at arm's length. The jump shrinks with distance. That jump is disparity, and its inverse relation to distance is the formula above.

5. Depth resolution: far away, each pixel of error costs more

Depth error grows with the square of distance. Differentiating Z = fB/d with respect to d gives the most important design equation for any stereo rig:

$$\delta Z \approx \left|\frac{\partial Z}{\partial d}\right| \delta d = \frac{fB}{d^2}\,\delta d = \frac{Z^2}{fB}\,\delta d$$

Doubling the distance quadruples the error. An equivalent form is often easier to remember: the **relative** depth error equals the relative disparity error, δZ / Z = δd / d.

Worked example. Take the rig of Section 4.2 (f = 700 px, B = 0.12 m, so fB = 84 px·m). The table gives the depth uncertainty for a one-pixel matching error and for a quarter-pixel (sub-pixel) error.

Depth Z	Disparity d	δZ at δd = 1 px	δZ at δd = 0.25 px
1 m	84 px	12 mm	3 mm
5 m	16.8 px	0.30 m	74 mm
10 m	8.4 px	1.19 m	0.30 m
20 m	4.2 px	4.76 m	1.19 m

There are three ways to buy accuracy, and each has a price:

- **A wider baseline B** cuts the error proportionally, but more of the scene is visible to only one camera (occlusion), surfaces look more different between the views (distortion), and the search range grows. Section 6 shows that all three make matching harder.
- **A longer focal length f** cuts the error too, but narrows the field of view.
- **Sub-pixel matching** reduces δd directly. Going from 1 px to 0.25 px is worth as much as a four times wider baseline, with none of its side effects. Section 7.4 shows how it is done.

The same Z^2 law explains the contrast in Section 2: triangulation sensors degrade quadratically with range, while time-of-flight error stays roughly constant. It also explains why stereo is usually evaluated in disparity rather than depth: a one-pixel disparity error means very different things at 1 m and at 20 m.

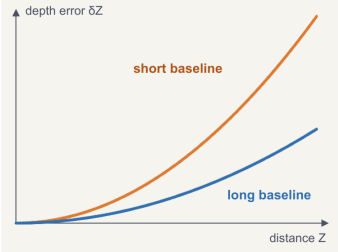


Figure 5.1 Depth error grows quadratically with distance; a longer baseline flattens the curve.

6. Why matching is hard

The difficulty of stereo is ill-posedness, not computation. Geometry has reduced the search to one number per pixel, yet that number is often not determined by the images. Matching rests on two assumptions, and every hard case breaks at least one of them:

- **Photo-consistency**: a scene point looks the same in both images.
- **Distinctiveness**: its neighbourhood identifies it uniquely along the epipolar line.

Algorithms also rely, usually implicitly, on further assumptions: Lambertian surfaces, identical cameras, visibility in both views, locally constant disparity, **uniqueness** (each pixel has at most one match), **ordering** (points keep their left-to-right order along a row) and **smoothness**.

6.1 Six ways correspondence fails

Case	What happens	Assumption broken	Typical scenes
Textureless	All disparities look alike	Distinctiveness	Walls, sky, road
Repetitive patterns	Many equally good matches	Distinctiveness	Fences, tiles, façades
Occlusion	Some pixels have no match at all	Visibility	Strips beside every foreground object
Depth edges	A window straddling a boundary mixes two depths and fattens the foreground	Local constancy	Object silhouettes
Slanted surfaces	Shapes stretch between views, so square windows do not correspond	Local constancy	Floors, roads, walls seen obliquely
Shine and glass	Highlights move with the viewer; transparency puts two depths in one pixel	Photo-consistency	Metal, water, windows

Other failure sources worth naming are thin structures, radiometric differences between the two cameras (gain, exposure, vignetting), and sensor noise in low light.

6.2 Ambiguity has a shape: the matching cost

For a single pixel, compute the dissimilarity between its neighbourhood and each candidate along the row. Plotted against candidate disparity, this **matching cost curve** shows the ambiguity directly:

- **Textured**: one sharp minimum, a confident match.
- **Textureless**: a flat curve; no disparity stands out.
- **Periodic**: several equal minima, one per period of the pattern.
- **Occluded or shiny**: the deepest minimum sits at the wrong disparity, while the true one is shallow.

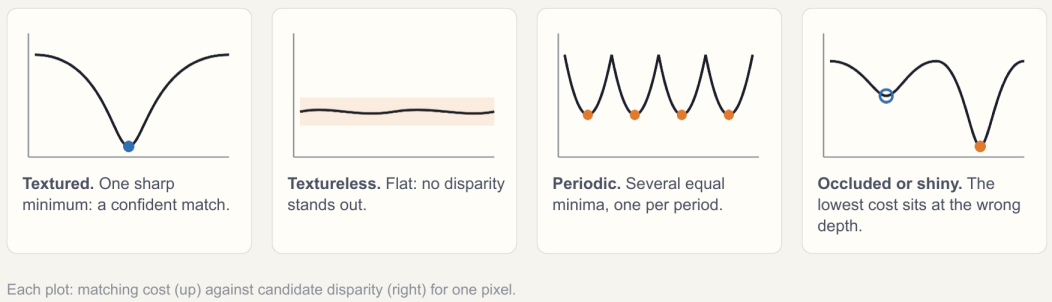


Figure 6.1 (from the lecture slides). Matching cost (vertical) against candidate disparity (horizontal) for one pixel in four situations. The hollow circle marks the true disparity.

When the cost alone is ambiguous, the answer must come from somewhere else. **Classical methods borrow evidence from neighbours** (Section 7). **Learned methods borrow it from experience** (Section 8).

7. Classical stereo matching

Every classical algorithm makes four choices (Scharstein and Szeliski, 2002):

1. **Matching cost**: how dissimilar two pixels or patches are (SAD, NCC, census).
2. **Aggregation**: summing the cost over a support window.
3. **Optimisation**: winner-take-all per pixel, or an energy over the whole image.
4. **Refinement**: sub-pixel estimation and consistency checks.

The central data structure is the **cost volume**, also called the disparity-space image: one cost C(x, y, d) for every pixel and every candidate disparity, W × H × D numbers in all. Local methods smooth it, global methods optimise over it, and, as Section 8 shows, networks learn it.

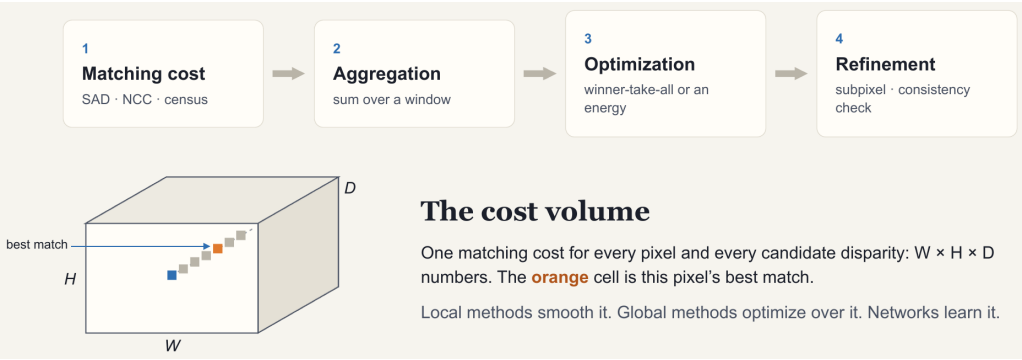


Figure 7.1 (from the lecture slides). The four choices of a classical method, and the W × H × D cost volume they operate on.

7.1 Matching costs

Let W(x, y) be a window around pixel (x, y). The classic costs are:

$$C_{\mathrm{SAD}}(x,y,d) = \sum_{(u,v)\in W(x,y)} \left| I_L(u,v) - I_R(u-d,\, v) \right|$$

$$\mathrm{NCC}(x,y,d) = \frac{\sum_W (I_L - \bar{I}_L)(I_R - \bar{I}_R)}{\sqrt{\sum_W (I_L - \bar{I}_L)^2 \, \sum_W (I_R - \bar{I}_R)^2}}, \qquad C_{\mathrm{NCC}} = 1 - \mathrm{NCC}$$

- **SAD / SSD** (sum of absolute or squared differences) are the cheapest, but any gain or exposure difference between the cameras biases them.
- **NCC** (normalised cross-correlation) subtracts the mean and divides by the contrast, so it is invariant to affine changes in brightness.
- **Census** (Zabih and Woodfill, 1994) replaces each pixel by a bit string recording which neighbours are brighter than it, and compares bit strings by Hamming distance. It survives any monotonic change of exposure and is the usual choice in practice.

Two refinements are common: the Birchfield–Tomasi cost, which is insensitive to image sampling, and mutual information, which Hirschmüller's original SGM used to cope with complex radiometric differences.

7.2 Local methods: compare windows

A local method aggregates the cost over a window and picks, for each pixel, the disparity with the lowest total (**winner-take-all**):

$$d^*(p) = \arg\min_d \; C_{\mathrm{agg}}(p,d)$$

The lesson of local methods is the **window dilemma**. A small window contains too little texture and gives noisy matches; a large window straddles depth edges and blurs them. The fixes make the window adapt to the image:

- **Adaptive support weights** (Yoon and Kweon, 2006) weight each neighbour by its colour similarity and distance to the centre.
- **Cost-volume filtering** with an edge-preserving guided filter (Rhemann et al., 2011) achieves the same effect in constant time per pixel.
- **PatchMatch Stereo** (Bleyer, Rhemann and Rother, 2011) gives each pixel a slanted plane instead of a constant disparity, which fixes the slanted-surface case of Section 6.1.

Local methods are fast and simple. Block matching runs inside many depth cameras.

7.3 Global methods: minimise an energy

Global methods treat stereo as a labelling problem on the pixel grid. They seek the disparity map d that minimises

$$E(d) = \sum_{p} C(p, d_p) \; + \; \lambda \sum_{(p,q)\in\mathcal{N}} V(d_p, d_q)$$

The **data term** asks each pixel to match well. The **smoothness term** asks neighbours p, q to agree, but must allow jumps at depth edges, so V is a truncated or Potts-type penalty rather than a quadratic one. With such an edge-preserving V, exact minimisation on a 2D grid is NP-hard, and the solvers differ in how they approximate it.

Solver	What it optimises	Strength	Weakness
Dynamic programming (Ohta and Kanade, 1985)	Exact along each row; rows solved independently	Fast; can model occlusions	Rows disagree, giving horizontal streaks
Semi-global matching (Hirschmüller, 2005, 2008)	1D optimisation along 8–16 directions, summed per pixel	Fast, parallel, runs on FPGAs and GPUs	Approximate; some directional artefacts
Graph cuts, belief propagation (Boykov, Veksler and Zabih, 2001; Sun, Zheng and Shum, 2003)	The full 2D field, every pixel tied to its neighbours	Best quality before learning	Slow, memory-hungry

Semi-global matching became the industry default and still ships in cars and aerial mapping. Along each direction r it runs the recursion

$$L_r(p,d) = C(p,d) + \min\Big(L_r(p-r,d),\; L_r(p-r,d\pm1) + P_1,\; \min_k L_r(p-r,k) + P_2\Big) - \min_k L_r(p-r,k)$$

and sums the paths: S(p, d) = Σr Lr(p, d), summed over the directions r. The small penalty P_1 allows gentle slopes, the large penalty P_2 allows true depth jumps, and the final subtraction keeps the values bounded. Winner-take-all on S gives the disparity.

A turning point. Tappen and Freeman (2003) and Szeliski et al. (2008) found that the best solvers reached energies *lower* than the energy of the ground-truth disparity map. The optimiser was no longer the bottleneck; the energy model itself was wrong. That observation opened the door to learning the model from data.

7.4 Refinement

- **Sub-pixel disparity.** Fit a parabola through the costs at the best integer disparity d and its two neighbours, and take its minimum:

$$d_{\text{sub}} = d + \frac{C(d-1) - C(d+1)}{2\,(C(d-1) - 2\,C(d) + C(d+1))}$$

- **Left–right consistency check.** Compute disparity maps with both images as reference. A pixel is kept only if the two agree, $|d_L(x) - d_R(x - d_L(x))| \le 1$. The pixels that fail are mostly occlusions and mismatches.
- **Post-processing.** A median filter removes isolated outliers, and the holes left by the consistency check are filled from the background side of the nearest valid neighbours.

The consistency check matters beyond tidiness. It makes a classical method **report where it failed**, a point Section 8.4 returns to.

8. Learned stereo and monocular depth

Learning kept the classical pipeline and learned its parts, one at a time. Each generation of networks replaced one hand-designed component of Section 7. The fuel was synthetic data with perfect depth, from SceneFlow (Mayer et al., 2016) to million-pair datasets in 2025.

8.1 Five generations of learned stereo

Years	Generation	What it replaced	Representative methods
2025–	Monocular priors	The smoothness prior	FoundationStereo, DEFOM-Stereo, MonSter, Stereo Anywhere
2021–25	Global attention	Window-based comparison; resolves repeats with context	STTR → S^2M^2
2021–24	Iterative refinement	Exhaustive search over the cost volume	RAFT-Stereo, IGEV-Stereo
2016–19	End-to-end cost volumes	The MRF solver	DispNet, GC-Net, PSMNet
2015	Learned matching costs	Hand-made costs (SAD, census)	MC-CNN + SGM

- **Learned costs.** MC-CNN (Žbontar and LeCun, 2015) trained a siamese network to say whether two patches match, and kept SGM for optimisation.
- **End-to-end cost volumes.** GC-Net (Kendall et al., 2017) and PSMNet (Chang and Chen, 2018) build a volume of feature vectors rather than scalar costs, regularise it with 3D convolutions, and read out disparity with a differentiable **soft-argmin**:

$$\hat{d}(p) = \sum_{d=0}^{D-1} d \cdot \sigma(-c_d(p)), \qquad \sigma = \text{softmax over } d$$

- **Iterative refinement.** RAFT-Stereo (Lipson, Teed and Deng, 2021) and IGEV-Stereo (Xu et al., 2023) build an all-pairs correlation pyramid once, then update a disparity estimate with a recurrent unit that looks up costs near the current guess. This works like a learned PatchMatch and avoids searching every disparity.
- **Global attention.** Transformers compare a pixel with every candidate on its row, so context can disambiguate repetitive patterns. STTR (Li et al., 2021) started this line; S²M² (ICCV 2025) made it scale and also predicts occlusion and confidence.
- **Monocular priors.** Since 2025, FoundationStereo, DEFOM-Stereo, MonSter and Stereo Anywhere import features or depth from a monocular foundation model such as Depth Anything. Where matching is silent, the monocular prior fills the gap: **it is the learned heir of the smoothness term**. These models generalise zero-shot to new domains, and 2026 work (Fast-FoundationStereo, Lite Any Stereo) makes them real-time.

8.2 One eye: depth by recognition, not measurement

A single image carries no geometric depth measurement, yet it carries many statistical **depth cues**: perspective, texture gradient, occlusion, familiar size, shading, defocus, atmospheric haze and height in the image. Monocular depth is recognition, not measurement. Networks learn these regularities from data.

They predict **shape** well, but **absolute scale is unobservable**. Scale the scene and the camera's distance by the same factor and the image does not change: a near, small object and a far, large one fill the same rays. Many models therefore predict depth (or inverse depth) only up to an unknown scale s and shift t. Aligning such a prediction $\hat{z}$ to a reference z is a small least-squares problem:

$$(s^*, t^*) = \arg\min_{s,t} \sum_p \left(s\,\hat{z}_p + t - z_p\right)^2$$

Stereo networks that borrow monocular depth must estimate this scale and shift from the stereo evidence, which is exactly what binocular geometry can supply.

Years	Step	Idea	Representative methods
2024–26	Metric and any-view	Recover scale by also predicting the camera; accept one image or many	UniDepthV2, Depth Pro, MoGe-2; DUSt3R, VGGT, Depth Anything 3
2020–24	Relative depth at scale	Mix many datasets with a scale- and shift-invariant loss	MiDaS, Depth Anything, Marigold
2016–19	Self-supervised from stereo	Learn by reconstructing one view from the other	Monodepth, SfMLearner
2014	Supervised networks	Regress depth from labelled images	Eigen, Puhrsch and Fergus

Two details deserve attention. Monodepth (Godard et al., 2017) needed no depth labels: it learned by warping the right image into the left, so **binocular geometry was the teacher**. Today the arrow also points back, as stereo networks borrow monocular priors.

Models such as DUSt3R, VGGT and Depth Anything 3 take any number of views, blurring the line between monocular, binocular and multi-view reconstruction.

8.3 The synthesis: where does the missing information come from?

The data term goes silent in the same regions for every method: textureless, repetitive, occluded and shiny surfaces. Each family fills those regions from a different source.

Source	Principle	Where it lives	Era
Assumed smoothness	Neighbours agree	MRFs, SGM	1990s–2010s
Projected light	Write texture onto the scene	Active sensors	Throughout
Learned context	Context resolves repeats	Networks	2016–
Monocular priors	What scenes look like	Foundation models	2025–

Stereo measures where texture exists; priors guess where it does not. The difference that matters is measurement versus guessing.

8.4 A caution: knowing when you don't know

A classical method knows when it doesn't know. When the cost curve is flat, the left–right check fails and a hole appears in the depth map. A network trained to fill every pixel instead produces a confident, smooth, plausible answer precisely where nothing was measured.

	Classical failure	Learned failure
Looks like	Holes, noise, failed left–right checks	Plausible, smooth, wrong depth
Visible?	Yes	No

Recent models (S^2M^2, UniDepthV2) output confidence or occlusion maps; that is the field's partial answer. When you use a learned depth model, ask for these maps and evaluate it specifically on the regions where matching is ill-posed. For a robot, a hole you can see is often safer than a surface that isn't there.

9. Benchmarks: making stereo measurable

Middlebury turned stereo from demonstrations into measurement. Scharstein and Szeliski's 2002 paper did three things at once: it proposed the taxonomy of Section 7, released shared images with ground-truth disparity, and opened a public results table. A depth map is useful only if we can test it, and a pretty depth map is not proof of geometric accuracy.

9.1 The Middlebury datasets

Year	Release	Ground truth
2021	24 scenes captured by a phone on a robot arm	Structured light
2014	33 scenes at 5–6 MP with sub-pixel accuracy (version 3)	Structured light
2005–06	30 scenes	Structured light
2003	2 scenes	Structured light
2001	6 piecewise-planar scenes	Hand-fitted planes

Since 2003 the ground truth has come from projecting coded patterns onto the scene: the structured light of Section 2, used as a measuring instrument. **Active sensing made passive stereo measurable.**

Version 3 (2014) deliberately keeps real-world difficulties: high resolution with disparities up to 800 px, imperfect rectification, and variants with different exposure and lighting. It has 15 training and 15 test pairs, and each method gets a single submission on the test set, which limits tuning to the benchmark. Well-known scenes include Adirondack, Motorcycle and Piano. Leaderboards change quickly, so check the live table at vision.middlebury.edu/stereo rather than relying on any ranking quoted in print.

9.2 Metrics

Three questions are asked of every disparity map:

1. **How far off is it?** The average absolute disparity error (also called end-point error, EPE).
2. **How often is it wrong?** The bad-pixel rate: the fraction of pixels whose error exceeds a threshold τ.
3. **Where are the errors?** Scores computed under masks, for example over non-occluded pixels only, or over all pixels.

$$\mathrm{EPE} = \frac{1}{|M|}\sum_{p\in M}|d_p - d_p^{\mathrm{gt}}|, \qquad \mathrm{bad}\,\tau = \frac{100}{|M|}\sum_{p\in M}\mathbf{1}\big[\,|d_p - d_p^{\mathrm{gt}}| > \tau\,\big]$$

Here M is the evaluation mask. Middlebury v3 ranks by default on **bad 2.0**: the weighted percentage of non-occluded pixels with error above 2 px. Averages hide the interesting failures, so also inspect thin objects and depth boundaries, where errors concentrate.

9.3 Other benchmarks

- **KITTI** (2012, 2015): driving scenes with ground truth from a LiDAR scanner, sparse and limited to static surfaces. Its D1 metric counts a pixel as an outlier when its error exceeds both 3 px and 5% of the true disparity.
- **ETH3D** (2017): indoor and outdoor scenes with ground truth from a high-precision laser scanner, including a low-resolution two-view stereo track.

- **Zero-shot evaluation**: recent foundation-style models are trained on synthetic data and tested on these benchmarks without fine-tuning, so that generalisation, not benchmark tuning, is what gets measured.

10. Binocular vision in biology, and what it suggests

Human stereo is passive in light but active in control. The eyes do not illuminate the scene; they move. Almost every method in Sections 7–8 assumes fixed, rectified cameras, and this section asks what changes when the cameras can choose where to look.

10.1 Where machine stereo started

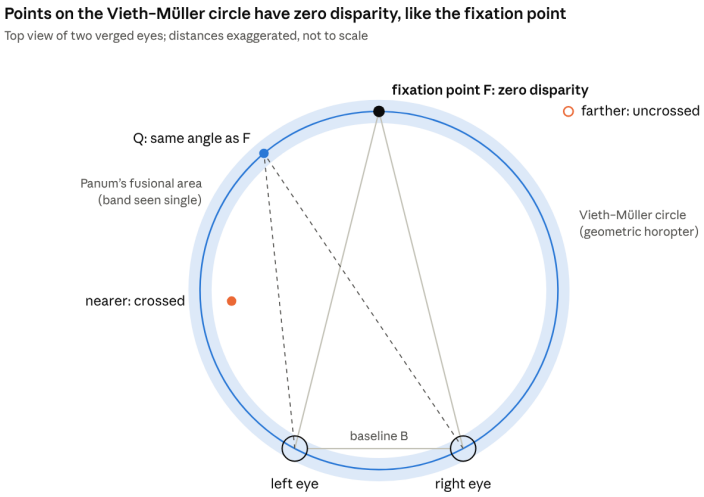
The field's usual starting point is Marr and Poggio's cooperative algorithm (1976). It took its constraints from human vision: **compatibility** (matched features look alike), **uniqueness** (each feature has one match) and **continuity** (disparity varies smoothly almost everywhere). These are the photo-consistency, uniqueness and smoothness assumptions of Section 6.

Julesz's random-dot stereograms (1960) had already shown that the visual system matches the two images *before* recognising objects. Each image alone is pure noise, yet a shape in depth appears when they are fused. Stereopsis is a low-level correspondence process, which is what made it a natural target for computation.

10.2 Vergence, the horopter and Panum's area

The two eyes are a stereo rig with a baseline of about 6.3 cm (the average adult interpupillary distance). Unlike a rectified camera pair, they **verge**: both rotate so that the point of interest falls on both foveas. The fixated point then has zero disparity by construction.

The set of points that also have zero disparity is the **horopter**. Geometrically, it is the Vieth–Müller circle through the fixation point and the optical centres of the two eyes: every point on that circle subtends the same angle at the two eyes. The empirical horopter measured in observers is flatter than this circle.



Vergence geometry · horopter, Panum's area, crossed and uncrossed disparity

Q sits on the circle and subtends the same angle at the two eyes as F, so its disparity is zero too. Points inside the circle are nearer and have crossed disparity; points outside are farther and have uncrossed disparity.

- Points **nearer** than the horopter have **crossed** disparity; points **farther** have **uncrossed** disparity.
- For a point near the midline at distance ZP, with fixation at ZF, the angular disparity is approximately $\eta \approx B / ZP - B / ZF$. For a small depth difference ΔZ this becomes $\eta \approx B \, \Delta Z / Z^2$: the angular form of the Z^2 law of Section 5.
- Only a thin band around the horopter, **Panum's fusional area**, is seen single. It is narrowest at the fovea (of the order of a few to a few tens of arcminutes) and widens with eccentricity. Beyond it, points are seen double (diplopia), although their depth can still be judged coarsely.

Human stereoacuity is remarkable. Good observers detect disparities of the order of 10–20 arcseconds, a small fraction of the spacing between foveal cones. With $\eta = 20$ arcsec $\approx 9.7 \times 10^{-5}$ rad and $B = 6.3$ cm, the formula gives $\Delta Z \approx Z^2 \, \eta / B \approx 1.5$ mm at $Z = 1$ m.

10.3 Foveation and active stereo

Vergence and foveation work together. Resolution is high only in the central degree or two of the visual field, which is also where disparity sensitivity is best and Panum's area is smallest. By verging on what it looks at, the visual system keeps disparities small exactly where it needs precision. The search range shrinks, matching becomes easier, and the coarse periphery only has to guide the next fixation.

In computer vision this idea has a long history under the name **active** or **animate vision** (Aloimonos, Weiss and Bandyopadhyay, 1988; Bajcsy, 1988; Ballard, 1991): an observer that controls its viewpoint can turn ill-posed problems into well-posed ones. Verged camera pairs can still be rectified, but the rectifying homographies change with every fixation, and foveated or log-polar sensors break the uniform pixel grid that cost volumes assume.

A research question to keep in mind: what changes if the cameras can choose where to look, so as to make matching easier? Note that this is a second meaning of *active*. Active sensors in Section 2 add energy to the scene; active observers add control.

10.4 Fifty years of stereo

Year	Era	Key ideas
2025	Measurement + priors	Stereo meets monocular and multi-view foundation models
2015	Learning	Learned costs, cost volumes, iterative refinement
2002	Energy and benchmarks	Graph cuts, SGM, Middlebury
1976	Correspondence	Marr and Poggio: constraints from human vision

11. Summary

- Depth from two images is a correspondence problem. Geometry turns a match into a depth; the hard part is the match.
- Calibration and the epipolar constraint reduce the search to a line; rectification makes that line an image row.
- In a rectified rig, $Z = fB/d$, and the depth error grows as $Z^2\,\delta d / fB$. Baseline, focal length and sub-pixel accuracy are the three levers.
- Matching fails where photo-consistency or distinctiveness fail: textureless, repetitive, occluded, slanted, shiny surfaces and depth edges.
- Classical methods combine a cost, aggregation, an optimiser and refinement over a cost volume; SGM is the practical standard.
- Learned methods kept that pipeline and learned its parts; the newest ones fill silent regions with monocular priors.
- Measurement and guessing look the same in a depth map. Ask for confidence and occlusion maps, and evaluate where matching is ill-posed.
- Biological stereo is active in control: vergence and foveation keep disparities small where precision is needed.

12. Exercises

1. **Design.** A rig has $f = 1000$ px and $B = 0.20$ m, and matches to $\delta d = 0.5$ px. Beyond what distance does the depth error exceed 10 cm? (Answer: about 6.3 m.) What baseline would push that distance to 20 m?
2. **Epipolar geometry.** For two identical cameras with $R = I$ and the right centre at $(B, 0, 0)$, compute $E = [t]_\times R$ with $t = (-B, 0, 0)$. Show that the constraint $\hat{x}_R^T E \hat{x}_L = 0$ reduces to $y_L = y_R$.
3. **Window dilemma.** Implement SAD and census block matching with winner-take-all on the Middlebury Motorcycle pair (available as `skimage.data.stereo_motorcycle()`). Plot bad 2.0 against window size from 3 to 21 px, and describe where each window size fails.
4. **Refinement.** Add the left–right consistency check and parabolic sub-pixel interpolation to your matcher. Report how EPE and bad 2.0 change, and visualise the pixels the check rejects.
5. **Semi-global matching.** Run OpenCV's `StereoSGBM` on the same pair. Vary P_1 and P_2 and explain the effect on slanted surfaces and depth edges.
6. **Learned stereo.** Run a pretrained network (for example RAFT-Stereo or FoundationStereo) on the same pair and on a pair you capture yourself. Compare it with SGM in textureless and occluded regions. Where does it produce plausible but wrong depth?
7. **Vergence.** You fixate a point 50 cm away. Compute the angular disparity of a point 10 cm behind it, with $B = 6.3$ cm. Is it crossed or uncrossed? Is it likely to fall inside Panum's area?

13. Further reading

Textbooks

- R. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*, 2nd ed., Cambridge University Press, 2004. Chapters 9–12 on epipolar geometry, F and triangulation.
- R. Szeliski, *Computer Vision: Algorithms and Applications*, 2nd ed., Springer, 2022. The chapter on depth estimation covers rectification and dense stereo.
- I. P. Howard and B. J. Rogers, *Perceiving in Depth*, Oxford University Press, 2012. The reference on biological binocular vision.

Classical stereo

- D. Marr and T. Poggio, "Cooperative computation of stereo disparity", *Science*, 1976.
- D. Scharstein and R. Szeliski, "A taxonomy and evaluation of dense two-frame stereo correspondence algorithms", *IJCV*, 2002.
- H. Hirschmüller, "Stereo processing by semiglobal matching and mutual information", *IEEE TPAMI*, 2008.
- Y. Boykov, O. Veksler and R. Zabih, "Fast approximate energy minimization via graph cuts", *IEEE TPAMI*, 2001.
- M. Bleyer, C. Rhemann and C. Rother, "PatchMatch Stereo", *BMVC*, 2011.

Learned stereo and monocular depth

- J. Žbontar and Y. LeCun, "Stereo matching by training a convolutional neural network to compare image patches", *JMLR*, 2016.
- A. Kendall et al., "End-to-end learning of geometry and context for deep stereo regression" (GC-Net), *ICCV*, 2017.
- L. Lipson, Z. Teed and J. Deng, "RAFT-Stereo: multilevel recurrent field transforms for stereo matching", *3DV*, 2021.
- C. Godard, O. Mac Aodha and G. Brostow, "Unsupervised monocular depth estimation with left-right consistency" (Monodepth), *CVPR*, 2017.
- R. Ranftl et al., "Towards robust monocular depth estimation: mixing datasets for zero-shot cross-dataset transfer" (MiDaS), *IEEE TPAMI*, 2020.
- B. Wen et al., "FoundationStereo: zero-shot stereo matching", *CVPR*, 2025.

Benchmarks and data

- Middlebury Stereo Vision page · 2014 datasets · Evaluation v3 · Submission rules
- scikit-image `stereo_motorcycle` sample

The content of Sections 1–2 and 6–10 follows the lecture slides *Binocular Stereo: from correspondence to learned depth* (L. Velho / ChatGPT). Derivations, formulas, worked examples and exercises were added for this note.