Stereo Matching

Sangeon Jeon

April 2025

1 Problem Description

In many computer vision applications, understanding the three-dimensional structure of a scene from images is essential. A common approach to achieve this is to use two cameras placed side by side (a stereo pair). When the same point in a scene is viewed from these two slightly different horizontal positions, it appears shifted horizontally between the two images. This horizontal shift, called the disparity, directly encodes information about the distance of that point from the cameras.

Our goal in this work is to reconstruct the three-dimensional shape of a scene from a pair of stereo images. By determining how far each point in the left image needs to be shifted horizontally to align with its corresponding point in the right image, we generate a disparity map—an image where each pixel value represents this horizontal shift.

In solving this problem, we are guided by two competing objectives:

- **Data Fidelity:** After shifting one image according to the estimated disparity, corresponding points in the two images should appear visually similar. Significant differences in color or intensity after shifting indicate incorrect matches. Thus, minimizing these differences ensures accurate correspondence.
- **Smoothness:** Nearby pixels generally belong to the same surface, and therefore, should have similar disparity values, except at depth discontinuities or edges. This smoothness constraint helps prevent erroneous disparities, especially in regions that lack distinctive textures or colors, where matching could otherwise become ambiguous.

These two objectives typically complement each other within uniform regions of a surface. However, near object boundaries, they may conflict. The smoothness term encourages uniform disparities across the boundary, potentially blurring sharp edges, while the data fidelity term seeks to preserve these abrupt transitions. Balancing these objectives depends heavily on the specifics of the scene and application.

Real-world stereo matching presents additional challenges such as occlusions (regions visible in one image but hidden in the other), textureless areas leading to ambiguous matches, and differences in lighting or image noise. In this preliminary implementation, however, we simplify the scenario by assuming there are no occlusions and minimal noise, allowing us to focus on the core disparity estimation process.

2 Mathematical Formulation

This project explores four energy-based stereo matching strategies—pixel vs. patch matching, and brightness vs. color matching. We begin with the simplest case, pixel-based brightness matching, and then extend the formulation to the other three. Our algorithm is a variational gradient-descent method, which iteratively adjusts the disparity map to minimize a carefully defined energy functional. In the following sections, we will show how we construct that energy so that its minimization naturally fullfills our objectives. Let's dive into the pixel-brightness matching formulation first.

Pixel Brightness Matching

Let $\Omega \subset R^2$ represent the common image domain. We denote the left and right grayscale images by

$$I_L,\, I_R : \Omega \to [0,1],$$

where each pixel has a brightness value. We seek a disparity field

$$d : \Omega \to [0, d_{\max}],$$

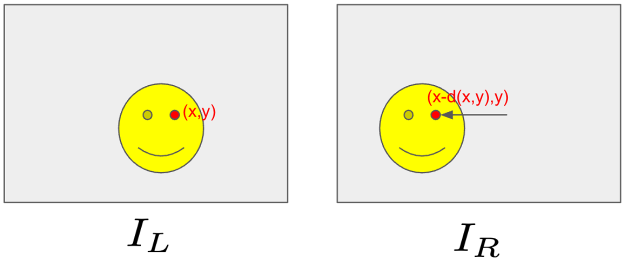
such that horizontally shifting the right image by $d(x,y)$ aligns it with the left image at each pixel $(x,y) \in \Omega$.

Here, we define an energy functional as:

$$E(d) = \underbrace{\frac{\lambda}{2} \int_\Omega \Big(I_L(x,y) - I_R\big(x - d(x,y), y\big) \Big)^2 \, dx\, dy}_{E_{\text{data}}(d)} + \underbrace{\frac{1-\lambda}{2} \int_\Omega \|\nabla d\|^2 \, dx\, dy}_{E_{\text{smooth}}(d)} .$$

The first term, $E_{\text{data}}(d)$, is known as the *data term* because it measures how well the left image aligns with the horizontally shifted right image. Since the right image is taken from a viewpoint further to the right, objects in the scene appear shifted to the left relative to their positions in the left image. Accordingly, we evaluate the right image at the shifted coordinate $(x - d(x,y), y)$, where $d(x,y)$ is expected to be nonnegative.

Under the assumption that the right image is simply a shifted version of the left image, an accurate disparity map should align the two images perfectly. The data term quantifies this alignment by integrating the squared intensity differences between the left image and the warped right image. Because it is a sum of squared differences, this term is always nonnegative and equals zero if and only if the images match exactly after warping. Any discrepancy between the two images contributes positively to this term, reflecting mismatches in correspondence.



The second term $E_{\text{smooth}}(d)$ is the *smoothness term* which measures the spatial smoothness of the disparity map by integrating the squared gradient over the entire domain. This term also integrates nonnegative values, and it achieves a value of zero only if the disparity is constant across the region. Any variation or roughness in the disparity field increases the value of this term.

By combining these two terms, our energy functional simultaneously represents the objectives of data fidelity and spatial smoothness. The parameter $\lambda \in [0,1]$ balances these competing objectives, providing flexibility to adapt to different image characteristics. However, since every pixel in the right image is shifted to the left, certain pixels on the left edge might be shifted outside of the image domain. To reflect this, we introduce a selection function **Sel** into the energy formulation:

$$E(d) = \frac{\lambda}{2} \int_\Omega \mathbf{Sel}(x - d(x,y)) \Big(I_L(x,y) - I_R\big(x - d(x,y), y\big) \Big)^2 \, dx\, dy \; + \; \frac{1-\lambda}{2} \int_\Omega \|\nabla d\|^2 \, dx\, dy,$$

where the selection function is defined as $\mathbf{Sel}(z) = 1$ if $0 \le z \le w$ and 0 otherwise. With this selection function, the data term only includes points that remain within the valid image domain after warping.

Color Matching

Next, we extend this formulation to incorporate color information, potentially yielding a more precise disparity map. A color image contains three RGB channels:

$$I_L,\, I_R : \Omega \to [0,1]^3.$$

Considering that brightness compresses the rich RGB information into a single scalar value, it may potentially neglect the detailed color structure of the images. Therefore, we modify the energy functional to explicitly account for the color channels (i.e. $c \in \{R, G, B\}$):

$$E(d) = \frac{\lambda}{2} \int_\Omega \frac{1}{3}\mathbf{Sel}(x - d(x,y)) \sum_{c\in\{R,G,B\}} \Big(I_L^c(x,y) - I_R^c\big(x - d(x,y), y\big)\Big)^2 \,dx\,dy \;+\; \frac{1-\lambda}{2} \int_\Omega \|\nabla d\|^2 \,dx\,dy.$$

Note that only the data term is modified since the smoothness term depends solely on the disparity map d. Here, the algorithm computes the squared difference for each color channel separately and averages these differences. The averaging ensures that the color-based energy term gradient is directly comparable to that of brightness-based one.

Patch Matching

Finally, let us incorporate the patch-matching strategy into our framework. Patch matching involves comparing the brightness or color of neighboring pixels around a given point in both images. Rather than comparing individual pixels, it compares small regions (patches) from the left and right images. This approach enhances the robustness of the algorithm, particularly in relatively textureless regions. Although a single incorrect pixel in the right image might coincidentally match the brightness or color of a pixel in the left image, it is highly unlikely that an entire patch of pixels will incorrectly match.

$$E(d) = \frac{\lambda}{2} \int_\Omega \frac{1}{3}\,\mathbf{Sel}\big(x - d(x,y)\big) \sum_{c\in\{R,G,B\}} \frac{1}{\displaystyle\int_{N_{x,y}} dx'\,dy'} \left[\int_{N_{x,y}} \big(I_L^c(x',y') - I_R^c(x' - d(x,y),\, y')\big)^2 \,dx'\,dy'\right] dx\,dy$$

$$+\frac{1-\lambda}{2} \int_\Omega \|\nabla d\|^2 \,dx\,dy.$$

Here, we compute the data fidelity by summing the squared mismatches within a local neighborhood. The normalization factor $\frac{1}{\int N_{x,y}(x',y')\,dx'dy'}$ ensures that the patch-based data term is comparable in scale to the pixel-based data term. To clarify the patch integral, we express the neighborhood function explicitly:

$$\int_{N_{x,y}} \left(I_L^c(x',y') - I_R^c\left(x' - d(x,y), y'\right)\right)^2 \,dx'dy' = \int_\Omega N_{x,y}(x',y') \left(I_L^c(x',y') - I_R^c\left(x' - d(x,y), y'\right)\right)^2 \,dx'dy'.$$

By defining the kernel K such that $N_{x,y}(x',y') = K(x - x', y - y')$, and letting

$$M^c(x',y') = \left(I_L^c(x',y') - I_R^c\left(x' - d(x,y), y'\right)\right)^2,$$

the expression simplifies into a convolution:

$$\int_{N_{x,y}} \left(I_L^c(x',y') - I_R^c\left(x' - d(x,y), y'\right)\right)^2 \,dx'dy' = \int_\Omega K(x - x', y - y')\, M^c(x',y')\,dx'dy' = (K * M^c)(x,y).$$

Letting the kernel weight be $W_K = \int K(x,y)\,dx\,dy$, the full energy functional becomes:

$$E(d) = \frac{\lambda}{2} \int_\Omega \frac{1}{3}\,\mathbf{Sel}(x - d(x,y)) \sum_{c\in\{R,G,B\}} \frac{1}{W_K}(K * M^c)(x,y)\,dx\,dy \;+\; \frac{1-\lambda}{2} \int_\Omega \|\nabla d\|^2 \,dx\,dy.$$

In this section, we have seen how our objectives—data fidelity and smoothness—can be encoded into an energy functional. In the next section, we will derive the variational gradient descent PDE used to minimize this energy.

3 PDE Derivation

To minimize $E(d)$, we apply variational gradient descent. We begin with the simplest case—pixel-based brightness matching as well—before extending the derivation to color and patch-based formulations.

Pixel Brightness Matching

We consider the energy functional:

$$E(d) = \frac{\lambda}{2} \int_{\Omega} \mathbf{Sel}(x - d(x,y)) \left(I_L(x,y) - I_R\left(x - d(x,y), y\right)\right)^2 \, dx\, dy \;+\; \frac{1-\lambda}{2} \int_{\Omega} \|\nabla d\|^2 \, dx\, dy.$$

To derive the corresponding PDE, we compute the variational derivative(i.e. Eular-Lagrangian Equation):

$$\frac{\delta E}{\delta d} = \frac{\partial E}{\partial d} - \frac{\partial}{\partial x}\left(\frac{\partial E}{\partial d_x}\right) - \frac{\partial}{\partial y}\left(\frac{\partial E}{\partial d_y}\right).$$

Before proceeding, let us examine the role of the selection function **Sel**. This function is introduced for implementation convenience—it is not a formal part of the energy minimization objective. If it were, one could trivially minimize the energy by shifting pixels far enough outside the image so that $\mathbf{Sel} = 0$, effectively excluding those terms. However, this is undesirable. We want the algorithm to match as many pixels as possible, only excluding those near the boundaries where warping goes out of range.

To make this assumption reasonable, we hope that the disparity updates are monotonic and positive, such that a point does not overshoot and re-enter the domain later. In practice, though, disparities fluctuate during optimization, and some points may need to move in and out of valid regions. Thus, we do not treat **Sel** as part of the true energy, but instead apply it at the end of differentiation for implementation purposes.

The variational derivative of the data term becomes:

$$E_d = \lambda \int_{\Omega} \left(I_L(x,y) - I_R\left(x - d(x,y), y\right)\right) \frac{\partial I_R}{\partial x}\left(x - d(x,y), y\right) \, dx\, dy.$$

The smoothness term yields:

$$E_{d_x} = (1-\lambda) \int_{\Omega} d_{xx} \, dx\, dy, \quad E_{d_y} = (1-\lambda) \int_{\Omega} d_{yy} \, dx\, dy.$$

Combining these, the gradient flow PDE is:

$$\frac{\partial d}{\partial t} = -\frac{\delta E}{\delta d} = -\lambda \int_{\Omega} \left(I_L(x,y) - I_R\left(x - d(x,y), y\right)\right) \frac{\partial I_R}{\partial x}\left(x - d(x,y), y\right) \, dx\, dy + (1-\lambda) \int_{\Omega} \Delta d \, dx\, dy.$$

We then apply $\mathbf{Sel}(x - d(x,y))$ to the data term gradient for numerical stability:

$$\frac{\partial d}{\partial t} = -\lambda \int_{\Omega} \mathbf{Sel}(x - d(x,y)) \left(I_L(x,y) - I_R\left(x - d(x,y), y\right)\right) \frac{\partial I_R}{\partial x}\left(x - d(x,y), y\right) \, dx\, dy + (1-\lambda) \int_{\Omega} \Delta d \, dx\, dy.$$

Color and Patch Matching Extensions

To incorporate color matching, we extend the data term to average over RGB channels:

$$\frac{\partial d}{\partial t} = -\lambda \int_{\Omega} \frac{1}{3} \mathbf{Sel}(x - d(x,y)) \sum_{c \in \{R,G,B\}} \left(I_L^c(x,y) - I_R^c(x - d(x,y), y)\right) \frac{\partial I_R^c}{\partial x}(x - d(x,y), y) \, dx\, dy + (1-\lambda) \int_{\Omega} \Delta d \, dx\, dy.$$

To further incorporate patch matching, we consider the patch-based convolutional formulation:

$$\frac{\partial d}{\partial t} = -\lambda \int_{\Omega} \frac{1}{3} \mathbf{Sel}(x - d(x,y)) \sum_{c \in \{R,G,B\}} \frac{1}{W_K} \int_{N_{x,y}} \left(I_L^c(x',y') - I_R^c(x' - d(x',y'), y')\right) \frac{\partial I_R^c}{\partial x}(x' - d(x,y), y') \, dx'\, dy'$$

$$+(1-\lambda)\int_\Omega \Delta d\, dx\, dy.$$

Letting

$$M_d^c(x',y') = 2\left(I_L^c(x',y') - I_R^c(x'-d(x',y'),y')\right)\frac{\partial I_R^c}{\partial x}(x'-d(x,y),y'),$$

we can simplify the patch-based PDE using convolution as:

$$\frac{\partial d}{\partial t} = -\frac{\lambda}{2}\int_\Omega \frac{1}{3}\mathbf{Sel}(x-d(x,y))\sum_{c\in\{R,G,B\}}\frac{1}{W_K}(K * M_d^c)(x,y)\, dx\, dy + (1-\lambda)\int_\Omega \Delta d\, dx\, dy.$$

This PDE evolves the disparity map d over time toward a local minimizer of the energy functional, balancing color-consistent patch matching against smoothness.

4 Discretization and Implementation

Having derived a variational gradient–descent PDE for estimating the disparity map d, we now discuss its discretization and implementation. Below we summarize our choices for spatial discretization, interpolation, boundary conditions, and time-stepping.

In our framework, we must approximate the partial derivative of the right image I_R with respect to x. We use a backward-difference scheme:

$$\frac{\partial I_R}{\partial x}(i,j) \;\approx\; I_R(i,j) \;-\; I_R(i,j-1),$$

replicating the first column. Since our disparity updates $d(x,y)$ are nonnegative, a backward difference aligns with the warp direction (right→left) and better captures intensity changes along that axis.

At each iteration, we evaluate

$$I_R\big(x-d(x,y),\,y\big) \quad and \quad \frac{\partial I_R}{\partial x}\big(x-d(x,y),\,y\big).$$

Because $x-d(x,y)$ is generally noninteger, we apply linear interpolation on the discrete grids of both I_R and $\partial_x I_R$, enabling subpixel-accurate warping.

In addition, we enforce

$$d(x,y) \;\geq\; 0$$

at every iteration, since negative disparities are not physically meaningful for rectified stereo images.

Time-Stepping: Δt

Because this PDE is updated explicitly, stability enforces a maximum Δt. We address two aspects:

(1) Diffusion-only CFL Constraint. Suppose we temporarily ignore the data term and consider only the smoothness (diffusion) part of our PDE:

$$\frac{\partial d}{\partial t} \;=\; (1-\lambda)\,\Delta d.$$

We discretize in space with a five-point stencil on a grid where $\Delta x = \Delta y$ and in time with an explicit scheme. The discrete update at iteration t for a pixel (x,y) is:

$$d(x,y,t+\Delta t) = d(x,y,t) + \Delta t\,(1-\lambda)\Big[\frac{d(x+\Delta x,y,t) - 2\,d(x,y,t) + d(x-\Delta x,y,t)}{\Delta x^2}$$

$$+\frac{d(x,y+\Delta y,t) - 2\,d(x,y,t) + d(x,y-\Delta y,t)}{\Delta y^2}\Big]$$

$$= d(x,y,t) + \frac{\Delta t\,(1-\lambda)}{\Delta x^2}\big(d(x+\Delta x,y,t) + d(x-\Delta x,y,t) + d(x,y+\Delta y,t) + d(x,y-\Delta y,t) - 4\,d(x,y,t)\big).$$

Applying the discrete Fourier transform to both sides gives:

$$\hat{d}(\omega_x,\omega_y,t+\Delta t) = \hat{d}(\omega_x,\omega_y,t) \; + \; \frac{\Delta t\,(1-\lambda)}{\Delta x^2}\,\hat{d}(\omega_x,\omega_y,t)\,\Big[e^{\,j\omega_x\Delta x} + e^{-\,j\omega_x\Delta x} + e^{\,j\omega_y\Delta y} + e^{-\,j\omega_y\Delta y} - 4\Big].$$

where ω_x, ω_y are spatial frequencies and t is the iteration (time) index. Substituting into the discrete update of the explicit scheme After factoring out the common terms, the amplitude update becomes:

$$\hat{d}(\omega_x,\omega_y,t++\Delta t) = \hat{d}(\omega_x,\omega_y,t)\Big\{1 \; + \; \frac{\Delta t\,(1-\lambda)}{\Delta x^2}\,\Big[2\cos(\omega_x\Delta x) + 2\cos(\omega_y\Delta y) - 4\Big]\Big\}.$$

Hence, the *amplification factor* α for frequency (ω_x,ω_y) is:

$$\alpha(\omega_x,\omega_y) = 1 \; + \; \frac{\Delta t\,(1-\lambda)}{\Delta x^2}\,\Big[2\cos(\omega_x\Delta x) + 2\cos(\omega_y\Delta y) - 4\Big].$$

For stability, we require $\alpha \le 1$ for *all* frequencies ω_x, ω_y. The worst-case (most negative sum inside brackets) is when $\cos(\omega_x\Delta x) = \cos(\omega_y\Delta y) = -1$, yielding:

$$2\cos(\omega_x) + 2\cos(\omega_y) - 4 = -8.$$

Hence,

$$\alpha_{min} = 1 - 8\,\frac{\Delta t\,(1-\lambda)}{\Delta x^2}$$

To satisfy $\alpha_{\min} \le 1$, we need:

$$-1 \;\le\; 1 - 8\,\frac{\Delta t\,(1-\lambda)}{\Delta x^2} \;\le\; 1, \quad\Longrightarrow\quad 0 \;\le\; 8\,\frac{\Delta t\,(1-\lambda)}{\Delta x^2} \;\le\; 2.$$

Thus,

$$\Delta t \;\le\; \frac{2\Delta x^2}{8\,(1-\lambda)} \;=\; \frac{\Delta x^2}{4\,(1-\lambda)}.$$

We call this bound `dt_cfl`. In the presence of non-diffusion terms, the actual time step must be at most this value to maintain stability.

(2) Curbing Large Disparity Jumps. We also want to prevent a single iteration from moving the disparity by more than one pixel. This ensures that each pixel "does not skip over" relevant grid positions it might otherwise match. Concretely, after computing the PDE update $\delta_d(i,j)$ for each grid point, we let

$$\delta_{\max} \;=\; \max_{i,j}\big|\delta_d(i,j)\big|$$

and then choose

$$\Delta t \;\le\; \min\big(\texttt{dt\_cfl},\; \alpha\,\frac{\Delta x}{\delta_{\max}}\big),$$

where Δx is the pixel size, and $\alpha \le 1$ is a safety factor.

This strategy ensures that if $\delta_{\max}$ is very large in one iteration, Δt shrinks accordingly, limiting any pixel's disparity update to at most $\alpha\,\Delta x$.

Energy Logging and Backtracking

To ensure a fair and consistent comparison of energy values across different modes (e.g., grayscale vs. color, pixel vs. patch), we unify the energy measurement used for logging. Specifically, we always convert images to grayscale and use pixel-matching loss when computing the logged energy, regardless of the mode used during optimization.

Additionally, to compensate for pixels that are excluded due to warping beyond the image boundary, we scale the data term by the factor $Area\ of\ \Omega/Mask\ Weight$, where the mask weight is given by

$$Mask\ Weight = \int_{\Omega} \mathbf{Sel}(x, y)\, dx\, dy.$$

so that the logged energy functional becomes

$$E(d) = \frac{Area\ of\ \Omega}{Mask\ Weight} \times (data\ term) + (smoothness\ term).$$

The rationale behind this scaling is that we want the logged energy to reflect the average cost per pixel over the entire image. If the image is wide enough, we expect a similar rate of data loss to occur for the pixels that are currently excluded. Note that this scaling is applied only for logging purposes; the actual energy functionals minimized during optimization vary depending on the case (e.g., grayscale or color, pixel or patch).

This unified logging allows meaningful comparisons between different cases when the same λ is used. Comparing results across different values of λ, however, is more complicated and was not attempted in this project. In fact, defining a fair loss metric across different λ values is difficult, as any evaluation using a specific λ-weighted energy functional would unfairly favor the model trained with that same weight. Moreover, there is no definitive evidence that a particular value of λ is ideal for capturing the true disparity of a given image. Therefore, to fairly compare models trained with different λ values, one would need access to the true disparity map.

Although the dataset does include a ground-truth disparity map, it is quantized into only five discrete levels. To use it for evaluation, we would need to quantize our estimated disparity map (e.g., using a KNN algorithm), and also apply a suitable transformation to align the estimated disparity range with the ground-truth. This alignment would require nontrivial post-processing and was not carried out in this project.

During experiments, we observed that the energy initially decreases but then begins to oscillate as the algorithm approaches a local minimum. This oscillation occurs because, near the minimum, the fixed step size becomes too large to precisely reach the optimum. To address this, we implemented a backtracking-style strategy.

The algorithm samples the logged energy every 50 iterations. If the energy increases compared to the previous sample, we reduce the step-size scaling factor α by a factor of 4. This in turn reduces the maximum allowed time step (as Δt is scaled by α), allowing the algorithm to take more cautious steps and better follow the local gradient.

If α drops below a threshold (e.g., 1×10^{-3}), we consider the algorithm to have converged to a local minimum and terminate the iteration early.

Summary

We now consolidate the above ideas into a complete explicit time-stepping scheme. Each iteration of the algorithm proceeds as follows:

1. **Compute the total gradient δ_d:**

$$\delta_d(x, y) = \underbrace{-\lambda\, \nabla_d E_{\text{data}}}_{datatermgradient} + \underbrace{(1 - \lambda)\, \Delta d}_{smoothnesstermgradient}.$$

   The data term gradient is computed using linear interpolation, while the smoothness term is obtained via the discrete Laplacian with Neumann boundary conditions.

2. **Determine the maximum gradient magnitude:**

$$\delta_{\max} = \max_{(x,y)} |\delta_d(x,y)| \,.$$

3. **Select a stable time step Δt:**

$$\Delta t \leq \min\left(\texttt{dt\_cfl},\ \frac{\alpha\,\Delta x}{\delta_{\max}}\right),$$

ensuring both numerical stability (via the diffusion-based CFL condition) and that no pixel updates its disparity by more than $\alpha\,\Delta x$ in a single iteration.

4. **Update the disparity field:**

$$d_{\text{new}}(x,y) = \max\left(0,\ d_{\text{old}}(x,y) + \Delta t \cdot \delta_d(x,y)\right),$$

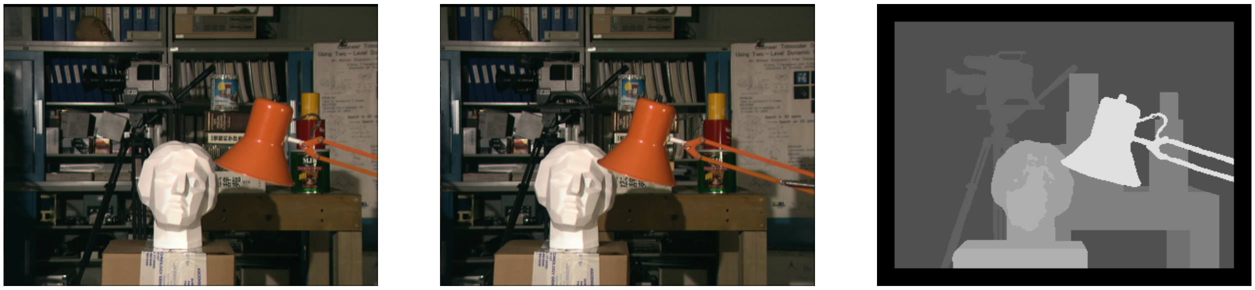
where the max enforces the physical constraint that disparity must remain nonnegative.

5. **Evaluate and log the energy:**

   - The energy is computed in grayscale using pixel-based loss, regardless of the optimization mode.
   - To compensate for any excluded (masked) pixels, the data term is rescaled by $Area\ of\ \Omega/Mask\ Weight$, so that the energy reflects the average cost over all pixels.

6. **Backtracking:** Every 50 iterations, the logged energy is compared to its previous value. If the energy increases, the step scaling factor α is reduced by a factor of 4. This effectively shrinks the maximum step size, allowing the algorithm to converge more precisely. If α falls below a predefined threshold (e.g., 1×10^{-3}), the algorithm is considered to have converged and terminates early.

5 Experimental Results



(a) Photo 1 (leftmost)

(b) Photo 5 (rightmost)

(c) Ground-truth disparity map

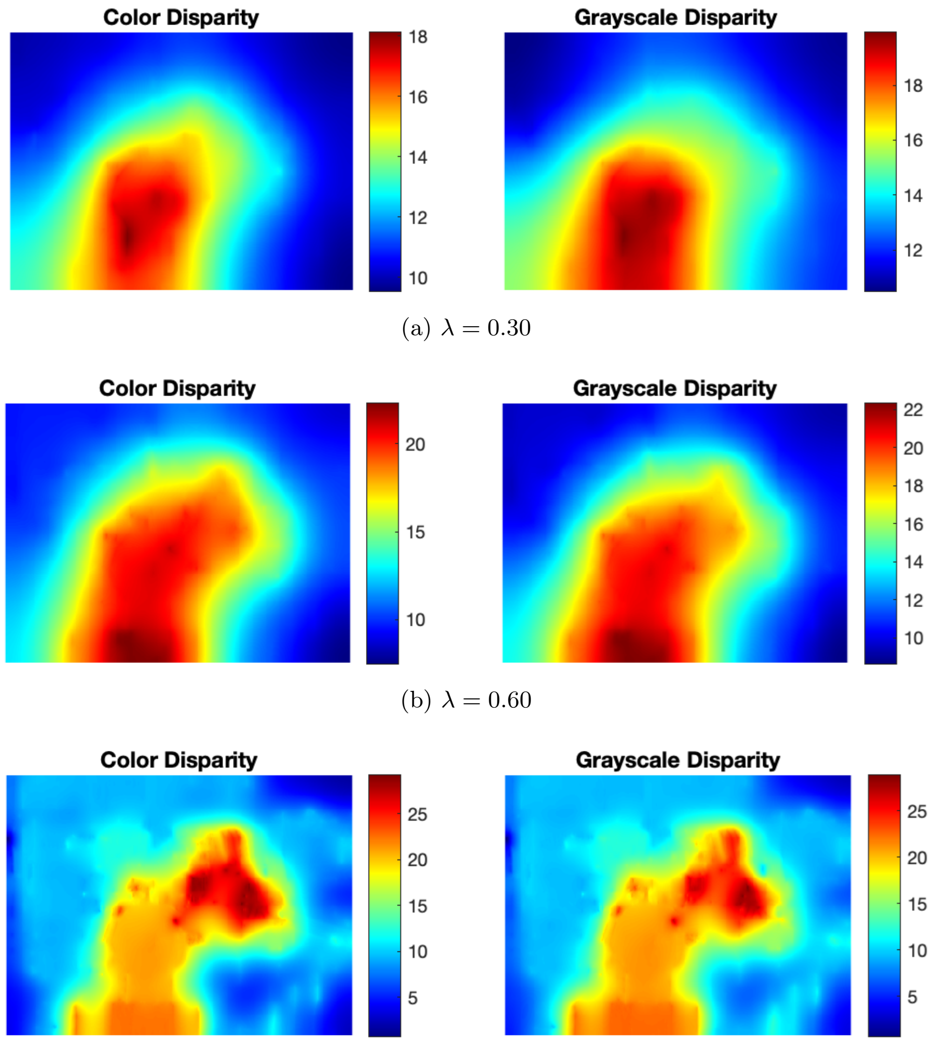
Figure 1: Stereo image pair (Photos 1 and 5) and corresponding ground-truth disparity.

We use the five-view Tsukuba stereo dataset, which comprises images taken from five horizontally shifted viewpoints (Photo 1 is the leftmost, Photo 5 is the rightmost). This dataset is one of the most widely used benchmarks for stereo disparity. We ran experiments under the following configurations:

- Color mode: `grayscale`, `color`.
- Data–smoothness weight: $\lambda = 0.3, 0.6, 0.9, 0.95$.
- View pairs: Photo 5 – Photo 1, Photo 5 – Photo 3, Photo 5 – Photo 4.
- Patch kernels: 1 × 1, 1 × 3, 1 × 5.

We present a qualitative comparison across different λ values, and a quantitative analysis for the remaining parameters.

Qualitative Analysis



(a) $\lambda = 0.30$

(b) $\lambda = 0.60$

(c) $\lambda = 0.95$

Figure 2: Disparity estimates using Photo 4 and Photo 5 with a 1×1 kernel, under different λ values.

As λ decreases, the disparity map becomes significantly blurrier and its maximum disparity value decreases, reflecting stronger diffusion. Objects farther from the camera are harder to segment (smaller disparity), while thin structures (e.g., lamp stand) are often lost. Grayscale and color modes yield qualitatively similar results, which we will see on the quantitative analysis later as well.

Quantitative Analysis

As the distance between viewpoints increases, both the required iterations and final energy generally rise, because matching more widely separated images is inherently harder and requires much more steps as there is a limit on the maximum step size. Interestingly, color mode and patch kernel size have only modest effect on convergence speed and final loss. Color information marginally accelerates convergence and does not significantly change the final energy, suggesting that brightness alone carries most of the matching signal. Similarly, larger kernels did not substantially alter the loss, likely because our scenes exhibit strong local intensity variation. Smaller lambda enlarged the number of iterations for convergence. This is because as the diffusion process becomes more active with the smaller lambda.

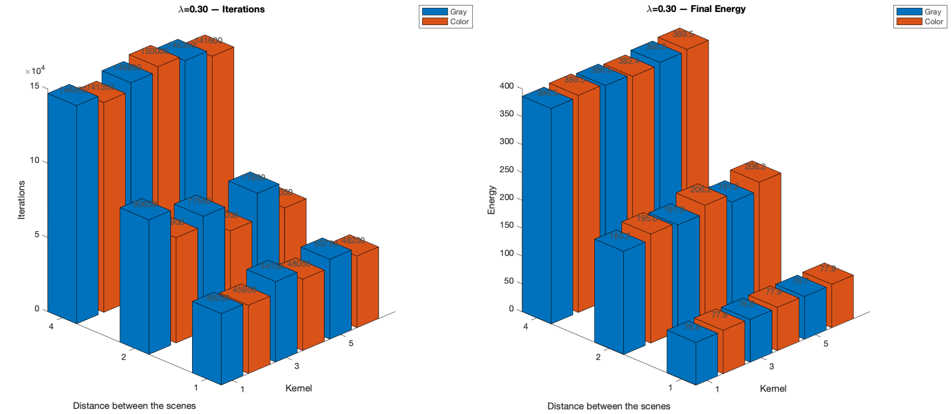


Figure 3: Iteration count and final energy for $\lambda = 0.30$.

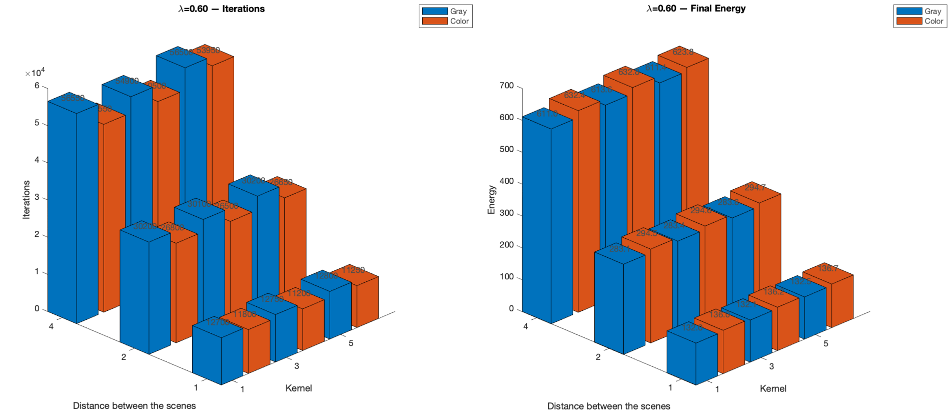


Figure 4: Iteration count and final energy for $\lambda = 0.60$.

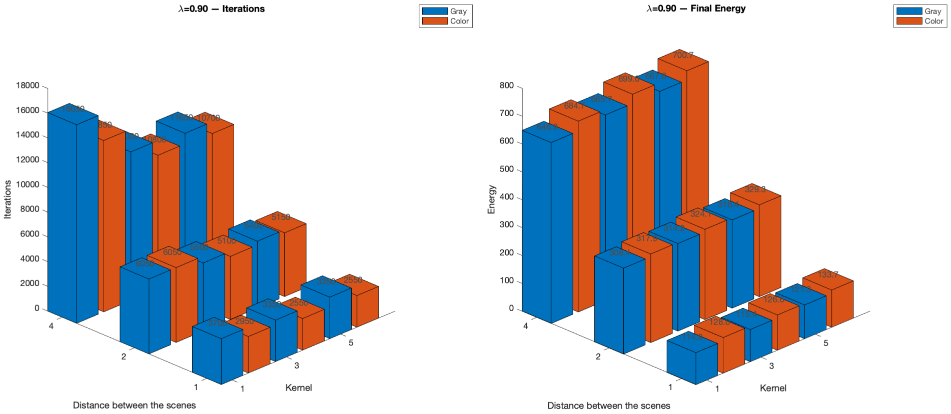


Figure 5: Iteration count and final energy for $\lambda = 0.90$.

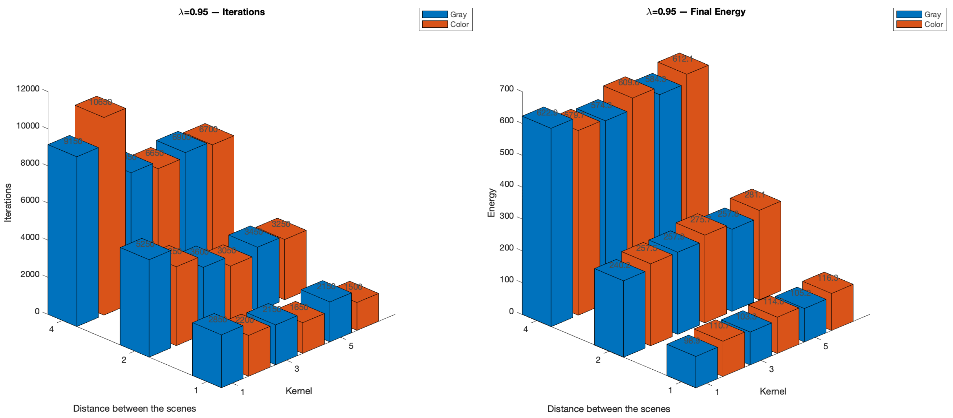


Figure 6: Iteration count and final energy for $\lambda = 0.95$.

6 Conclusion

In this project, we explored a variational-PDE approach to stereo disparity and uncovered fundamental limitations.

Interpolation and gradient consistency. We used linear interpolation to evaluate both the intensity $I_R(x-d)$ and its discrete derivative $\partial_x I_R(x-d)$. However, interpolating the intensity and then differentiating does not generally match differentiating first and then interpolating. This inconsistency can introduce bias in the data-term gradient. In future work, one could employ higher-order interpolation schemes (e.g., cubic or spline) or construct an interpolant whose analytic derivative coincides with the interpolated discrete gradient.

Step-size selection in a nonlinear PDE. Because our evolution PDE contains a nonlinear data term, we cannot rely on a single, globally valid CFL bound. In this project, we used a heuristic that clamps Δt by both a diffusion-CFL limit and by $\Delta x / \max |\delta_d|$. While this often stabilized the iteration, there were cases in which even very small Δt led to divergence. A more principled remedy I guess would be to pre-filter the images (low-pass) to limit their maximum spatial frequency and then derive a provably stable time step. Developing such an analysis would be challenging and is left for future work.

Effectiveness of patch and color extensions. We augmented the basic pixel-wise brightness matching with both RGB-color matching and patch matching. Experimentally, neither extension produced significant improvement in final energy or convergence speed. This suggests that—for these particular stereo scenes—color channels do not add robust correspondence cues beyond grayscale, and small horizontally-uniform patches may not capture enough contextual variation. In future work, one could explore anisotropic or data-adaptive kernels (e.g., vertical neighbors or Gaussian-weighted patches). Of course, each extension increases computational cost, so a careful trade-off between accuracy and efficiency would be needed.

Generalization and future experiments. All experiments in this project used a single Tsukuba scene. To draw broader conclusions about robustness, one must evaluate on diverse datasets—including real-world images with noise and varying illumination. Additionally, it would have been better to use Photo 3 as the reference, since its ground-truth disparity map is provided. Having a true disparity map would allow quantitative comparison across different λ values.

Overall, our variational-PDE framework provides a transparent way to encode data fidelity and smoothness and to derive an explicit update rule. However, achieving reliable convergence and high-quality depth estimates in realistic settings demands more sophisticated interpolation, stability analysis, and contextual matching strategies. These are promising directions for future work.