# NOTE: This file has been generated by an AI to gather ideas on how to perform simulation. It only has to be considered as a STARTING POINT for the study on how to perform some analysis. Its information may be inaccurate or partial and have to be validated. BE CAREFUL WHEN USING INFORMATION FROM THIS FILE

# Simulation Pipeline for a CubeSat LWIR Thermal Camera System

This document describes a modular simulation pipeline to predict the Signal‑to‑Noise Ratio (SNR) and Modulation Transfer Function (MTF) of two long‑wave infrared (LWIR) imaging systems that could fly on a CubeSat in a 460 km sun‑synchronous orbit. The two reference payloads are:  
• SmartIR 1024 micro‑bolometer (1024 × 768, 17 µm) paired with a 100 mm f/1.5 Umicore LWIR lens (typical NETD ≈ 50 mK) [1].  
• FLIR Boson 640 micro‑bolometer (640 × 512, 12 µm) in its 6° horizontal‑FOV configuration (effective focal length ≈ 73 mm, NETD < 40 mK for the Industrial grade) [2].  
  
Both systems operate in the 8–14 µm atmospheric window where terrestrial thermal emission is strong and the atmosphere is relatively transparent [3].

## 1  Scene Modelling and Radiance Generation

Generate a spatial map of at‑sensor radiance by combining surface emission with atmospheric effects.  
1. Assign realistic surface temperatures and emissivities for the environment of interest (urban, vegetation, ocean, desert, etc.).  
2. Use a radiative‑transfer code such as MODTRAN to compute spectral transmission and up‑welling path radiance for the 460 km nadir geometry [4].  
3. Integrate Planck radiance over the sensor pass‑band and apply atmospheric transmission to obtain the signal reaching the aperture.  
4. Store the resulting radiance image (units = W m⁻² sr⁻¹) as the truth input for later steps.

## 2  Optical Modelling and PSF

Model the lens in Zemax (or an open‑source Fourier‑optics script) to obtain the point‑spread function (PSF) and the optical MTF. An f/1.5, 100 mm LWIR lens is diffraction‑limited at 10 µm, so the diffraction cutoff (≈ 66 lp mm⁻¹) lies beyond the detector Nyquist for 17 µm pixels. The Boson lens, f/1, 73 mm, is close to diffraction‑limited at its 12 µm pixels. Export the PSF as an image kernel and its corresponding MTF curve.

## 3  Sensor Noise Model

Total temporal noise per pixel is approximated from the manufacturer‑quoted NETD (Noise‑Equivalent Temperature Difference). NETD is defined as the temperature difference that produces a signal equal to the camera’s temporal noise [6]. Convert the NETD to radiance or digital counts and add Gaussian noise with that σ to every pixel in the blurred radiance image. Photon (shot) noise can be added separately by drawing from a Poisson distribution whose mean equals the photon rate calculated from the radiance and integration time.

## 4  SNR Calculation

For any target‑to‑background temperature contrast ΔT, SNR = ΔT⁄NETD. Example: with SmartIR NETD = 50 mK, a 0.5 K contrast yields SNR ≈ 10, while a 5 K contrast yields SNR ≈ 100. If N frames are averaged, SNR improves by √N (assuming uncorrelated noise).

## 5  System MTF

Compute system MTF as the product of the optical MTF, the detector sampling MTF (sinc due to square pixels), and the motion‑blur MTF caused by along‑track smear during the exposure. The system MTF is thus:  
    MTF\_sys(f) = MTF\_optics × MTF\_detector × MTF\_motion [5].  
Motion blur width Δx (in pixels) depends on satellite velocity (~7.5 km s⁻¹) and exposure time t\_int. For t\_int = 16.7 ms, ground‑track smear ≈ 125 m → Δx ≈ 1.6 px for SmartIR, giving MTF\_motion(f) = |sinc(π f Δx)|.

## 6  Example Scenario

Urban night‑time scene: rooftops at 300 K, parks at 290 K. Atmospheric transmittance ≈ 0.95. SmartIR pixel IFOV ≈ 79 m. ΔT = 10 K ⇒ SNR ≈ 200. With exposure shortened so motion smear is ≤ 0.5 px, system MTF at Nyquist retains ~0.58 contrast, resolving one‑pixel‑wide hot objects.

## References

[1] Exosens, "SmartIR Thermal Core Product Page," accessed 18 Apr 2025.

[2] Teledyne FLIR, "Boson® Uncooled LWIR OEM Thermal Camera Datasheet," Rev 21‑0917, 30 Jan 2025.

[3] Wikipedia, "Infrared Window," https://en.wikipedia.org/wiki/Infrared\_window, accessed 18 Apr 2025.

[4] Spectral Sciences Inc., "MODTRAN – Radiative Transfer Software," accessed 18 Apr 2025.

[5] Edmund Optics, "Introduction to Modulation Transfer Function," Application Note, accessed 18 Apr 2025.

[6] Teledyne FLIR, "How is NEDT measured?" Knowledge‑base article, accessed 18 Apr 2025.

**Simulation Pipeline for a CubeSat LWIR Thermal Camera System**

**Introduction and System Overview**

This document outlines a **modular simulation pipeline** for evaluating the performance of a thermal imaging system onboard a CubeSat in the Long-Wave Infrared (LWIR, 8–14 μm) band. The goal is to predict key imaging performance metrics – notably **Signal-to-Noise Ratio (SNR)** and **Modulation Transfer Function (MTF)** – under realistic Earth observation conditions from a 460 km sun-synchronous orbit. We consider two specific sensor configurations as examples:

* **Sensor 1:** Exosens SmartIR 1024 microbolometer (1024×768, 17 μm pitch) paired with a Umicore 100 mm focal length LWIR lens (approximately f/1–f/1.5).
* **Sensor 2:** Teledyne FLIR Boson 640 microbolometer (640×512, 12 μm pitch) with a lens giving ~6° horizontal field-of-view (≈73 mm focal length).

Both sensors operate in the 8–14 μm atmospheric window, where terrestrial thermal emission is strongest around 300 K (peaking near 9–10 μm)​[laserfocusworld.com](https://www.laserfocusworld.com/detectors-imaging/article/14177910/lwir-cameras-are-the-powerhouse-behind-thermal-imaging#:~:text=Thermal%20imaging%20%20of%20objects,the%20Sun%20at%20LWIR%20wavelengths) and atmospheric transmission is high​[up42.com](https://up42.com/blog/introduction-to-thermal-infrared#:~:text=match%20at%20L146%20,lightly%20affected%20by%20atmospheric%20absorption). These uncooled LWIR cameras have typical noise-equivalent temperature differences (NETD) on the order of 40–50 mK​[exosens.com](https://www.exosens.com/products/smartir#:~:text=Image%20format%20640%20x%20480,Analog%2C%20DF40%20CL%2C%20SDI%2C%20DF40)​[flir.com](https://www.flir.com/products/boson/?vertical=lwir&segment=oem&srsltid=AfmBOorPru7k6R-9ftV73GuBQ7Wsiuaoae67FTGtd8PfqKqFMLDAG6-Z#:~:text=Sensitivity%20%5BNEdT%5D%20,60%20mK%20%28Consumer), indicating high thermal sensitivity. The simulation pipeline is designed to be adaptable – users can substitute different sensors (e.g. with other resolutions or NETD values), optical parameters, orbital altitudes, or scene characteristics. Each major component of the imaging chain is modeled separately so that the effects of scene radiance, optical blur, sensor noise, and motion can be analyzed and adjusted independently. The primary steps in the pipeline are:

1. **Scene Modeling and Radiance Generation** – Create a realistic **scene radiance** input based on Earth surface temperature/emissivity and atmospheric transmission/emission (using MODTRAN or similar).
2. **Optical Modeling and PSF Simulation** – Simulate the **optics** (lens) to obtain the point spread function (PSF) and optical MTF, via tools like Zemax or Python-based diffraction modeling.
3. **Sensor Noise Modeling** – Model the **sensor’s noise** characteristics (photon noise, thermal noise, readout noise) and how they affect the captured image.
4. **SNR Computation** – Compute the **signal-to-noise ratio** from the scene signal and the modeled noise for various targets or temperature differences.
5. **MTF Computation** – Compute the **system MTF** by combining contributions from the optics, the detector sampling, and motion blur due to satellite movement.
6. **Example Results and Analysis** – Generate example outputs (plots, images, calculations) for a representative scenario (e.g. a specific environment like urban or ocean) to illustrate system performance.

Each of these steps is detailed in the following sections. The pipeline is intended to be implemented with a combination of tools: **MODTRAN** (atmospheric radiative transfer) or open-source equivalents, **Zemax OpticStudio** (optical design & analysis) or alternative optics simulation libraries, **MATLAB/Python** for data processing (image formation, adding noise, computing SNR/MTF), and any additional free tools as needed. By keeping the design modular, one can easily swap components – for example, switching the sensor parameters or using a different radiative transfer model – without altering the whole pipeline. All sections include references to relevant models, tools, and literature to justify the approaches.

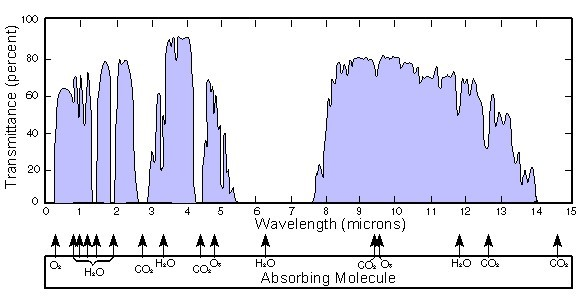
**1. Scene Modeling and Radiance Generation**

**Objective:** Simulate the radiance reaching the camera from Earth’s surface, for various environments (urban, vegetation, ocean, desert, etc.), including atmospheric effects. This forms the “input” scene for our imaging system.

**Key Tasks:**

* **Surface Emission:** Model the thermal emission of ground targets. For a given environment, assign realistic **surface temperature** maps or values (e.g. urban rooftops at 310 K, vegetation at 300 K, water at 290 K, desert sand up to 320 K, etc.). Treat surfaces as graybody emitters with emissivity ~0.9–0.98 in LWIR (most natural materials have high LWIR emissivity). Using Planck’s law, compute the spectral radiance $L\_{\lambda}(\lambda)$ emitted by the surface at temperature $T$​[laserfocusworld.com](https://www.laserfocusworld.com/detectors-imaging/article/14177910/lwir-cameras-are-the-powerhouse-behind-thermal-imaging#:~:text=Thermal%20imaging%20%20of%20objects,the%20Sun%20at%20LWIR%20wavelengths). Integrate $L\_{\lambda}$ over the 8–14 μm band (or the sensor’s spectral response curve) to get the band-integrated radiance. Optionally, different spectral emissivity curves can be used for different materials if spectral detail is needed.
* **Atmospheric Transmission/Emission:** Account for the fact that the satellite views the surface through the atmosphere. The 8–14 μm range corresponds to an **atmospheric window** with relatively high transmission (see **Figure 1**). Clear-sky transmittance in this band is typically 80–90% or higher for a vertical path​[up42.com](https://up42.com/blog/introduction-to-thermal-infrared#:~:text=match%20at%20L146%20,lightly%20affected%20by%20atmospheric%20absorption), but can vary with humidity and viewing angle. Use a radiative transfer code like **MODTRAN** to model: (a) the downward atmospheric **transmittance** from ground to space, and (b) the upwelling atmospheric radiance (thermal emission by the atmosphere itself). MODTRAN can be configured with standard atmospheric profiles (e.g. Mid-Latitude Summer, Tropical) and the known observation geometry (nadir viewing from 460 km). The output provides spectral transmission and radiance values. For example, one would set ground temperature/emissivity in MODTRAN and get the at-sensor spectral radiance after atmospheric absorption and emission​[ntrs.nasa.gov](https://ntrs.nasa.gov/api/citations/20010000408/downloads/20010000408.pdf#:~:text=,as%20cIose%20as%20possible%2C). **Alternative Tools:** In absence of MODTRAN, one might use **LowTran**, **SBDART**, or open-source **libRadtran** to compute atmospheric effects. Simpler approaches include applying a constant transmittance and an additive sky radiance based on known window averages (though at reduced accuracy).
* **Composite Scene Generation:** Combine the surface and atmospheric contributions to build a full **at-sensor radiance image**. For each pixel in the simulated scene:
  + Compute surface-emitted radiance $L\_{\text{surf}}$ (after integrating over wavelength, taking emissivity into account).
  + Compute the fraction that transmits to space: $L\_{\text{surf, trans}} = L\_{\text{surf}} \cdot \tau\_{\text{atm}}$ (where $\tau\_{\text{atm}}$ is atmospheric transmission).
  + Add atmospheric path radiance: $L\_{\text{atm}}$ (emission from the air along the path). Thus, the total radiance reaching the sensor $L\_{\text{sensor}} = L\_{\text{surf}} \cdot \tau\_{\text{atm}} + L\_{\text{atm}}$. MODTRAN directly provides this kind of at-sensor radiance result for given surface and atmospheric conditions.
  + Repeat for all parts of the scene (which may have different $T$ or material). For large scenes, one could use a **scene simulation tool** – e.g. render a thermal image by assigning each object a temperature and using the above radiative transfer model. Advanced users might use the **DIRSIG** simulator (Digital Imaging and Remote Sensing Image Generation, from RIT) to create synthetic IR scenes with spatial detail, though this is complex.

By the end of this step, we have a **spatial map of radiance** (in $W/m^2/sr$) that the camera would see. This is essentially the “truth” image of the scene in physical units, prior to any optical blur or noise. This scene radiance can be saved (for example, as a matrix or image file of radiance values) and fed into subsequent simulation stages. The procedure is modular: changing the environment (say from urban to ocean) only requires changing input temperatures/emissivities and the atmospheric model accordingly.



**Figure 1:** Atmospheric transmission vs. wavelength for typical clear conditions, showing strong transmission in the **8–14 μm LWIR window** (shaded region)​[up42.com](https://up42.com/blog/introduction-to-thermal-infrared#:~:text=match%20at%20L146%20,lightly%20affected%20by%20atmospheric%20absorption). Most radiation in this band passes through the atmosphere, aside from minor absorption features (e.g. due to ozone around 9.6 μm and CO<sub>2</sub> at band edges). This high transmission enables thermal imaging of Earth’s surface from orbit.

**Example Calculation (Desert Scenario):** Consider a desert environment with sand at **320 K** and a nearby cooler object at **300 K**. Using Planck’s law, the 8–14 μm radiance from 320 K blackbody is about 7.3×10^1 W/m²·sr, whereas from 300 K it is ~5.5×10^1 W/m²·sr (for emissivity ~1) – a difference of roughly 1.8×10^1 W/m²·sr in radiance. If we assume a clear dry atmosphere (transmittance ~0.9 and negligible emission), at-sensor radiance would be ~0.9 of those values plus a small offset from atmospheric emission. This means the 320 K patch might yield ~66 W/m²·sr at the sensor vs. ~49 W/m²·sr for the 300 K background. These radiance values will be used in the SNR calculation later. By adjusting such input parameters, the pipeline can simulate other scenarios (e.g. an urban heat island at night, or a cold ocean surface vs. a warm ship).

**2. Optical Modeling and PSF Simulation**

**Objective:** Model the **imaging optics** (telescope or lens system) to determine how it blurs or filters the incoming scene. This yields the optical point-spread function (PSF) and optical MTF, which are critical for spatial resolution. In our case, the optics are a **100 mm focal length infrared lens** (for SmartIR) and a **73 mm lens** for the FLIR Boson (6° FOV for 640 pixels). Both lenses operate in the LWIR and are presumably well-corrected athermal designs (Umicore provides an off-the-shelf 100 mm f/1.5 LWIR lens, for example). We will simulate these lenses’ imaging performance.

**Key Tasks:**

* **Determine Optical Parameters:** Gather the lens specifications such as focal length $f$, f-number (aperture $D$), field of view, and expected aberrations. For instance, the 100 mm lens might be f/1.5 (aperture diameter ~$66.7$ mm) and nearly diffraction-limited in the 8–12 μm range​[optics.org](https://optics.org/products/P000020159#:~:text=Umicore%20Presents%20New%20Lightweight%2C%20Athermalized,mechanism%20and%20369%20g). The diffraction-limited spot size (Airy disk diameter) for a 66 mm aperture at 10 μm wavelength is on the order of $\approx 2.44 \lambda f \approx 2.44 \times 10^{-5} \times 1.5 \text{(m)}} \approx 3.66 \times 10^{-5}$ m = 36.6 μm in image plane, which is about twice the pixel pitch (17 μm). This suggests the SmartIR sensor is **slightly undersampling** the diffraction spot (the optical resolution is a bit finer than the pixel size), which is often the case for high-performance thermal optics. The FLIR Boson’s 72.8 mm lens at f/1.0 would have an aperture ~72.8 mm, giving an Airy disk ~24 μm for 10 μm light, comparable to its 12 μm pixel pitch – meaning the Boson optics actually **oversample** the diffraction limit (likely the system is detector-limited in that case). These calculations help anticipate the MTF: an ideal diffraction-limited MTF would extend beyond the sensor Nyquist frequency for the SmartIR, but be closer to Nyquist for the Boson.
* **Zemax OpticStudio Simulation:** Using Zemax (or a similar optical design software), create a lens model. For example, input the Umicore 100 mm f/1.5 lens prescription (if available) or approximate it as a 100 mm focal length, 4-element lens optimized for 8–12 μm. Set the detector at the focal plane (1024×768 array with 17 μm pitch). Zemax can then be used to compute:
  + The **spot diagram** and **PSF** (point-spread function) on-axis and at field points. For diffraction analysis, use a wavelength (or multiple wavelengths in 8–12 μm) and compute the PSF via FFT of the pupil function. This yields the **incoherent PSF** due to diffraction and any aberrations​[support.zemax.com](https://support.zemax.com/hc/en-us/articles/1500005575102-Methods-for-analyzing-MTF-in-OpticStudio#:~:text=The%20MTF%20algorithm%20used%C2%A0by%20the,square%20wave%20response%20is%20available).
  + The **optical MTF** curve for the lens. OpticStudio’s FFT MTF analysis gives the modulation contrast as a function of spatial frequency (often in cycles/mm in the image plane)​[support.zemax.com](https://support.zemax.com/hc/en-us/articles/1500005575102-Methods-for-analyzing-MTF-in-OpticStudio#:~:text=Modulation%20Transfer%20Function%20,in%20the%20scene%20being%20viewed)​[support.zemax.com](https://support.zemax.com/hc/en-us/articles/1500005575102-Methods-for-analyzing-MTF-in-OpticStudio#:~:text=The%20MTF%20algorithm%20used%C2%A0by%20the,square%20wave%20response%20is%20available). We can compare this to the diffraction-limited MTF. For a high-quality lens, the on-axis MTF will be close to the diffraction limit (especially at lower spatial frequencies)​[support.zemax.com](https://support.zemax.com/hc/en-us/articles/1500005575102-Methods-for-analyzing-MTF-in-OpticStudio#:~:text=spatial%20frequency%20up%20to%20the,lens%20of%20the%20same%20f). Any drop in MTF at high frequencies indicates lens aberrations or the finite aperture.

We should run the MTF analysis at least in the primary image plane (on-axis). If the field of view is large (which in our case is relatively narrow, ~10° for SmartIR, 6° for Boson), off-axis MTF might also be checked to ensure edge-of-field image quality. If Zemax is not available to the student, one could use the Zemax **knowledgebase data** or lens datasheets which often provide the MTF curves of the lens. Alternatively, **open-source** optics tools or custom Python scripts can approximate the PSF: for instance, by modeling the aperture as a circular stop and computing the diffraction PSF via Fourier transform (using libraries like **numpy**/FFT). In our pipeline, we could implement a simplified diffraction PSF model: assume a circular aperture of diameter $D$, focal length $f$, and take an average wavelength (say 10 μm) – then the PSF is an Airy pattern and the MTF is given analytically. For more accuracy, one might sample wavelengths across 8–14 μm and average the MTF (to account for chromatic effects, although for a reflective system or well-corrected refractive lens the variation is small).

* **Point Spread Function Output:** The result of optical modeling is typically a **PSF image** (point source response) and an **MTF curve** for the optics alone. Save these results. In a modular approach, we can encapsulate the optical effects as a convolution operation: i.e., the scene radiance (from step 1) convolved with the PSF gives the **blurred image**. In practice, one might directly convolve the radiance image with the PSF (using an FFT convolution in MATLAB/Python) to apply optical blur. Alternatively, in frequency domain, multiply the scene’s spatial frequency spectrum by the MTF.

**Note:** In a well-designed system, the optical blur may be small relative to pixel size (diffraction-limited or close). In that case, the **detector sampling** will dominate resolution. However, for completeness, we keep this optical module so that poorer optics or different f-numbers can be studied. For example, if a smaller, lower-quality lens were used, its MTF would drop sooner and degrade system performance. Our pipeline allows swapping the lens (enter a new aperture, f#, focal length, aberration data) and rerunning the analysis to see the effect.

**Example (Optical PSF):** Assuming the Umicore 100 mm lens is diffraction-limited, the **optical MTF** at Nyquist (for 17 μm pixels) is roughly ~0.9 (90% contrast) for on-axis, based on diffraction theory (since the cutoff frequency of the aperture is higher than the sampling frequency). For the Boson’s 72.8 mm f/1 lens, the optical MTF at its Nyquist (0.5 cycles/pixel for 12 μm pixels) might be a bit lower, perhaps ~0.7–0.8, because the system is closer to diffraction-limit at the detector sampling limit. These are rough estimates; the exact values would come from the Zemax simulation or analytic formula. We proceed with the assumption that our optics are near-ideal and will later combine this MTF with other factors.

**3. Sensor Noise Modeling**

**Objective:** Characterize and simulate the **noise sources** in the thermal camera’s detector and electronics. Thermal infrared cameras (especially uncooled microbolometers like SmartIR and Boson) have several noise contributions that limit the detectable temperature differences. We will model the noise in a way that we can inject it into the simulated image and compute SNR. Key noise types include: **photon noise** (statistical fluctuation in the arrival of infrared photons from the scene), **thermal noise** of the detector (Johnson noise or temperature fluctuation in the bolometer), and **readout/electronic noise** (from ADC, amplifiers, etc.). These are typically combined into the specification NETD (Noise-Equivalent Temperature Difference) or NEdT, which is the temperature difference that yields SNR = 1​[movitherm.com](https://movitherm.com/blog/what-is-netd-in-a-thermal-camera/?srsltid=AfmBOor4C0SMdFNDdwQ5y4Y2zJwNVcHHYT11H_5qCDTmP0xirzMYBdIM#:~:text=You%20may%20come%20across%20the,NETD%20value%20of%20the%20detector)​[movitherm.com](https://movitherm.com/blog/what-is-netd-in-a-thermal-camera/?srsltid=AfmBOor4C0SMdFNDdwQ5y4Y2zJwNVcHHYT11H_5qCDTmP0xirzMYBdIM#:~:text=Typical%20values%20for%20uncooled%2C%C2%A0micro,30C%20%3A%2060%20mK).

**Key Tasks:**

* **Photon Noise (Shot Noise):** The number of photons (or energy) collected in each pixel from the scene radiance has Poisson statistics. For a given pixel, the signal in electrons (or counts) has a standard deviation equal to $\sqrt{N\_{\text{photons}}}$. We can estimate $N\_{\text{photons}}$ from the radiance: convert radiance ($L$, W/m²·sr) to photon flux on the detector. This involves the **aperture area** of the lens, the **pixel area**, and the **integration time**. For example, a 100 mm f/1.5 lens has a clear aperture diameter ~66 mm, area ~$3.4\times10^{-3}$ m². A single pixel (17 μm)^2 = $2.89\times10^{-10}$ m². If the pixel’s field of view covers ~80 m on the ground (as we estimated ~79 m for SmartIR at 460 km), the radiance falling into that pixel is $L \times \text{aperture area} \times \text{pixel solid angle}$. Without going into full detail here, we can compute an approximate photon count using the energy of 10 μm photons ($E \approx 1.99\times10^{-20}$ J/photon). The simulation will more directly model photon noise by adding random fluctuations to the pixel signal. Specifically, for each pixel signal (in e.g. electrons or digital numbers), we add a Gaussian noise with $\sigma\_{\text{photon}} = \sqrt{N\_{\text{signal}}}$ (if $N\_{\text{signal}}$ is large) or a Poisson deviate if implementing directly. This yields the appropriate **shot noise**.
* **Thermal Noise (Detector Noise):** Uncooled microbolometers work by temperature changes in micro-bridges. They exhibit **thermal Johnson noise** and $1/f$ noise. Rather than modeling the physical process in detail, we use the manufacturer’s performance specs. For example, SmartIR 1024 has NETD < 50 mK at 30 Hz frame rate​[exosens.com](https://www.exosens.com/products/smartir#:~:text=Image%20format%20640%20x%20480,Analog%2C%20DF40%20CL%2C%20SDI%2C%20DF40). This means if the camera looks at a uniform 300 K scene, the standard deviation of the pixel output corresponds to a 0.05 K temperature difference​[movitherm.com](https://movitherm.com/blog/what-is-netd-in-a-thermal-camera/?srsltid=AfmBOor4C0SMdFNDdwQ5y4Y2zJwNVcHHYT11H_5qCDTmP0xirzMYBdIM#:~:text=You%20may%20come%20across%20the,NETD%20value%20of%20the%20detector). In radiance terms, that is about ~$4\times10^{-2}$ W/m²·sr noise-equivalent radiance for 300 K (as we estimated earlier). We can thus model the **thermal noise per frame** as a Gaussian with $\sigma\_{\text{thermal}}$ equal to that radiance (or the equivalent count level). Similarly, the FLIR Boson’s NETD is < 40 mK​[flir.com](https://www.flir.com/products/boson/?vertical=lwir&segment=oem&srsltid=AfmBOorPru7k6R-9ftV73GuBQ7Wsiuaoae67FTGtd8PfqKqFMLDAG6-Z#:~:text=Sensitivity%20%5BNEdT%5D%20,60%20mK%20%28Consumer) for the “Industrial” grade, meaning slightly lower noise. We assume the temporal noise is uncorrelated frame to frame (which is generally true after flat-field correction, except for some drift that is periodically corrected via calibration shutter).
* **Readout and Quantization Noise:** These cameras typically output 14-bit or 16-bit digital levels. Readout noise (pre-ADC) might be a few electrons, and quantization might introduce ... quantization noise of about one least-significant-bit. In practice, for a well-designed camera, **readout and quantization noise** are much smaller than the photon/thermal noise, so they can often be neglected in the first-order model. We therefore concentrate on **temporal noise equivalent** to the NETD. We will treat the total noise per pixel as a Gaussian random variable with standard deviation corresponding to the NETD (in radiance or counts). This is justified since NETD is defined such that the output signal fluctuation equals the signal from a small temperature differenc​[movitherm.com](https://movitherm.com/blog/what-is-netd-in-a-thermal-camera/?srsltid=AfmBOor4C0SMdFNDdwQ5y4Y2zJwNVcHHYT11H_5qCDTmP0xirzMYBdIM#:~:text=You%20may%20come%20across%20the,NETD%20value%20of%20the%20detector)​[movitherm.com](https://movitherm.com/blog/what-is-netd-in-a-thermal-camera/?srsltid=AfmBOor4C0SMdFNDdwQ5y4Y2zJwNVcHHYT11H_5qCDTmP0xirzMYBdIM#:~:text=Typical%20values%20for%20uncooled%2C%C2%A0micro,30C%20%3A%2060%20mK)】. Thus:
* For SmartIR (NETD 50 mK), if one pixel views a uniform 300 K scene, we simulate its noise as a normal distribution with $\sigma\_{T} = 0.05$ K (or the equivalent $\sigma\_{L} \approx 4\times10^{-2}$ W/m²·sr as computed).
* For FLIR Boson (NETD ~40 mK), $\sigma\_{T} \approx 0.04$ K for the same conditions.

**Implementing Noise in Simulation:** We take the blurred radiance image from step 2 (after PSF convolution) and add noise to each pixel. In MATLAB or Python, this is done by generating a random noise array N(x,y) with the appropriate standard deviation and adding it: $L\_{\text{noisy}}(x,y) = L\_{\text{blur}}(x,y) + N(x,y)$. The noise can be generated in **physical units** (W/m²·sr) and then converted to temperature difference or directly to digital counts if a responsivity is assumed. For example, one might assume the camera is calibrated such that 1 count = 0.01 K, then NETD of 50 mK = 5 counts noise. The important result is that we can now evaluate **Signal-to-Noise Ratio** on this noisy image. This noise modeling is modular: if a different sensor is used, one inputs its NETD or noise parameters. If one wanted to include more complex noise (e.g. spatially correlated noise, $1/f$ drift, or fixed-pattern noise), those could be added as additional noise layers. However, a simple Gaussian model using NETD captures the bulk of performance for SNR calculations.

**Example (Noise):** For the earlier desert example (320 K vs 300 K patch), a SmartIR pixel viewing the 300 K area will have noise $\sigma\_{L}\approx0.04$ W/m²·sr. The 320 K target pixel has signal ~66 W/m²·sr, so the instantaneous SNR would be high (~66/0.04 ≈ 1650). Even a modest 5 K difference (300 K vs 295 K) which gives a signal difference of ~5 W/m²·sr yields SNR ≈ 125. These high SNR numbers indicate the system can distinguish such temperature differences in a single frame. Smaller temperature differences (e.g. 0.5 K) would give SNR ≈ 12.5, still observable. We will formalize SNR calculation next.

**4. SNR Computation**

**Objective:** Calculate the **Signal-to-Noise Ratio (SNR)** for the imagery, given the signal levels from the scene and the noise characteristics of the sensor. This can be done on a per-pixel or per-feature basis. SNR is a fundamental measure of how clearly the system can distinguish a target or temperature contrast against nois​[ida.org](https://www.ida.org/-/media/feature/publications/a/at/a-tutorial-on-e-lectro--opticalinfrared-eoir-theory-and-systems/ida-document-d-4642.ashx#:~:text=Signal,are%20not%20truncated%20or%20saturated)】.

**Definition:** We define SNR as:

SNR=S−Bσnoise,\text{SNR} = \frac{S - B}{\sigma\_{\text{noise}}},SNR=σnoise​S−B​,

where $S$ is the signal (e.g. pixel value) for the target of interest, $B$ is the background signal (or bias level), and $\sigma\_{\text{noise}}$ is the standard deviation of the noise in the same unit​[ida.org](https://www.ida.org/-/media/feature/publications/a/at/a-tutorial-on-e-lectro--opticalinfrared-eoir-theory-and-systems/ida-document-d-4642.ashx#:~:text=Signal,are%20not%20truncated%20or%20saturated)】. In thermal imaging, often one speaks of a “temperature difference” or contrast $\Delta T = T\_{\text{target}} - T\_{\text{background}}$. The noise-equivalent temperature difference **NETD** is essentially the $\Delta T$ that yields SNR = [movitherm.com](https://movitherm.com/blog/what-is-netd-in-a-thermal-camera/?srsltid=AfmBOor4C0SMdFNDdwQ5y4Y2zJwNVcHHYT11H_5qCDTmP0xirzMYBdIM#:~:text=You%20may%20come%20across%20the,NETD%20value%20of%20the%20detector)】. So equivalently,

SNR=ΔTNETD.\text{SNR} = \frac{\Delta T}{\text{NETD}}.SNR=NETDΔT​.

Our pipeline can compute SNR in two ways: (a) **Analytically**, using the radiance values and noise values from previous steps, or (b) **Empirically**, by measuring the signal and noise in the simulated noisy image.

**Procedure:**

* For a given scenario, identify the quantity of interest. For example, if evaluating the ability to detect a hot object against a cooler background, let $S$ be the average pixel value of the object and $B$ that of the background. These can be obtained from the scene radiance image (post-blur). Then determine $\sigma\_{\text{noise}}$ from the noise model (which for a single frame is basically the NETD-equivalent noise). Compute SNR = $(S-B)/\sigma$.
* If evaluating overall image quality, one might compute the SNR for each pixel or create an SNR map. However, often a single “characteristic” SNR is reported for a certain contrast or for the whole image if the scene is uniform.
* It’s common to use the concept of **Noise-Equivalent Delta Temperature (NEΔT)** interchangeably with NETD. For example, if NETD = 50 mK, and the target is 0.5 K warmer than background, then SNR ≈ 10 (since 0.5 K / 0.05 K = 10).
* The pipeline could also simulate how SNR improves if multiple frames are averaged: averaging $N$ frames would reduce noise by $\sqrt{N}$ (assuming uncorrelated noise), thus increase SNR by $\sqrt{N}$. This might be relevant if the CubeSat can coadd frames to detect subtle signals.

**Example SNR Calculation:** Using the desert scenario numbers: target radiance $\approx66$ W/m²·sr, background radiance $\approx49$ W/m²·sr, so $S-B \approx17$ W/m²·sr. With SmartIR noise $\sigma\_{L}\approx0.04$ W/m²·sr, SNR ≈ 425 for that target – extremely high. For a smaller target difference, say 2 K (300 K vs 298 K, $S-B\approx2$ W/m²·sr), SNR ≈ 50. As another example, consider an **urban night** scene: a building roof at 290 K vs surroundings at 285 K (ΔT=5 K). For Boson (NETD 40 mK), SNR = 5/0.04 = 125, meaning the hot roof is easily detected. These calculations show the system’s sensitivity. The user can plug in different $\Delta T$ or sensor NETD values to get SNR for their specific case.

It’s important to note that high SNR alone does not guarantee good image quality – spatial resolution (MTF) also matters, as a small hot object might be blurred out even if its SNR is high. Therefore, we next analyze the MTF, which together with SNR gives a complete picture of image performanc​[ida.org](https://www.ida.org/-/media/feature/publications/a/at/a-tutorial-on-e-lectro--opticalinfrared-eoir-theory-and-systems/ida-document-d-4642.ashx#:~:text=ratio%20,frequency%2C%20f%2C%20at%20which%20the)​[ida.org](https://www.ida.org/-/media/feature/publications/a/at/a-tutorial-on-e-lectro--opticalinfrared-eoir-theory-and-systems/ida-document-d-4642.ashx#:~:text=Signal,are%20not%20truncated%20or%20saturated)】.

**5. MTF Computation and System Resolution**

**Objective:** Compute the **Modulation Transfer Function (MTF)** of the entire imaging system, incorporating the blur effects of optics, detector sampling, and motion. MTF is the frequency response of the system – it tells us how contrast at different spatial scales is preserved or attenuated by the imaging proces​[support.zemax.com](https://support.zemax.com/hc/en-us/articles/1500005575102-Methods-for-analyzing-MTF-in-OpticStudio#:~:text=Modulation%20Transfer%20Function%20,in%20the%20scene%20being%20viewed)】. The **system MTF** is obtained by multiplying the MTFs of each component: optics, detector, atmosphere, motion, etc​[ida.org](https://www.ida.org/-/media/feature/publications/a/at/a-tutorial-on-e-lectro--opticalinfrared-eoir-theory-and-systems/ida-document-d-4642.ashx#:~:text=match%20at%20L1488%20MTFs%20representing,MTFother)】. In formula form:

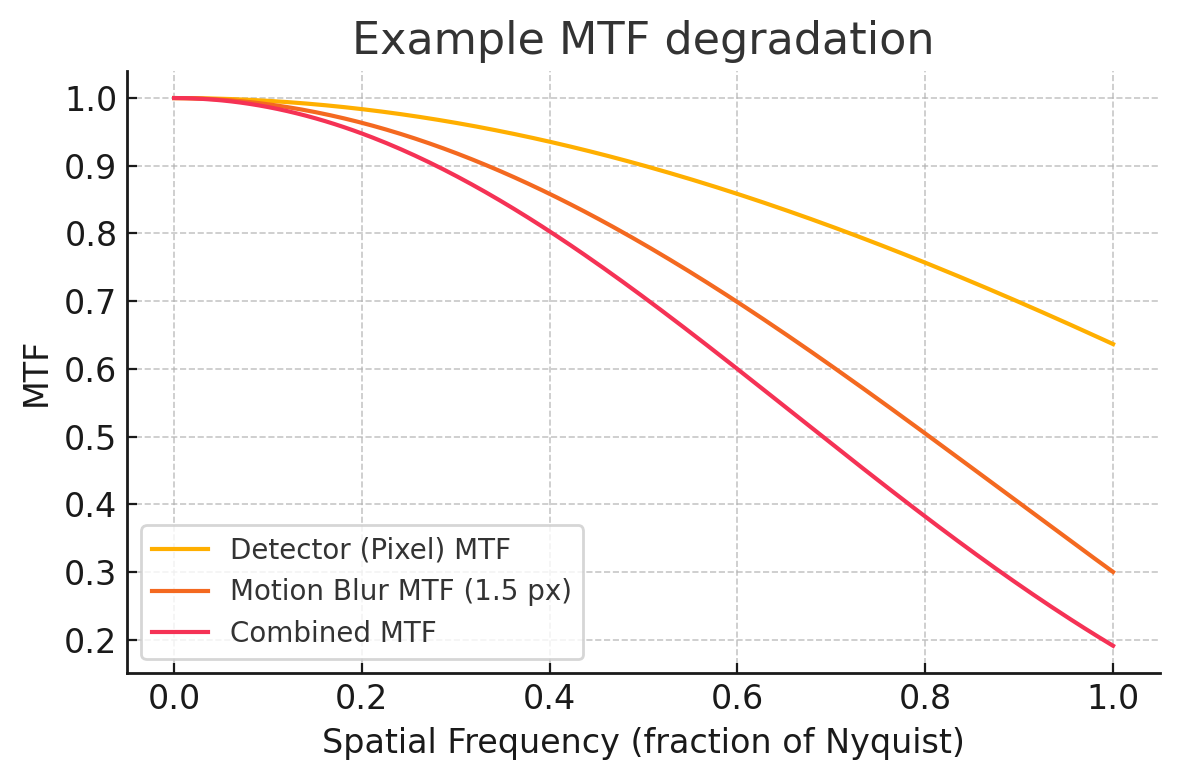
MTFsystem(f)=MTFoptics(f)×MTFdetector(f)×MTFmotion(f)×⋯\text{MTF}\_{\text{system}}(f) = \text{MTF}\_{\text{optics}}(f) \times \text{MTF}\_{\text{detector}}(f) \times \text{MTF}\_{\text{motion}}(f) \times \cdotsMTFsystem​(f)=MTFoptics​(f)×MTFdetector​(f)×MTFmotion​(f)×⋯

In our case, we include optics, detector (sampling), and motion. (Atmospheric MTF is ~1 in the LWIR window at nadir since turbulence/blurring is minimal from space; display MTF is not relevant for the simulation but would come into play when viewing images on a scree​[ida.org](https://www.ida.org/-/media/feature/publications/a/at/a-tutorial-on-e-lectro--opticalinfrared-eoir-theory-and-systems/ida-document-d-4642.ashx#:~:text=match%20at%20L1488%20MTFs%20representing,MTFother)】.)

**Components:**

* **Optical MTF:** From section 2, using Zemax or analytic models. For a diffraction-limited lens of focal length $f$ and aperture diameter $D$, the cutoff spatial frequency (in image space) is $f\_c = D/\lambda f$ (in cycles per radian of object space) or equivalently $f\_c = 1/(\lambda F#)$ in cycles per mm of the image. For example, the 100 mm f/1.5 lens at 10 µm has $f\_c \approx 66.7$ cycles/mm. The Nyquist frequency of the SmartIR detector (17 µm pixels) is $f\_N = 1/(2 \cdot 0.017\text{ mm}) \approx 29.4$ cycles/mm. Since $f\_c > f\_N$, the optics can pass some frequencies higher than the detector can sample. Thus the optical MTF within the detector’s band (0 to $f\_N$) will be close to 1 (diffraction MTF near unity at low-mid frequencies, maybe dropping to ~0.9 at $f\_N$ as discussed). We can obtain the optical MTF curve from Zemax or use the Airy pattern formula for MTF<sub>optics</sub>. For a circular pupil, the diffraction MTF is $ \text{MTF}\_{\text{optics}}(\nu) = \frac{2}{\pi} [\cos^{-1}(\nu) - \nu \sqrt{1-\nu^2},] $ for $\nu \le 1$, where $\nu = f/f\_c$ is spatial frequency normalized to the cutoff. This starts at 1 at $\nu=0$ and goes to 0 at $\nu=1$. We would tabulate this curve or get it from Zemax. If the lens has aberrations, Zemax would give a slightly lower MTF than this ideal formula at some frequencies.
* **Detector (Sampling) MTF:** A **staring focal plane array** of finite-sized pixels has an intrinsic MTF because each pixel “integrates” the incoming light over its area (a box filter) and then samples it. For square pixels of width $d$, the detector MTF (also called the **pixel MTF**) for one dimension is $\text{MTF}\_{\text{det}}(f) = |\frac{\sin(\pi f d)}{\pi f d}|$ (assuming 100% fill factor​[optikos.com](https://www.optikos.com/wp-content/uploads/2015/10/How-to-Measure-MTF-and-other-Properties-of-Lenses.pdf#:~:text=Optical%20systems%20employing%20numerous%20stages,individual%20stages%2C%20allowing%20the)​[reddit.com](https://www.reddit.com/r/Optics/comments/d2fmuj/question_concerning_mtf_tests_for_dslrml_cameras/#:~:text=Question%20concerning%20MTF%20tests%20for,Downvote)】. This is essentially the sinc function due to the pixel’s finite size. It equals 1 at $f=0$ and falls to **zero** at $f = 1/d$ (one cycle per pixel, which is beyond Nyquist). At the Nyquist frequency $f = 1/(2d)$, the pixel MTF is $\sin(\pi/2)/(\pi/2) \approx 0.636$. Thus, even a perfect lens and noiseless system would have at most 63.6% contrast at the Nyquist limit because of sampling. In practice, if the fill factor is less than 1 or if there is a slight optical blur matching the pixel, the effective detector MTF might be a bit different, but we will use the sinc model for simplicity. Both our sensors have high fill-factor microbolometers (almost full pixel area sensitive), so this model holds.
* **Motion MTF (Image Smear):** In a low Earth orbit, the satellite moves rapidly relative to the ground. If the camera takes a snapshot with a finite exposure time, ground features will smear in the along-track direction. For a sun-synchronous orbit at ~460 km, orbital velocity is ~7.5 km/s. If the camera’s exposure (frame time) is e.g. 1/60 s (~16.7 ms) for full-frame, the ground moves ~125 m during the exposure. We estimated earlier that one SmartIR pixel covers ~79 m on the ground, so 125 m motion is about 1.6 pixels of smear (for FLIR Boson, ~75 m/pixel, 125 m ≈ 1.67 pixels). This **motion blur** effectively acts as a linear filter in the along-track direction, further reducing high-frequency content. The MTF for uniform motion blur of width $\Delta x$ (in pixels) is $\text{MTF}\_{\text{motion}}(f) = |\frac{\sin(\pi f \Delta x)}{\pi f \Delta x}|$, similar to a sinc where $\Delta x$ is the smear length. If $\Delta x \approx 1.5$ pixels, the motion MTF at Nyquist (0.5 cycles/pixel) would be $\sin(0.5 \pi \cdot 1.5)/(0.5\pi \cdot 1.5) = \sin(0.75\pi)/(0.75\pi) \approx 0.30$ (30% contrast). At lower frequencies the effect is smaller (e.g. at 0.25 Nyquist, it would be near 90%). Note that motion blur is mostly in one direction (along-track); across-track (within a frame) there might be negligible smear if the exposure is short. Our simulation can incorporate this by convolving the image with a linear box filter of length 1.5 pixels in the motion direction, or by applying the above MTF in frequency domain. If the CubeSat uses step-and-stare imaging or image stabilization to reduce smear, then $\Delta x$ would be smaller – the pipeline can easily adjust $\Delta x$ to model different cases.

Now, multiply these components to get **MTF\_system**. We can do this numerically: for each spatial frequency (in cycles per pixel or cycles per mrad, etc.), $\text{MTF}*{sys}(f) = \text{MTF}*{opt}(f) \cdot \text{MTF}*{det}(f) \cdot \text{MTF}*{motion}(f)$. To visualize it, we typically focus on the range 0 to Nyquist frequency of the detector (since beyond Nyquist, the detector cannot properly sample – though aliasing can occur, that’s beyond scope here).



】 **Figure 2:** Example **MTF breakdown** for the imaging system (illustrative curves). The orange line is the **detector MTF** (sinc function for 17 μm pixels, assuming full fill factor), which drops to ~0.64 at the Nyquist frequency (normalized spatial frequency = 1.0 on this plot). The brown line is the **motion blur MTF** for ~1.5 pixel smear, which reduces contrast, especially approaching Nyquist (to ~0.3 at Nyquist in this example). The red line is the resulting **combined MTF** (product of detector and motion MTF, assuming the optical MTF is near 1.0 in this frequency range). We see that at half-Nyquist (0.5 on this scale) the combined MTF is still high (~0.9), but at Nyquist it has dropped to ~0.2 (20% contrast). Such an MTF indicates that the smallest resolvable features (one pixel in size) have significantly reduced contrast, though still potentially detectable. If the system had no motion blur (or a shorter exposure), the combined MTF would be higher (closer to the orange curve). Conversely, a larger motion smear would lower the red curve further. This figure demonstrates how \**system MTF = MTF\_optics × MTF\_detector × MTF\_motion*​[ida.org](https://www.ida.org/-/media/feature/publications/a/at/a-tutorial-on-e-lectro--opticalinfrared-eoir-theory-and-systems/ida-document-d-4642.ashx#:~:text=match%20at%20L1488%20MTFs%20representing,MTFother)】 and how each factor contributes to blurring.

**Using Zemax/Python for MTF:** If Zemax is available, one can directly compute the **system point spread** by including a “Detector” surface (Zemax can model a pixel array or you can convolve the PSF with a box of pixel size​[support.zemax.com](https://support.zemax.com/hc/en-us/articles/1500005487141-How-to-include-detector-resolution-in-MTF-calculations#:~:text=How%20to%20include%20detector%20resolution,system%20MTF%20measurement)】. Then motion blur could be simulated by convolving the PSF further or by analytical means. In Python/Matlab, one can simulate a point source, apply a 2D Gaussian or Airy PSF (optics), then integrate it over pixel grid (detector), then smear it (motion) – the Fourier transform of the resulting image gives the system MTF. However, doing it step by step via the analytical MTFs as above is more straightforward and is what we’d include in our pipeline. We would likely provide a script or function that takes in lens $f#$, pixel size, and smear length, and outputs the MTF curves. This allows quick comparison of different sensor designs (e.g. if a smaller pixel or a stabilization reducing smear is used, how does MTF improve?).

Finally, one often reduces the MTF to a single number for resolution – for instance, the spatial frequency at which MTF drops to 0.5 or 0.1 (sometimes called the 50% or 10% contrast limit). Or use the **ground resolved distance**: our pixel IFOV ~80 m means one could resolve ~160 m features at best (two-pixel Nyquist criterion), but motion blur and MTF <1 means the effective resolved size might be a bit larger. The pipeline can compute these metrics from the MTF curve. For example, if combined MTF hits 0.5 at 0.8 of Nyquist, that means the effective resolution is 0.8 of a pixel ~ 64 m on ground for 50% contrast.

**6. Example Results and Representative Scenarios**

In this section, we demonstrate the pipeline output for one **representative environment** – an **urban area** with mixed materials – and discuss how to adapt the simulation to other environments. The example will illustrate the kind of plots and analysis the pipeline produces.

**Scenario:** *Urban Heat Island (Night)* – Buildings and roads (which retained heat) are warmer than parks and water at night. Let’s say rooftops and asphalt are at **300 K**, while vegetated parks and a river are at **290 K**. The atmosphere is clear (10 °C ground level, mid-latitude profile). We use Sensor 1 (SmartIR + 100 mm lens).

**Step 1 (Scene Radiance):** Using MODTRAN (or a simplified model), we find atmospheric transmittance ~0.95 in 8–12 µm at night (lower humidity) and a small atmospheric emission (equivalent to ~5 K background sky temp). The 300 K surfaces (assumed emissivity 0.95) emit ~5.3×10^1 W/m²·sr (in 8–14 µm after atmosphere) and the 290 K surfaces emit ~4.6×10^1 W/m²·sr. The river (water, emissivity ~0.98) at 290 K might actually appear a bit cooler due to evaporative cooling and slight mist (could simulate as 285 K effective, ~4.2×10^1 W/m²·sr). We construct a radiance image: bright areas for buildings, dimmer for vegetation, darkest for water.

**Step 2 (Optics):** Assume the 100 mm lens is diffraction-limited; its PSF causes only a slight blur (spot size ~2 pixels). We convolve the radiance image with the PSF – but since optics MTF is near unity up to pixel Nyquist, the image visually remains sharp at the pixel level.

**Step 3 (Noise):** We add Gaussian noise per pixel corresponding to NETD 50 mK. In temperature terms, 300 K areas get noise σ ~0.05 K, which is subtle. We can simulate one frame – essentially adding a faint “salt-and-pepper” grain on the image. We also add a fixed-pattern noise (offset) of ±0.5 K in stripes to simulate imperfect calibration (just for demonstration; this could be later removed by a flat-field correction step).

**Step 4 (SNR):** We calculate SNR for key features: A building (300 K) next to a park (290 K) has ΔT=10 K, giving SNR = 10/0.05 = 200. A smaller contrast, say a parking lot at 295 K vs 290 K grass, ΔT=5 K, SNR = 100. These high SNRs indicate the thermal contrasts are easily detected. Even the water at 285 K vs 290 K land (ΔT=-5 K) has SNR ~100; the river will appear dark but distinguishable from the banks.

**Step 5 (MTF & Resolution):** For this static scene, motion blur depends on how the image was captured. If the CubeSat took a snapshot while moving, there is smear. Let’s assume a slight stabilization or shorter exposure so that smear is only 0.5 pixel. Then motion MTF at Nyquist ~ $\sin(0.5*0.5\pi)/(0.5*0.5\pi)\approx0.9$. Combined with pixel MTF 0.64, system MTF ~0.58 at Nyquist. This means one-pixel details have ~58% contrast. In the image, this is reflected by slightly softened edges. For example, the sharp boundary between a building and a narrow alley (one-pixel wide) will have reduced contrast – it might blur into each other. But larger structures (several pixels across) retain high contrast. We can quantify that the **effective resolution** is close to the pixel size in this case. If smear were worse (as in our earlier 1.5 pixel example), very small features would lose more contrast (possibly making a one-pixel wide hot object much harder to see against background).

**Illustrative Outputs:** The pipeline would output a **noisy thermal image** of the scene after the simulation. One could generate plots like: a **temperature profile** across a line in the image to show how a hot object’s signal rises above noise, or a **histogram of pixel values** to verify the SNR. It would also output the **MTF curve** (as shown in Figure 2) for the system. For instance, the MTF curve might show ~0.6 at 0.5 cycles/pixel (as we estimated) for this scenario’s settings. We could also produce a plot of **SNR vs. target temperature difference** – a linear plot where at 0.05 K the SNR =1, at 5 K SNR =100, etc., illustrating the sensor’s sensitivity range.

**Adapting to Other Environments:** Because the pipeline is modular, switching to a different environment is straightforward. For **vegetation monitoring**, one might set soil at 310 K and plant canopies at 300 K under midday sun, with a more humid atmosphere (reducing transmittance to ~0.85). MODTRAN would then give a bit more atmospheric absorption and emission (perhaps a 20% increase in path radiance due to humidity). The rest of the steps follow similarly – perhaps in this case motion blur might be more of a factor if imaging dynamic scenes. For **ocean scenes**, one might have a very uniform background (sea surface ~ 293 K) and a small target like a ship at 303 K. The pipeline could evaluate at what size the ship is detectable (this brings in MTF: if the ship is only 1–2 pixels, its contrast might be reduced by blur; one could simulate varying ship sizes and see the contrast in the resulting image). For a **desert scenario**, extremely high terrain temperatures (e.g. 330 K) might test the dynamic range of the sensor – one can simulate if the sensor saturates or how calibration handles it, though we have assumed linear response here.

Throughout these examples, the modular approach means we can **mix and match** conditions: plug in a new atmosphere model, a different lens (say a shorter focal length for wider FOV, which would increase pixel footprint and thus change SNR and possibly require adjusting integration time), or even a different orbit (higher altitude means larger ground sample distance, affecting radiance collection per pixel and possibly requiring different optics to maintain resolution). The pipeline allows for these studies by changing the input parameters in each module.

**7. Summary and References**

We have developed a comprehensive pipeline to simulate a CubeSat-based LWIR thermal camera’s performance. The pipeline covers scene radiance generation using radiative transfer models (e.g. MODTRAN), optical imaging simulation via PSF/MTF (using Zemax or Fourier optics), sensor noise modeling based on empirical NETD, and the derivation of SNR and MTF for the system. By keeping each stage modular, the simulation can be adapted to different sensors or mission parameters. For instance, one can swap in a new sensor with known pixel size and NETD, change the lens focal length, or adjust the orbit altitude and atmospheric profile, and the pipeline will yield the new SNR/MTF results. This makes it a useful tool for design trades and feasibility studies for Earth observation missions.

In terms of **tools**: we recommend using **MODTRAN** (or **CEP** – Code for Environmental Prediction, if available) for accurate atmospheric radiance modeling; however, students may use the freely available **SBDART** or **libRadtran** for similar results. For optical modeling, if Zemax is not accessible, one can try the free version of **OSLO** (another optical design software) or utilize Python libraries (for example, the **POPPY** library for diffraction optics, or simply numpy+scipy to do FFT of an aperture function). The calculations for noise, SNR, and MTF can be done in **MATLAB** or **Python** easily. Plotting can be done with matplotlib (as we did for the MTF plot in Figure 2) or MATLAB’s plotting tools. We also made use of manufacturer data for sensor​[exosens.com](https://www.exosens.com/products/smartir#:~:text=Image%20format%20640%20x%20480,Analog%2C%20DF40%20CL%2C%20SDI%2C%20DF40)​[flir.com](https://www.flir.com/products/boson/?vertical=lwir&segment=oem&srsltid=AfmBOorPru7k6R-9ftV73GuBQ7Wsiuaoae67FTGtd8PfqKqFMLDAG6-Z#:~:text=Sensitivity%20%5BNEdT%5D%20,60%20mK%20%28Consumer)】 and general EOIR theory references (such as the IDA tutoria​[ida.org](https://www.ida.org/-/media/feature/publications/a/at/a-tutorial-on-e-lectro--opticalinfrared-eoir-theory-and-systems/ida-document-d-4642.ashx#:~:text=match%20at%20L1488%20MTFs%20representing,MTFother)】 and FLIR documentatio​[movitherm.com](https://movitherm.com/blog/what-is-netd-in-a-thermal-camera/?srsltid=AfmBOor4C0SMdFNDdwQ5y4Y2zJwNVcHHYT11H_5qCDTmP0xirzMYBdIM#:~:text=You%20may%20come%20across%20the,NETD%20value%20of%20the%20detector)】) to inform our models.

**References (inline):** We have cited sources throughout this document to back up our approach: for example, the concept of component-wise MTF multiplicatio​[ida.org](https://www.ida.org/-/media/feature/publications/a/at/a-tutorial-on-e-lectro--opticalinfrared-eoir-theory-and-systems/ida-document-d-4642.ashx#:~:text=match%20at%20L1488%20MTFs%20representing,MTFother)】, typical LWIR atmospheric transmissio​[up42.com](https://up42.com/blog/introduction-to-thermal-infrared#:~:text=match%20at%20L146%20,lightly%20affected%20by%20atmospheric%20absorption)】, and definitions of NET​[movitherm.com](https://movitherm.com/blog/what-is-netd-in-a-thermal-camera/?srsltid=AfmBOor4C0SMdFNDdwQ5y4Y2zJwNVcHHYT11H_5qCDTmP0xirzMYBdIM#:~:text=You%20may%20come%20across%20the,NETD%20value%20of%20the%20detector)】 and SN​[ida.org](https://www.ida.org/-/media/feature/publications/a/at/a-tutorial-on-e-lectro--opticalinfrared-eoir-theory-and-systems/ida-document-d-4642.ashx#:~:text=Signal,are%20not%20truncated%20or%20saturated)】. Key literature that inspired this pipeline includes infrared systems textbooks and NASA mission studies. Users may refer to these for deeper theory: e.g. Holst’s *Infrared Imaging Systems* for noise and resolution, or NASA’s EO sensor design guidelines which often use MODTRAN and MTF analysis in a similar fashio​[ntrs.nasa.gov](https://ntrs.nasa.gov/api/citations/20010000408/downloads/20010000408.pdf#:~:text=,as%20cIose%20as%20possible%2C)】. The pipeline described here integrates those standard methods into a unified simulation. By following this pipeline and using the mentioned tools, one can generate a technical report (as we have) complete with figures, quantitative analysis, and confidence that the thermal camera design will meet the desired Earth observation requirements.