Low-Light Pick-and-Place in Webots with UR5e: Retinex Enhancement and Multi-Rate Visual Servoing

1st Haoyu Wang
Computer Science Department and Math Department
University of Wisconsin Madison
Madison, WI, the United States
hwang2376@wisc.edu

***Abstract*—Many vision-based manipulation pipelines break down under extreme low-light conditions, where RGB observations lose contrast and color separability, causing standard segmentation and tracking to fail. In this project, we build a low-light pick-and-place system in Webots using a UR5e arm with a wrist-mounted RGB camera to detect and sort red and blue cubes on a tabletop. To recover usable visual cues at illumination levels where raw thresholding fails, we estimate the reflectance of the image and perform HSV-based color segmentation with morphological refinement to estimate cube centers in image space. A finite-state manipulation pipeline (search, align, descend, grasp, lift, place) then executes Cartesian corrections using damped least-squares inverse kinematics with the Newton-Raphson method. A key engineering challenge is the mismatch between perception latency (Retinex runs approximately 6 s per frame) and the simulator control period (0.16 s). To address this, we adopt a multi-rate "sense-then-move" strategy that triggers Retinex sparsely, holds the last motion command between measurements, and re-corrects when new observations arrive. Experiments across varying illumination settings evaluate detection robustness, Retinex call frequency, end-to-end pick success, noise, and texture richness, demonstrating reliable operation in near-dark scenes while substantially reducing expensive enhancement invocations.**

***Index Terms*—Robotics, perception, low-light, webots, Retinex, CS 639**

I. Introduction

Vision-based robotic manipulation can fail catastrophically in extreme low-light environments. As illumination decreases, objects lose contrast and perceived color while sensor noise becomes dominant, causing simple segmentation pipelines to produce fragmented masks or miss the target entirely. In a closed-loop manipulation system, these perception failures quickly propagate into control errors: the robot may align to an incorrect location, descend off-target, and ultimately fail to grasp. This project studies low-light robustness in a controlled tabletop pick-and-place task: a UR5e arm equipped with a wrist-mounted RGB camera must detect, grasp, and sort colored cubes in the Webots simulator.

CS 639.

II. Technical Approach

This project addresses vision-based robotic pick-and-place under extreme low-light conditions in Webots, where raw RGB observations lose contrast and color separability and standard threshold-based segmentation fails. The system integrates algorithms from **Perception** and **Kinematics** to enable reliable cube detection and manipulation in near-dark scenes.

A. Perception: Low-Light Enhancement and Segmentation

To recover usable visual cues from near-black images, we employ the Weighted Variational Retinex model [1]. Retinex theory models an observed image S as the element-wise product of illumination L and reflectance R ($S = L \cdot R$). In our pipeline, we use the estimated reflectance R as an illumination-invariant representation that preserves object structure under severe lighting changes. Implementation details regarding the Retinex model are available in [2].

Given the reflectance image, we segment the target cube (red or blue) using HSV thresholding. To improve robustness against noise and mask fragmentation introduced by low-light enhancement, we apply morphological filtering:

1) **Opening**: erosion followed by dilation to remove small isolated blobs.
2) **Closing**: dilation followed by erosion to fill small holes and reconnect fragmented regions.

From the resulting mask, we extract contours and select the best valid component; its centroid (c_x, c_y) is treated as the image-space measurement z_t.

B. Kinematics and Control: Image-Based Corrections with DLS IK

The manipulation controller uses an image-based alignment signal. Let the pixel error be $e_u = c_x - c_x^\star$ and $e_v = c_y - c_y^\star$, where $(c_x^\star, c_y^\star)$ is the desired pixel (typically the image center with a small camera offset). Rather than commanding a continuous velocity servo loop, we convert (e_u, e_v) into a *small Cartesian position correction* of the end-effector, $\Delta p = [\Delta x, \Delta y, 0]^\top$, using a tuned proportional rule with conservative clamping to avoid overshoot.

To convert this Cartesian correction into a joint-space command, we solve inverse kinematics with the Damped Least-Squares (DLS) method. Given the current joint configuration q and geometric Jacobian $J(q)$ (computed from forward kinematics), we use the update:

$$\Delta q = J(q)^\top \left(J(q)J(q)^\top + \lambda^2 I \right)^{-1} e_{\text{pose}}, \tag{1}$$

where λ is a damping factor for numerical stability near singularities and e_{pose} is the 6-DoF pose error vector. This component directly connects **Kinematics** (Jacobian-based modeling) with **Control** (stable updates via damping).

III. Implementation Detail

A. System Design and Environment Setup

We implement the system in Python using the Webots controller API. The simulated platform is a UR5e manipulator with a parallel gripper 2f85 and a wrist-mounted RGB camera. The simulation timestep is fixed at 0.16 s.

- **Sensor:** wrist camera at 640×480 resolution with horizontal FOV ≈ 1.05 rad.
- **Environment:** to model extreme low light, we reduce scene illumination in Webots down to 1×10^{-6}, where raw color thresholding fails.

B. Engineering Decisions and Optimization

To balance robustness with execution speed, we make several key engineering decisions:

1) Dual-path perception. Because Retinex is computationally expensive, we implement a fast/slow switch. The system first attempts segmentation on the raw RGB frame (fast path). Only if no valid contour is found does it trigger Retinex and rerun HSV segmentation on the reflectance output (slow path). This reduces unnecessary Retinex calls when illumination is sufficient.

2) Finite-state machine (FSM) with latency handling. The high-level logic is encapsulated in an FSM (`search` → `align` → `descend` → `grasp` → `lift` → `place`).

A critical system constraint is latency mismatch: Retinex enhancement takes $\sim$ 6 s per frame, while the simulator advances every 0.16 s. Continuous, per-step visual feedback is therefore not preferable. To speed the system under extreme illumination, the multi-rate strategy in Sec. II is implemented. During align and descend states, the system takes decreasing step sizes to approach the cube.

3) Failure recovery. To prevent deadlock when perception fails (e.g., empty mask after enhancement), we maintain a failure counter. If detection fails for n consecutive sense attempts, the FSM resets to the `start` state to re-acquire a stable viewing pose.

During the descending state, the end-effector frequently occludes the cube, resulting in a loss of detection. This situation can be confounded with the lost case or perception failure. To solve this, we maintain the previously calculated moves and use a proportion of these values to take small steps away. If the cube is re-detected, then the system can continue execution; if the detection fails for adequately long ticks, then the FSM resets to the `start` state.

TABLE I
Most critical tuned parameters.

Parameter	Value
Timestep (Δt)	≈ 0.16 s
Lost tolerance (fail counter)	30 ticks (≈ 4.8 s)
Camera pixel offsets ($\Delta u, \Delta v$)	(+160, +10) px
Align pixel tolerance / gain	20 px; 8×10^{-4} (Cartesian, scaled by z)
Descend XY tolerance	8 px
Descend goal height z_{goal}	0.164 m
Close gripper command / hold	$g_{\text{close}} = 0.8$; > 150 ticks
Retinex gamma	2.2
Retinex hyper-parameters	$c_1 = 0.02,\ c_2 = 5.0,\ \lambda = 10.0$
Retinex stop / iterations	$\epsilon_1 = 6 \times 10^{-3},\ \epsilon_2 = 10^{-3}$; max outer 30
HSV (red) hue ranges	$[0, 10] \cup [170, 179]$
HSV (blue) hue range	$[100, 130]$
HSV thresholds ($s_{\min}, v_{\min}$)	(20, 20)
Min contour area / shape filters	900 px; square tol 1.50; extent min 0.35

C. Reproducibility and Parameters

The project code is modularized into:
`ur5econtroller.py` (main loop),
`control.py` (FSM + kinematics + IK),
`lib.py` (Retinex),
`detect.py` (segmentation utilities), and
`superv.py` (supervisor control).

Table I lists the key tuned parameters used in the experiments.

IV. Experimental Design

We evaluate the system with two experiments.

First, we vary illumination levels and measure Retinex call frequency, end-to-end task time, and pick success rate. This tests the hypothesis that sparse enhancement enables reliable operation under near-dark conditions with limited additional compute.

Second, we analyze the image quality metrics to investigate the impact of Retinex enhancement on the signal-to-noise ratio. Specifically, we calculate a high-frequency energy proxy and a residual noise proxy across different illumination levels. This evaluates the hypothesis that while Retinex recovers structural details (reflectance), it inevitably amplifies sensor noise, which degrades detection stability in extreme low-light conditions.

A. Experiment i

1) Quantitative Results Summary: Table II summarizes end-to-end performance under different illumination settings. It reports success rate, run time, and Retinex call frequency. Note that `sim_time` is measured in simulation steps, while `wall_time` and `ret_wall` are measured using `perf_counter`.

TABLE II
OVERALL RESULTS BY ILLUMINATION.

Cond.	N	Succ.	Time (s)	Ret.
1.0	20	90.0%	16.37	0.45
0.1	20	85.0%	333.39	40.75
2×10^{-6} (i)	4	50.0%	309.40	43.25
2×10^{-6} (ii)	4	100.0%	501.95	42.00
1×10^{-6} (ii)	4	0.0%	N/A	1

2) Key Observation and Discussion: Under low illumination (e.g., illum $= 0.1$ or 2×10^{-6}), the wall-clock runtime is dominated by Retinex enhancement, with Retinex accounting for $> 90\%$ of end-to-end time in our runs. Decreasing the descend alignment step size at illum $= 2\times10^{-6}$ increased total wall-clock time (501.95 s vs. 309.40 s), indicating that more alignment iterations amplify the cumulative enhancement cost. However, the extent of low light condition does not significantly affect the Retinex method's runtime, as shown by the similar runtimes between illum = 0.1 and illum = 2×10^{-6} (i) (where step size is shared).

In (i) with illumination 2×10^{-6}, the system frequently exhibited persistent oscillation due to the aggressive alignment step size (0.0004), leading to task failure. However, reducing the step size to 0.00012 in (ii) successfully resolved these failures. Given that the 0.0004 step size remains stable under higher illumination (1.0 and 0.1), this result indicates a critical trade-off: while the variational Retinex model recovers visibility in near-dark conditions, it concurrently amplifies sensor noise. This noise introduces jitter in the detected centroids, necessitating a more conservative control gain to ensure convergence.

At illumination 1×10^{-6}, all trials failed with only a single Retinex enhancement call, indicating that the detection pipeline could not identify any targets even after enhancement. This failure effectively marks the sensitivity limit of the variational Retinex model, where the signal-to-noise ratio becomes too low for structure recovery.

TABLE III
SINGLE-IMAGE HIGH-FREQUENCY ENERGY PROXY $hf = \mathbb{E}_{\mathrm{ROI}}[|\nabla^2 Y|]$ (MEAN±STD OVER FRAMES)

Illumination	Original hf	Enhanced hf	Reflectance R hf
1.0	0.01044 ± 0.00504	0.01427 ± 0.00881	0.02008 ± 0.01358
0.1	0.00490 ± 0.00139	0.01001 ± 0.00278	0.02045 ± 0.00640
2×10^{-6}	0.01038 ± 0.00277	0.01629 ± 0.00340	0.03893 ± 0.00911

TABLE IV
SINGLE-IMAGE RESIDUAL NOISE PROXY $\sigma_{\mathrm{RMS}} = \sqrt{\mathbb{E}_{\mathrm{ROI}}[(Y - \mathcal{G}(Y))^2]}$ WHERE $\mathcal{G}$ IS A GAUSSIAN BLUR (MEAN±STD OVER FRAMES)

Illumination	Original σ_{RMS}	Enhanced σ_{RMS}	Reflectance R σ_{RMS}
1.0	0.00660 ± 0.00203	0.00652 ± 0.00312	0.00844 ± 0.00491
0.1	0.00303 ± 0.00009	0.00481 ± 0.00035	0.00884 ± 0.00060
2×10^{-6}	0.00851 ± 0.00117	0.00944 ± 0.00100	0.01934 ± 0.00206

B. Experiment ii

1) Quantitative Results Summary: Tables III and IV summarize the single-image quality metrics across different illumination settings. Table III reports the high-frequency energy proxy (hf), defined as the expectation of the Laplacian magnitude, measuring texture richness. Table IV reports the residual noise proxy (σ_{RMS}), calculated via the root-mean-square deviation from a Gaussian-blurred baseline.

2) Key Observation and Discussion: Under moderate low illumination (illum = 0.1), the enhancement algorithm demonstrates effective detail restoration. The hf metric recovers from 0.00490 (Original) to 0.01001 (Enhanced), effectively matching the structural detail of the well-lit baseline ($hf \approx 0.010$ at illum = 1.0). However, this enhancement introduces a trade-off: noise amplification. The residual noise proxy σ_{RMS} increases from 0.00303 to 0.00481, indicating that the Retinex model amplifies sensor noise concurrently with the signal.

In extreme low light (illum = 2×10^{-6}), the high hf value in Original images (0.01038) is misleading, representing high-frequency sensor noise rather than valid object texture. This is corroborated by Table IV, where the Reflectance component exhibits the highest noise level ($\sigma_{\mathrm{RMS}} = 0.01934$), nearly doubling the noise observed at illum = 0.1. This indicates that the variational Retinex model becomes unstable in near-dark conditions, causing the Reflectance layer to absorb excessive noise artifacts ("false details") when the input Signal-to-Noise Ratio (SNR) is critically low.

V. REFLECTION AND FUTURE WORK

This project demonstrates that sparse, on-demand enhancement can enable reliable pick-and-place behavior under extreme low illumination, but the experiments also expose clear limitations in both perception robustness and runtime efficiency. In particular, extremely dark settings amplify noise-like high-frequency artifacts in the reflectance layer, which can lead to unstable detections and longer recovery cycles with smaller steps. Additionally, the end-to-end runtime is dominated by expensive enhancement calls and conservative motion execution, suggesting that perception–control co-design is essential for practical deployment.

A. Future Work

- **Aligned End-Effector.** The current end-effector remains the same rotation throughout the process and is the main reason for failure. A more adaptive solution is to rotate the end-effector to match the edges of the cube.
- **Uncertainty Estimation Modeling.** The quantitative noise analysis suggests that in near-dark conditions, high-frequency metrics are dominated by noise rather than real texture. A natural extension is to incorporate a dedicated estimation model (e.g., Kalman filter, Gaussian Filter) into the controller to avoid over-trusting noisy measurements.
- **Learned or Adaptive Sensing Policy.** The current multi-rate strategy uses fixed hold durations and hand-tuned thresholds. Future work could learn a policy that

schedules sensing events based on confidence, trading perception cost against alignment accuracy in a principled way.

- **Acceleration of Retinex Inference.** Retinex is the primary computational bottleneck. Possible improvements include downsampling with scale-aware thresholds or porting the enhancement step to GPU/parallel backends.

REFERENCES

[1] X. Fu, D. Zeng, Y. Huang, X.-P. Zhang, and X. Ding, "A weighted variational model for simultaneous reflectance and illumination estimation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2016, pp. 2782–2790.

[2] H. Wang, "Low-Light Image Enhancement: Weighted Variational Retinex (Project Website)," Accessed: Dec. 15, 2025. [Online]. Available: https://loucooc.github.io/