HELIOS: HYPERSPECTRAL HINDSIGHT OSTRACKER

Rafał Muszyński

UGent-IMEC-IPI-URC
Rafal.Muszynski@UGent.be

Hiep Luong

UGent-IMEC-IPI-URC
Hiep.Luong@UGent.be

ABSTRACT

In this paper we present our video object tracking algorithm HELIOS (HyperspEctraL hIndsight OStracker) designed for snapshot mosaic hyperspectral cameras, as part of our submission to the Third Hyperspectral Object Tracking Challenge 2023. HELIOS is based on a high-performance RGB-based one-stream tracking framework for joint feature learning and relational modeling based on self-attention operators. We further extend this framework to leverage information in the spectral domain through a new band selection algorithm and in the temporal domain through forward-backward consistency constraints. We obtained an averaged AUC of 0.634 and DP@20 pixels score of 0.846 on the validation dataset, thereby outperforming several published SOTA hyperspectral object tracking algorithms.

***Index Terms*—** hyperspectral video, object tracking, dimensionality reduction

1. INTRODUCTION

Hyperspectral video object tracking has attracted more and more attention from the computer vision research community over the last years. The recent advances in snapshot mosaic imaging (using Fabry-Pérot interferometry filters on top of the sensor elements) allows for hyperspectral acquisition at high frame rate as opposed to linescan and snapscan technology, but with the trade-off that only a limited number of wavelength bands can be acquired at once. Hyperspectral video contains more spectral information than grayscale or RGB color video, which helps to differentiate objects with different materials better. Therefore, it can advance several applications for which grayscale or RGB sensing are limited by their discriminative power between the object and its background.

In order to tackle the problem of object tracking in hyperspectral video, information in the spatial, spectral and temporal domains should be exploited in a joint manner. However, the main challenges for hyperspectral object tracking are the lack of large annotated training datasets (while larger datasets exist for RGB-based video), the absence of actual physical reflectance (due to the lack and limitations of calibration procedures in real-world conditions) and complex scenarios which poses subchallenges on the spatial, temporal and spectral axes. Some examples of these subchallenges are: occlusion or no light conditions (loss of spectral and spatial information), similar objects (same spectral and spatial information), appearance changes (spatial information changes over time), varying light conditions and out-of-plane rotations (spectral information changes over time).

Our proposed HELIOS tracker is based on OSTrack, which was trained on a large RGB video dataset with a large variety of different scenarios. Therefore, we can assume that this tracker exploits the spatial information very well (i.e., focusing on object shapes and texture). We then extend this SOTA RGB tracker by exploiting additional temporal and hyperspectral information. Similarly to [1, 2] we run the tracker backward in time. Our main goal is to keep the tracking result consistent between forward and backward pass. We use backward pass to find good frames for template update and to avoid identity switches by identifying similar objects distracting the tracker. OSTrack requires a three-channel input. We decided to design a band selection algorithm that discriminate the features of the pixels within the bounding box and the rest of the image. For the final model we combine the predictions of two trackers: one using the proposed band selection and one based on the provided false-color sequences.

2. RELATED WORK

OSTrack (an one-stream tracking framework for joint feature learning and relational modeling based on self-attention operators) is an high-performance SOTA RGB object tracking algorithm [3], which won one of the VOT tracking challenges 2023 [4].

Examples of SOTA video object trackers that focus on leveraging temporal information and avoiding identity switches are Neighbour Track [1] (similar objects are discarded from tracking based on back-tracking), BackTrack [2] (robust template update via backward tracking of candidate templates) and Keep Track [5] (a dedicated network to handle similar objects).

Finally, we non-comprehensively list examples of SOTA hyperspectral video object tracking (more SOTA papers can be found in the recent special issue on Hyperspectral Object Tracking in the Remote Sensing journal): BS-SiamRPN [6]

(band selection combined with a Siamese region proposal network), TrTSN [7] (transformer-based three-branch siamese network) and A Fast Hyperspectral Object Tracking Method Based On Channel Selection Strategy [8] (band selection based on the target and its surroundings).

3. PROPOSED METHOD

In this section we describe the HELIOS tracking algorithm in more details. We propose running the tracker forward and backward in time. In the first section we explain how we use the backward passes for template update. The second section discusses how we reuse information from the backward pass in the subsequent forward passes. The last section explains how we extract information from the hyperspectral data.

3.1. Forward-backward framework

OSTracker and other video object tracking algorithms localize the object in the current frame by template matching. Updating the template allows the tracker to update the visual features of the tracked object. It is useful in many situations, but it raises a question by itself: when is a good time to update the template? If we update the template when the tracker is lost, we risk an irreversible identity switch.

Similarly to [2], we propose checking whether it is safe to update the template by tracking the object back in time. Inconsistency between forward and backward passes could point to a possible loss of tracked object during the forward pass. To mitigate this, we propose finding a range of initial frames, for which the tracking is consistent, and restarting the tracking from the end of that range with a new updated template. We define consistency based on the IOU between passes. Figure 1 presents an example of an observed IOU between forward and backward passes. We mark a frame as consistent if the IOU is larger than $\text{IOU}_c = 0.2$, and every frame afterward is also consistent. This splits the frames in two, with the inconsistent range starting from the first frame. Until the backward and forward passes are consistent, we start a new backward pass from the middle of the inconsistent range.

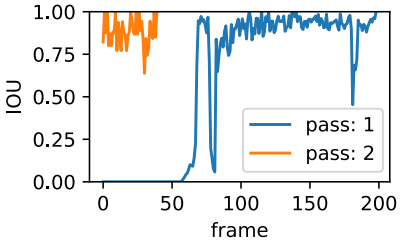


Fig. 1: Searching for consistent initial frames with backward passes

We repeat this procedure until the whole track is consistent or until we reach a maximum number of iterations (set to 3 in our experiments). The forward-backward framework is illustrated on figure 2.

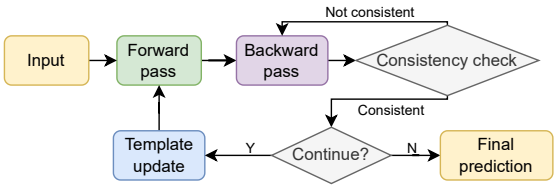
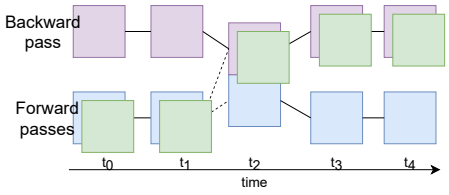


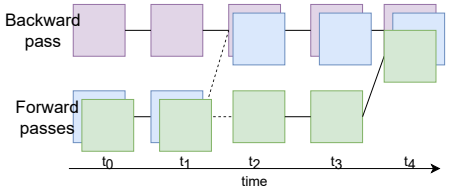
Fig. 2: Diagram depicting our forward-backward framework which allows us to reliably update the template.

3.2. Improving subsequent forward passes

Figure 3a presents a very common scenario that we aim to solve. During the initial forward pass (marked in green) the base tracker makes an incorrect decision at time t_2 and follows an incorrect object (a *distractor*). This becomes obvious after performing a backward track (marked in purple), which is not consistent with the initial forward pass. Our aim is to correct this mistake in the second pass (marked in blue).



(a) Similar objects



(b) Target early re-detection

Fig. 3: Examples of including information from the backward pass during the improved subsequent forward pass (blue)

However, because both the forward and the backward pass can, at any time, follow incorrect objects, the predictions from an inconsistent backward track can not be automatically discarded. As an example, in figure 3b we present a scenario in which the initial forward pass is not correct for frames t_2 and t_3, but recovers and re-detects the correct target at frame t_4. The subsequent backward pass follows the correct object, before losing it at frame t_1. The second forward pass can use the backward pass as a suggestion for the position of the target object. This can happen after occlusion, when the target is still barely visible.

We propose an algorithm for navigating the forward pass, which solves some of the described scenarios. Starting the tracking with frame at time t_0, for a frame at time $t > t_0$, we define a candidate set $\mathcal{C}$ consisting of bounding boxes C proposed by the base tracker and bounding boxes C' followed by the backward passes at the given frame. Each candidate c_i is assigned a score s_i. For $c_i \in C$ we use the prediction score outputted by our base tracker. For $c_i \in C'$ we use a constant value 0.2. We define the original prediction scores of the backward pass at time t as $s'_j(t)$. Next we evaluate whether the candidate c_i is a distractor or not. A candidate bounding box c_i is marked as a distractor and discarded under the following conditions:

1. There exists a $c_j \in C'$ which significantly overlaps c_i: $\text{IOU}(c_i, c_j) > \text{iou}_o$;
2. The prediction score is not very high: $s_i < 0.6$;
3. The backward pass is confident in its trajectory: $s_i\gamma < \min(s'_j(t), s'_j(t-1))$;
4. Current pass is not currently following the same trajectory as the backward pass of c_j;
5. At t_0, backward pass and the forward pass are not overlapping, i.e. $\text{IOU}(\text{forward}, j\text{-th backward}) < 0.2$.

In our experiments we set $\gamma = 1.1$ in order for the condition to be stricter. Finally, we pick the candidate c_i that maximizes the score and is not a distractor. If such a candidate does not exist, we reuse the bounding box from the previous frame, with a score equal to zero. Condition number 4 allows us to merge with trajectories as in 3b. After selecting the final bounding box c_{best}, we remember each backward pass i with $\text{IOU}(c_i, c_{\text{best}}) > \text{iou}_o$. In the next frame $t+1$ they will not be marked as distractors following the 4th condition.

3.3. Dimensionality reduction

Hyperspectral data contains a lot of information about the target object and the scene, which allows to more easily distinguish the target from the background. Our goal is to extract the most useful information from the hyperspectral data and pass it to a SOTA RGB tracker. To achieve this we've decided to perform dimensionality reduction via band selection. In this section we discuss the proposed band selection process and the processing steps required for transforming hyperspectral data into a three-bands video.

Similarly to [8] we select bands that maximize the separability between the target and the background pixels. Instead of relying on handcrafted features, we formulate a classification problem of labeling pixels as target (within the bounding box) and background (outside the bounding box) and select bands with the highest importance for the presented task. As the classifier, we use XGBoost [9], which is a well-proven algorithm for classification and regression tasks. A trained model can be used to assign weights to each of the bands. Finally, the bands with the highest weights are selected. The model is trained on an extracted patch centered around the target. We limit the training region to the search window size of the OSTracker. We perform band selection at the start of each new forward or backward pass.

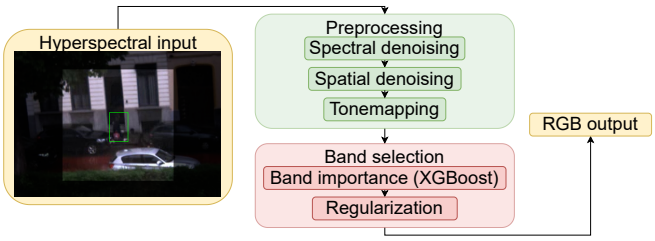


Fig. 4: Our pipeline transforming hyperspectral to RGB data. The band importance is calculated only for the initial frame. For the rest of the frames the same bands are selected.

During our experiments, we noticed several drawbacks of this approach: the model preferred noisy bands and highly correlated neighboring bands were often selected. This degraded tracker performance in the long run. To mitigate those issues we regularize the band selection, which increases the spread of the selected bands. Moreover, we apply preprocessing steps to denoise the hyperspectral data and increase the image contrast. The final pipeline, transforming hyperspectral data to RGB images, is presented in figure 4.

3.3.1. *Selecting bands based on their score and spread*

Let f_x be the importance score of a band x. We start by selecting the band i with the highest importance score. Next we select bands j, k that maximize the following score function $\sum_{x \in [i',j',k']} f_x + \alpha(\frac{n}{|i'-j'|} + \frac{n}{|j'-k'|})$ Where i', j', k' are the selected bands i, j, k in increasing order and n is the number of bands. α is a regularization strength we set to 0.005.

3.3.2. *Image processing*

To reduce image noise, we perform spectral denoising (averaging each band with neighboring bands) and spatial denoising (bilateral filter). Additionally, for each band, we perform tone mapping. The used tone mapping curve is a weighted average of a linear tone mapping curve, and tone mapping curves based on histogram equalization within the search window and for the whole image separately. With this approach we can balance the strength of local and global histogram equalization.

We select bands based on processed images. After the selected bands are fixed, there is no need to perform spatial denoising and tonemapping of the rejected bands. Therefore, we perform those operations only for the selected bands. This doesn't change the outcome while reducing computations.

4. EXPERIMENTS

In this section we describe our results on the Hyperspectral Object Tracking Challenge 2023 [10] dataset. For our experiments we used the original implementation of OSTracker [11] (model version: 384).

The challenge authors provide 95 training and 77 validation videos. We used the training dataset to adjust all parameters. The dataset contains videos captured in visible light (VIS, 16 bands), near infrared (RedNIR, 15 bands) and infrared (NIR, 25 bands). Additionally, a false color RGB version of each hyperspectral video is available.

We observed that in some of the VIS sequences the 13th band is corrupted. Therefore, we substituted the 13th band with an average of 12th and 14th band in all VIS sequences.

4.1. Results

Our approach improves the baseline OSTracker across multiple challenging scenarios. Figure 5 presents some examples. Two first rows present how our dimensionality reduction improved the results. The tracker is able to recover after occlusion, due to improved target visibility. In row 3 we present an example of improvements in the temporal domain. Small target can be better tracked with template update, and similar objects are not considered. In the 4th row we see an example where our algorithm fails. The correct object is tracked only a few frames longer than the baseline.

Table 1 presents the results of our work on the validation dataset. Our forward-backward framework as well as the band selection improve the results. Surprisingly, the results are not improved by combining both approaches. As an alternative, we combined two trackers: one with false color data and one with our dimensionality reduction. By weighing the predictions of both models we achieved the best model, achieving an AUC of 0.634.

model	AUC	DP20@20 pixels
Forward-backward Band selection False color	0.634	0.846
Forward-backward	0.628	0.842
Band selection	0.621	0.820
OSTracker	0.600	0.805
Forward-backward Band selection	0.596	0.799

Table 1: Ablation study on the validation dataset: surprisingly the combination of our band selection and forward-backward framework did not perform well, but we achieve the best result by adding false color data.

In Table 2 we compare the performance of our model with results of multiple SOTA approaches provided by the challenge organizers. OSTracker is a much more capable model than other architectures.

In Table 3 we compare how our proposed dimensionality reduction handles different camera types. Our approach improves the sequences with 15 & 16 bands, but underperforms on videos with 25 bands.

5. DISCUSSION

Our results demonstrate the importance of capitilizing on advancements in the field of RGB tracking, which benefits from larger datasets and more research. We further improved the chosen RGB tracker, which performs well using the spatial domain, by adding information from complementary tempral and spectral domain.

Currently, the pipeline shows promising results, but further work is required to transform it into a fast and reliable algorithm. Following the challenge deadline, we analyzed the results in more detail. We did not find any major reasons why using our dimensionality reduction in the forward-backward framework degraded tracking quality. In some cases the target object is wrongly labeled as a distractor. Possibly, parameters of distractor detection have to be adjusted. In at least one case our dimensionality reduction degrades after template update. As for the overall results, we found an implementation error which caused running 2 instead of 3 iterations of forward-backward framework for some scenes. We noticed, that our tone mapping can lead to decreased chances of target re-detection after concentrating on a wrong object. In some cases we observed flickering in video brightness.

6. CONCLUSIONS

In this work we presented our tracking algorithm HELIOS, submitted to the Third Hyperspectral Object Tracking Challenge 2023, which exploits information in temporal, spatial and spectral domain. The final model outperforms other approaches on the validation dataset provided by the challenge.

7. REFERENCES

[1] Yu-Hsi Chen, Chien-Yao Wang, Cheng-Yun Yang, Hung-Shuo Chang, Youn-Long Lin, Yung-Yu Chuang, and Hong-Yuan Mark Liao, "NeighborTrack: Improving Single Object Tracking by Bipartite Matching with Neighbor Tracklets," Apr. 2023, arXiv:2211.06663 [cs].

[2] Dongwook Lee, Wonjun Choi, Seohyung Lee, ByungIn Yoo, Eunho Yang, and Seongju Hwang, "BackTrack: Robust template update via Backward Tracking of candidate template," Aug. 2023, arXiv:2308.10604 [cs].

[3] Botao Ye, Hong Chang, Bingpeng Ma, Shiguang Shan, and Xilin Chen, "Joint Feature Learning and Relation Modeling for Tracking: A One-Stream Framework," Dec. 2022, arXiv:2203.11991 [cs].

Fig. 5: Visual examples of results produced by our models.

	MHT	**SiamBAN**	**SiamCAR**	**SiamGAT**	**STARK**	**TranST**	**OSTracker**	**Our result**
AUC	0.463	0.532	0.554	0.561	0.557	0.569	0.600	0.634
DP@20 pixels	0.733	0.756	0.768	0.770	0.762	0.777	0.805	0.846

Table 2: Comparison of our results with challenge authors' provided SOTA models on the validation dataset.

	VIS	**RedNIR**	**NIR**
False color	0.593	0.428	**0.674**
Our dimensionality reduction	**0.633**	**0.474**	0.655

Table 3: Tracking results (AUC) using falsecolor data vs our dimensionality reduction, for different input types.

[4] "The Tenth Visual Object Tracking VOT2022 Challenge Results :: ViCoS Prints," .

[5] Christoph Mayer, Martin Danelljan, Danda Pani Paudel, and Luc Van Gool, "Learning Target Candidate Association to Keep Track of What Not to Track," Aug. 2021, arXiv:2103.16556 [cs].

[6] ShiQing Wang, Kun Qian, and Peng Chen, "BS-SiamRPN: Hyperspectral Video Tracking based on Band Selection and the Siamese Region Proposal Network," in *WHISPERS*, Sept. 2022, pp. 1–8, ISSN: 2158-6276.

[7] Nan Su, Hongjiao Liu, Chunhui Zhao, Yiming Yan, Jinpeng Wang, and Jiayue He, "A Transformer-Based Three-Branch Siamese Network For Hyperspectral Object Tracking," in *WHISPERS*, Sept. 2022, pp. 1–5, ISSN: 2158-6276.

[8] Yifan Zhang, Xu Li, Feiyue Wang, Baoguo Wei, and Lixin Li, "A Fast Hyperspectral Object Tracking Method Based On Channel Selection Strategy," in *WHISPERS*, Sept. 2022, pp. 1–5, ISSN: 2158-6276.

[9] Tianqi Chen and Carlos Guestrin, "XGBoost: A Scalable Tree Boosting System," in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, Aug. 2016, pp. 785–794, arXiv:1603.02754 [cs].

[10] "Hyperspectral Object Tracking Challenge 2022," .

[11] Botao Ye, "OSTrack," Aug. 2023, original-date: 2022-03-22T07:02:22Z.