remote sensing

MDPI

Article

Urban Visual Localization of Block-Wise Monocular Images with Google Street Views

Zhixin Li [1], Shuang Li [1], John Anderson [2] and Jie Shan [1,*]

[1] School of Civil Engineering, Purdue University, West Lafayette, IN 47907, USA; li2887@purdue.edu (Z.L.); li4132@purdue.edu (S.L.)
[2] Geospatial Research Lab, Corbin Field Station, Woodford, VA 22580, USA; john.anderson@erdc.dren.mil
* Correspondence: jshan@purdue.edu

Abstract: Urban visual localization is the process of determining the pose (position and attitude) of the imaging sensor (or platform) with the help of existing geo-referenced data. This task is critical and challenging for many applications, such as autonomous navigation, virtual and augmented reality, and robotics, due to the dynamic and complex nature of urban environments that may obstruct Global Navigation Satellite Systems (GNSS) signals. This paper proposes a block-wise matching strategy for urban visual localization by using geo-referenced Google Street View (GSV) panoramas as the database. To determine the pose of the monocular query images collected from a moving vehicle, neighboring GSVs should be found to establish the correspondence through image-wise and block-wise matching. First, each query image is semantically segmented and a template containing all permanent objects is generated. The template is then utilized in conjunction with a template matching approach to identify the corresponding patch from each GSV image within the database. Through the conversion of the query template and corresponding GSV patch into feature vectors, their image-wise similarity is computed pairwise. To ensure reliable matching, the query images are temporally grouped into query blocks, while the GSV images are spatially organized into GSV blocks. By using the previously computed image-wise similarities, we calculate a block-wise similarity for each query block with respect to every GSV block. A query block and its corresponding GSV blocks of top-ranked similarities are then input into a photogrammetric triangulation or structure from motion process to determine the pose of every image in the query block. A total of three datasets, consisting of two public ones and one newly collected on the Purdue campus, are utilized to demonstrate the performance of the proposed method. It is shown it can achieve a meter-level positioning accuracy and is robust to changes in acquisition conditions, such as image resolution, scene complexity, and the time of day.

Keywords: monocular vision; visual localization; image similarity; template matching; deep learning; google street view; panorama

Citation: Li, Z.; Li, S.; Anderson, J.; Shan, J. Urban Visual Localization of Block-Wise Monocular Images with Google Street Views. *Remote Sens.* **2024**, *16*, 801. https://doi.org/10.3390/rs16050801

Academic Editors: Devrim Akca, Fabio Remondino and Rongjun Qin

Received: 16 September 2023
Revised: 22 February 2024
Accepted: 23 February 2024
Published: 25 February 2024

1. Introduction

Monocular image localization is to determine the pose (position and attitude or pointing) of a camera using an image. For urban environments, it has numerous applications in line of sight analysis, autonomous navigation, and robotics [1]. Hence, extensive efforts have been made to explore robust and precise methods by exploiting multiple GIS data as references, such as existing maps [2,3], semantic contexts in street views [4–7], geo-referenced traffic signs [8], aerial and satellite imagery [9–12], and features learned by deep neural networks [13].

However, accurately estimating the camera pose in urban environments is still challenging due to the unreliability and unavailability of GNSS data [4,5]. To address such difficulties, many techniques take advantage of unique environmental characteristics to determine the position of a vehicle. Among these techniques, Simultaneous Localization

and Mapping (SLAM) is considered the most advantageous one due to its ability to create a map of the surroundings while simultaneously locating the vehicle within it. Nonetheless, when absolute positioning is required or revisiting is not possible, SLAM based methods can be problematic due to the accumulation of errors caused by local measurements and expensive deployment costs [13]. Alternatively, an increasing number of studies are focusing on harnessing public multi-source geospatial data, such as Google Street View (GSV) and Mappy. The rapid improvement in the quality of these data sources and their widespread distribution, covering a significant portion of the world, makes them valuable. The 3D information they collect paves the way for enhanced functionalities like automated route planning, navigation, 3D buildings, fly-over tours, street views, and the overlay of public transport filters (subway and bus routes, travel schedules, etc.) [14]. This wealth of information, as indicated by reference [15–17], offers visual, spatial, and geographic insights, contributing to the creation of a unified global representation of our world.

We developed a framework for visual localization in urban areas with GSV panoramic images (or GSV images for short) as reference. The central idea is to match the query images captured on a moving vehicle, whose poses are to be determined, with the geo-referenced GSV images in the same geographic area. Once the correspondence is established, photogrammetric triangulation or structure from motion can then be used to determine the pose of the query images. The method starts with matching every query image with all GSV images in the database. A pre-trained semantic segmentation model, named SegFormer [18], is applied to segment the query image and extract a template containing all time-invariant, i.e., permanent objects, such as buildings and roads. The corresponding image patches are then found on every GSV image in the database using the Quality-Aware Template Matching (QATM) method [19]. To determine the similarity of the query template and GSV patch, a pre-trained Contrastive Language-Image Pretraining (CLIP) model [20] is used to convert them into deep features, followed by computing their cosine similarity. Such image-wise similarity is calculated for all pairs of query templates and GSV patches. In the next step, we divide all query images into a number of query blocks by their acquisition time and all GSV images into GSV blocks by their locations. By using the image-wise similarities, a block-wise similarity can be calculated that establishes the correspondence of a query block to its several best-matched GSV blocks. Finally, the images of a query block and the images of its top matched GSV blocks undergo a photogrammetric triangulation or structure from motion process (including tie point selection and bundle adjustment) to determine the poses of the query images. This process repeats for every query block until the poses of all query images are determined. As a nominal practice, one query block may consist of three query images, and one GSV block typically has three images. For each query block, we select the top three corresponding GSV blocks. As such, the photogrammetric triangulation could involve twelve images, including three query images and nine GSV images, to assure its reliability. Since GSV images have known interior and exterior orientation parameters (IOPs and EOPs), they essentially provide the geo-reference needed for the bundle adjustment calculation.

The rest of this paper is divided as follows: Section 2 exposes the related work regarding the state-of-the-art visual localization techniques and template matching approaches. Section 3 introduces the proposed framework and focuses on the image-wise and block-wise matching metrics computation. In Section 4, the results and effectiveness of our approach using three different mobile image datasets are evaluated. Finally, Section 5 discusses the factors affecting the performance of the block-wise matching strategy, while Section 6 concludes our work.

2. Related Work

2.1. Visual Localization with Perspective Images

Visual localization is critical for sophisticated computer vision applications. Most existing technologies predominantly depend on reverse perspective image search based on standard Field of View (FOV) and content-based image retrieval (CBIR). Initial studies

concentrated on employing a Bag-of-Visual-Words (BoVW) model [21] to extract visual features like ORB [22], SIFT [23], or SURF [24] for image representation and measuring image similarity using the cosine distance [21,25]. The BoVW model is robust to variations in scale, rotation, and translation, as it focuses on the presence and frequency of visual words rather than their exact spatial locations. However, the spatial information among visual words is ignored, which can lead to the loss of discriminative power in distinguishing images belonging to different conceptual categories. The VLAD (Vector of Locally Aggregated Descriptors) [26] approach accomplishes the same objective with condensed representations, allowing for the utilization of extensive datasets. Though VLAD shows good performance, it ignores high-order statistics of local descriptors, and its dictionary needs to be optimized for localization tasks. Over time, the improvements of these handcrafted feature based methods have demonstrated greater resilience to recurring structures [27], alterations in lighting and perspective [28], and even temporal changes, such as seasonal variations.

Recent advancements in deep neural networks (DNNs) have demonstrated that learning features from vast training datasets [29] can attain top-tier recognition accuracy, resulting in a shift towards deep learning-centric visual localization approaches [30–33]. A comparison of various feature extraction methods for convolutional neural networks (CNNs), as well as non-CNN techniques, is provided in [34]. NetVLAD [35] incorporates a layer into a standard CNN, transforming the final convolutional layer into a compact descriptor that emulates VLAD's [26] functionality. Lately, a method utilizing learned semantic descriptors [36] has surpassed NetVLAD and other prior methods on multiple visual localization benchmark datasets.

2.2. Visual Localization with Panoramic Images

In comparison to localization issues utilizing perspective images, employing panoramic images demonstrates broader applicability, as numerous geolocated street views are readily available and accurately captured across multiple periods. Generally, visual localization using panoramic images can be categorized into two distinct scenarios [37]. In the first scenario, both database images and query images are panoramic images, resulting in a panorama-to-panorama image matching problem [38–43]. In this case, image descriptors, such as SIFT and CNN-derived image features, serve as essential components.

The second scenario encompasses query images captured by a smartphone or standard monocular camera mounted on a moving vehicle, while the database comprises panoramic street views acquired through fisheye cameras. Addressing this perspective-to-panorama image matching challenge, prior research has generated simulated perspective images by projecting panoramas along the equator (i.e., a horizontal line) with a specific FOV [17,36], essentially transforming the problem into a perspective-to-perspective image matching one. Nonetheless, this approach results in an increased database size and fails to deliver high-quality matching outcomes due to an insufficient overlap ratio. To mitigate this issue, researchers have suggested employing CNN feature representations for the matching between monocular and panoramas without generating perspective images [19,37]. A sliding window is used on the CNN feature maps to search for matched monocular and panoramas.

2.3. Template Matching

Template matching is a computer vision technique that identifies the best corresponding part of an image (source image) for a given image pattern (template image) [44]. Traditional template matching methods use Sum-of-Squared-Difference (SSD) [45] or Normalized Cross-Correlation (NCC) [46,47] to estimate the similarity between the template and the source image. However, these methods become less effective as the transformation between the template image and the source image becomes more complex or nonrigid with different scales, which is common in real-world scenarios. Additionally, factors such as occlusions and color shifts make the above approaches more fragile and susceptible to failure. To address these issues, various approaches have been proposed. For instance, the Best-Buddies-Similarity (BBS) metric is introduced, which focuses on the nearest-neighbor

matches to exclude potential and bad matches by using the background pixels [48]. In [49], the Deformable Diversity Similarity (DDIS) concept is presented, which considers the likelihood of template deformation and measures the range of nearest neighbor feature matches between a template and a potential matching area within the search image. However, the decision of the threshold limits the usage of the mentioned methods. Hence, deep neural networks (DNN) based methods are developed to mimic the functionality of template matching [50,51] by extracting the deep features with pre-trained deep models. Furthermore, QATM is proposed to take the uniqueness of pairs into consideration rather than simply evaluating the matching score [19].

3. Methodology

Our work aims to address the urban visual localization problem under a mobile scenario, which involves determining the pose of a sequence of monocular mobile images by using the geo-referenced GSV images. We are targeting applications when GNSS signals are weak or not available in complex urban environments or under difficult conditions. Our method has three prerequisites. First, location knowledge of the geographic area of the mobile images is necessary to access the GSV images that encompass the same area. Second, treating the mobile images as query images and geo-referenced GSV images as source images, we assume they should contain a sufficient number of permanent or time-invariant objects for successful matching, since they are likely not collected during the same time. Thirdly, the exterior orientation information of the GSV images should be known so that they can be used for the pose determination of the query images in a world coordinate system. To be more specific, the longitude, latitude, altitude, and northing angle of the GSV images recorded by Google will be available as reference data. Since the GSV image is a spherical panorama with a 360-degree view, the northing angle is defined as the clockwise angle difference (0–360°) between the geographic north direction and the direction from the optical center to the center pixel of the GSV image. Lastly, the interior orientation of the query images should be available as a common practice in photogrammetry. It should be noted that no initial precise location for the query image is needed because the workflow will search the entire GSV image database to identify the most suitable GSV images. Instead, only the geographic area of the operation needs to be known in advance.

Figure 1 shows the workflow of our solution. It starts with calculating image-wise similarities. The query images are semantically segmented using a pre-trained model named SegFormer [18]. As the result of this segmentation, a template, i.e., a rectangular region of the query image, is formed that encloses all permanent objects, such as buildings and roads. Secondly, for each detected query template, we use the QATM technique [19] to find the corresponding patches from all GSV images. In the next step, we use a pre-trained Contrastive Language-Image Pretraining (CLIP) model [20] to convert a query template and its corresponding GSV patch into feature vectors, which are then used to calculate the cosine similarity as the measurement of image-wise similarity. Such image-wise similarity calculation will be repeated for all pairs of query images and GSV images in the database. Based on these image-wise similarities, the fourth step calculates the block-wise similarity. To do so, we form a query block with n consecutive images in the sequence according to their acquisition time. Similarly, starting from the most northwestern GSV image, m spatially adjacent GSV images along the trajectory of the query images are iteratively grouped into one GSV block. It should be noted that one query image and one GSV image are only assigned to one corresponding block. Next, we calculate the block-wise similarity between a query block and a GSV block. For each query and GSV block pair, we select the best image-wise similarity for each query image and take the average for all query images in the query block. This average of the top n image-wise similarities is used as the block-wise similarity between the query block and the GSV block.

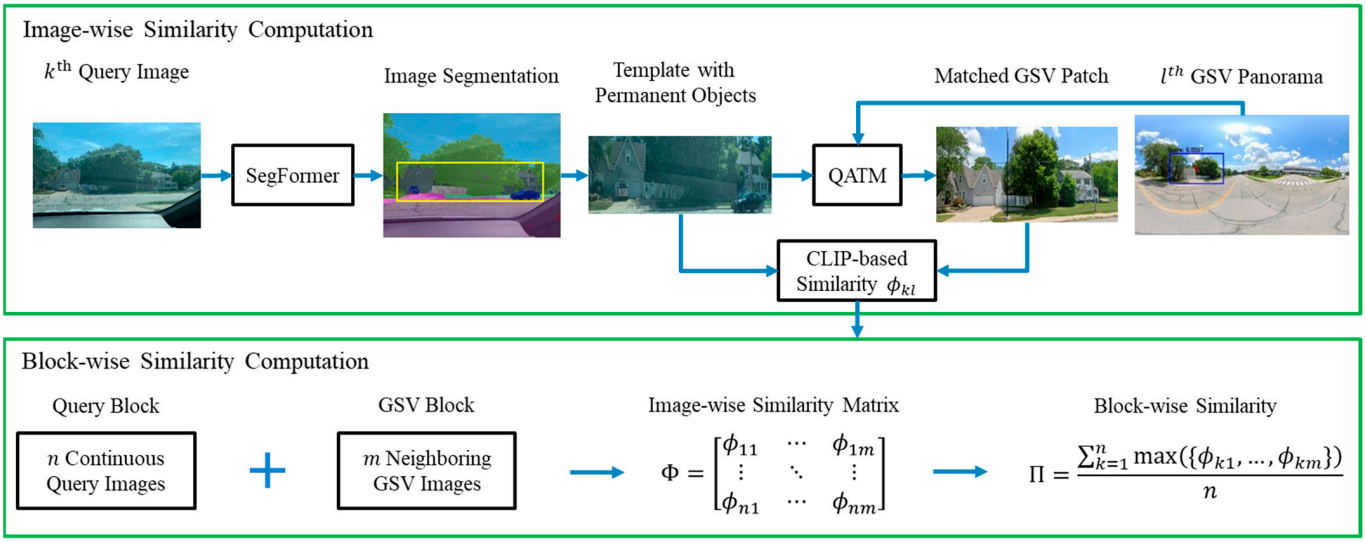


Figure 1. Workflow of query image and GSV image matching. The pretrained SegFromer is first applied to segment each query image to generate a template consisting of all permanent objects. QATM is used to find a corresponding image patch on all GSV images for every query template. A pair of a query template and GSV patch are converted to feature vectors using CLIP, whose cosine is calculated as the image-wise similarity. Block-wise similarity is the average of the top image-wise similarities within a block pair. For each query block, the GSV block with the highest blocks-wise similarity is regarded as its best match.

For a query block, the GSV block with the max block-wise similarity is identified as its best match. However, to assure reliable performance, for each query block, we select several top matched GSV blocks for the subsequent photogrammetric triangulation. We adopt Agisoft Metashape 2.0 (https://www.agisoft.com/, accessed on 31 December 2022) for this process. Its input is the images in a query block and the images in all selected top matched GSV blocks. It starts with routine tie point extraction and matching, followed by a bundle adjustment calculation, in which the known exterior orientation of the GSV images is used as reference. The poses of the images in a query block are the result of the bundle adjustment.

3.1. Permanent Object Segmentation

We recognize that query images often contain time-variant objects like trees, cars, people, and clouds, which are inconspicuous and not useful for finding similar images in the GSV database. However, time-invariant objects like buildings and traffic signs retain their spectral information over time and are considered permanent objects. To identify these objects, we utilize the SegFormer [18] model trained using Cityscapes [52], available from the MMSegmentation toolbox (https://github.com/open-mmlab/mmsegmentation, accessed on 10 December 2023), to segment all query images and assign a class code to every pixel. SegFormer presents a straightforward, efficient, and robust semantic segmentation framework. This framework consists of a hierarchical Transformer encoder [53] and a lightweight multilayer perception (MLP) decoder head. Given our goal of estimating the pose of a moving vehicle, it is impractical to train or fine-tune a deep model to achieve improved segmentation performance, especially considering the constraints of time and computational resources. The SegFormer model not only delivers top-tier results in semantic segmentation but also exhibits excellent transferability [54]. This means that once the model is trained on a specific street view image dataset, it can be directly applied to other datasets without the need to train from scratch or fine-tune, resulting in strong performance. The segmentation results with SegFormer consist of label maps, in which each pixel is assigned to a specific class. These classes consist of 8 permanent objects, with

Cityscapes label IDs ranging from 0 to 7. The objects include road, sidewalk, building, wall, fence, pole, traffic light, and traffic signs, as displayed in Figure 2.

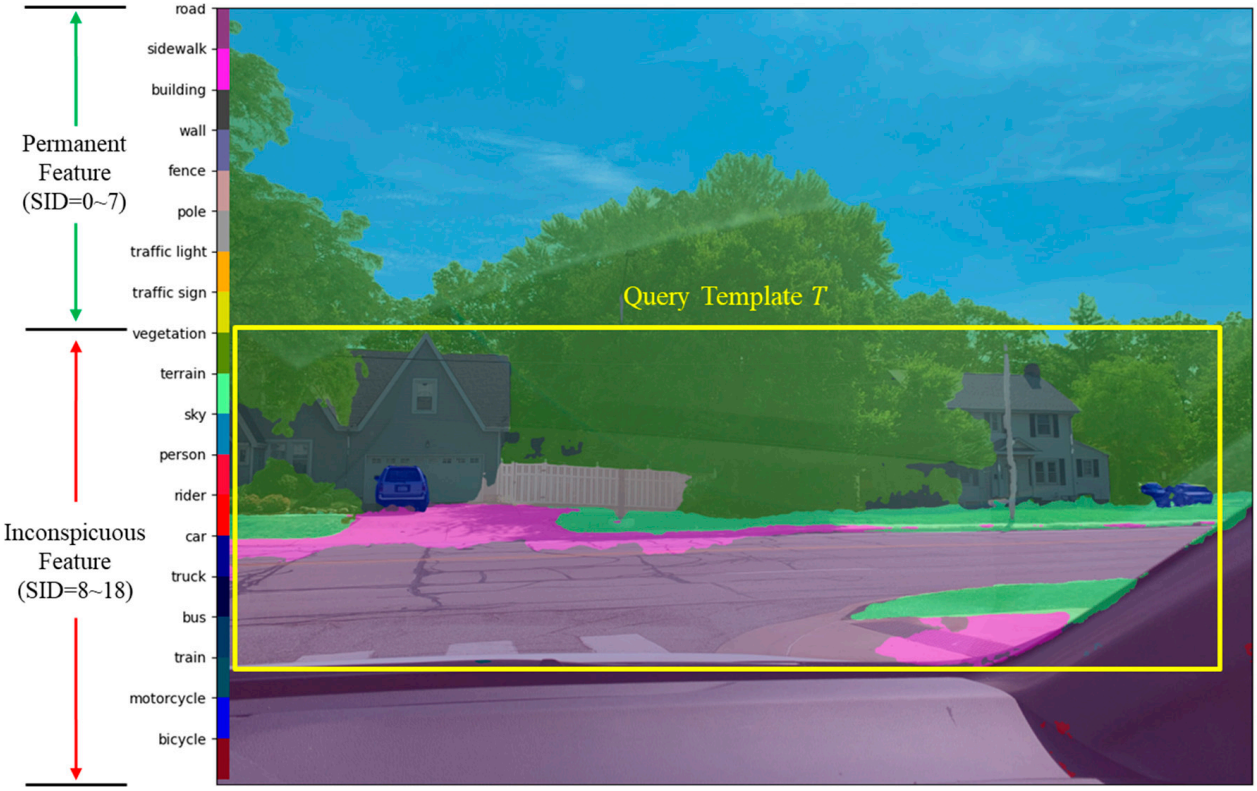


Figure 2. Colored permanent objects segmentation output and label legend of a query image captured using a smartphone inside a moving vehicle. Only the image patch, or the yellow rectangle denoted as T, encompassing all permanent objects (label ID = 0~7) will be used as a query template for searching the correspondence from the GSV images.

After segmenting the query image, we extract a rectangular template based on the label map. This template encompasses all pixels marked as belonging to permanent object classes, even if some of these areas are relatively small. This strategy allows us to keep the integrity and leverage all the information of the non-permanent objects simultaneously. When there are multiple clusters of permanent objects in an image, their minimum bounding rectangle is considered as the template. Figure 2 shows an example of the query template in one image, where the left part demonstrates the legend of the semantics labels of permanent and inconspicuous features and the yellow rectangle marks the template with all permanent features. It is worth noting that despite the inclusion of some non-permanent elements, such as trees and cars, the template ensures integrity rather than dividing it into several sub-templates. Although there may be pixels belonging to non-permanent objects in the template, our image-wise matching method is robust to handle such cases. This is because permanent objects play a more important role than non-permanent objects in template matching, which is demonstrated in further discussion from Section 5.1.

3.2. GSV Correspondence Finding with Template Matching

GSV has a vast global coverage and provides abundant street-level semantic data, which is employed in numerous applications. Current GSV images have an ultra-high resolution with equirectangular projection, and their field-of-view encompasses a 360° horizontal direction and a 180° vertical direction, suggesting there is likely to be substantial overlap with the query image. Our goal is, for a query template T, to find its correspondence P^*, i.e., the GSV patch that covers the same objects with T, in every GSV image in the

database. This is achieved using a learning-based template matching technology, Quality-Aware Template Matching (QATM) [19]. By using the VGG19 architecture [55], QATM achieves a dual-directional template matching with a likelihood function for evaluating matching quality. In QATM, deep feature maps are extracted for both the template query image and the GSV reference image, enabling the discovery of templates within the reference image and vice versa. Compared with the traditional template matching method that calculates the normalized cross-correlation or sum-of-squared differences of each original pixel [45–47], the deep features used in QATM are more robust when the transformation between the query and GSV images is non-rigid, such as stretching or shrinking differently in different directions. Also, the deep features highlight attention areas that are typically objects of interest and can avoid the effects of background variations. In addition, QATM employs a likelihood function which gives a soft ranking for all possible matchings with a learnable parameter. This soft similarity measurement is more robust to distortion than the widely used measurements NCC and SSD. Consequently, this technique is versatile enough to address a range of matching scenarios, encompassing 1-to-1, 1-to-many, many-to-many, and no-matching situations [19].

In our workflow, the process of QATM is formulated as follows. Based on the description of QATM, feature representations or maps of the template T and the GSV image are inferred by the pre-trained VGG19 [55]. Here T refers to the feature maps of the template. A set of patches with the same size as T are extracted in the feature representations of the GSV image, denoted as P. The patches P are compared with the template T. In the matching process, tiles of the same size (5×5 in our experiments) are obtained from P and T using a sliding window with a stride of 1 pixel, represented as p for P and t for T, respectively. For each tile t of T, the QATM algorithm compares it with each p in P. P^* is the patch that best matches the template T, and its tiles have the highest overall match quality. The function of assessing the template matching is shown in Equation (1).

$$P^* = QATM(T,\ P) = arg\ \max_{P}\left\{\sum\nolimits_{p\in P} \max\{\Theta(p,t)|t \in T\}\right\} \tag{1}$$

The function $\Theta(p,t) = L(t|p)\cdot L(p|t)$ defines the matching quality between (p,t). As shown in Equation (2a,b)

$$L(t|p) = \frac{\exp\{\alpha\cdot\rho(f_t,\ f_p)\}}{\sum_{t'\in T}\exp\{\alpha\cdot\rho(f_{t'},\ f_p)\}} \tag{2a}$$

$$L(p|t) = \frac{\exp\{\alpha\cdot\rho(f_p,\ f_t)\}}{\sum_{p'\in P}\exp\left\{\alpha\cdot\rho\left(f_{p'},\ f_t\right)\right\}} \tag{2b}$$

$L(t|p)$ defines the likelihood function that a tile t is matched, where f_t and f_p respectively denote the feature representations of tile t and p; t' and p' respectively represent all tiles of template T and GSV patch P. Utilize the identical location and dimensions as t in the template and p in the GSV image, f_t and f_p denote feature representations extracted from the template and GSV image using the pre-trained VGG19 [55], and are flattened to a 1×25 feature vector. Suggested as from 12.5 to 33.7 [19], α is an empirical parameter used for making the matched and unmatched patches balanced, and ρ is the cosine similarity function: $\rho(f_t,\ f_p) = \frac{f_t \cdot f_p}{\|f_t\|\left\|f_p\right\|}$.

It should be noted that since the query images and GSV images are likely collected with different sensors under considerably different environmental conditions, significant scale differences may exist, and a scaling factor should be determined to ensure that the template can be correctly matched to the GSV images. The determination of this factor involves testing scaling factors ranging from 1 to 10, and the selected number corresponds to the one yielding the highest template matching score using the QATM method. In our experiments with three datasets, this process identifies a scaling factor of 2.

3.3. Image-Wise and Block-Wise Similarity Computation

After locating the best-matched GSV patch P^* for each template T, the image-wise similarity will be calculated. To do so, the pretrained CLIP model is first applied to generate deep feature vectors from the template T and GSV patch P^*. The CLIP model is designed to measure the similarity between a pair of texts or images and is pretrained on a large dataset of image (or text) pairs using OpenAI [20]. Hence, CLIP is an appropriate choice for our task of measuring the similarity of image pairs. Furthermore, it demonstrates a strong performance even without the need for fine-tuning, rendering it well-suited for image matching, where it can be employed as a feature extractor [20]. In this study we choose to use the CLIP model released on Hugging Face (https://huggingface.co/sentence-transformers/clip-ViT-B-32, accessed on 22 June 2021). The cosine similarity is then used to measure the image-wise similarity between the CLIP feature vectors extracted from the template T and its matched GSV patch P^*. To consider the performance of the template matching in the previous Section 3.2, the QATM score is considered as the weight. As such, the image-wise similarity between the query image and the GSV image can be denoted in Equation (3):

$$\phi = QATM(T,\ P^*) \times \rho(CLIP(T), CLIP(P^*)) \tag{3}$$

where ρ is the cosine similarity; the function $CLIP$ is the image encoder using pre-trained CLIP model to represent the image patch into a 1×512 feature vector; T is the query template containing permanent objects; and P^* represents the best-matched image patch from the GSV image. To consider the performance of the template matching in the previous step, the QATM score is imported and considered as the weight, a larger QATM score means a better template matching. Cosine similarity is employed in this context because the angles between the feature vectors $CLIP(T)$ and $CLIP(P^*)$ are more relevant than their magnitudes. Cosine similarity displays a higher resilience on feature vectors with extreme values since it highlights the concordance of feature orientations by normalizing the feature vectors [56,57]. On the contrary, Euclidean distance tends to be significantly affected by such disturbances. Additionally, cosine similarity performs admirably in high-dimensional spaces, rendering it well-suited for numerous image feature extraction techniques [58].

There might be inaccurate matching outcomes with the highest image-wise similarity due to the presence of similar objects. Nevertheless, addressing this mismatch can be enhanced by incorporating neighboring images to generate block-wise similarity, building upon the image-wise similarity and thereby improving overall robustness. To do so, the query images and GSV images are grouped into non-overlap blocks where there are no duplicate images in different blocks. That means, a query block includes several sequentially captured query images, and these query images are excluded while forming the following query blocks, the same strategy is applied to the GSV block grouping. Specifically, this grouping strategy starts with grouping n neighboring query images into a block by their acquisition time. Similarly, starting from the most northwestern GSV image, every m spatially closest GSVs are grouped to a GSV block.

For a pair of a query and GSV blocks, the image-wise similarities of n query images and m GSV images within the block, represented as ϕ_{kl} ($k \in \{1, \cdots, n\}$, $l \in \{1, \cdots, m\}$), form a n by m image-wise similarity matrix Φ as below:

$$\Phi = \begin{bmatrix} \phi_{11} & \cdots & \phi_{1m} \\ \vdots & \ddots & \vdots \\ \phi_{n1} & \cdots & \phi_{nm} \end{bmatrix} \tag{4}$$

In the similarity matrix Φ the highest value of the kth row, denoted as $\max(\{\phi_{k1}, \ldots, \phi_{km}\})$, represents the similarity between the kth query template in the query block and the best-matched GSV patch within the GSV block. To simultaneously consider the combined influence of each query image within the query block, the average of the highest values of every row of the similarity matrix Φ is computed to represent the block-

wise similarity. As such, the block-wise similarity Π between each pair of query block and GSV block is denoted as Equation (5)

$$\Pi = \frac{\sum_{k=1}^{n} \max(\{\phi_{k1}, \ldots, \phi_{km}\})}{n} \quad (5)$$

By calculating the block-wise similarity over all GSV blocks, the best-matched GSV block for a query block can be found. In practice, for each query block we keep several top best-matched GSV blocks for the subsequent pose estimation process to enhance its performance.

3.4. Pose Estimation of the Query Image

To estimate the pose of query images in a query block, we use the GSV images in the corresponding GSV block as reference. It should be noted that in the matched query and GSV blocks, the query images are monocular images, while the corresponding GSV images are panoramas. Such differences make it challenging to achieve accurate pose estimates. To address this challenge, a rectilinear project function (https://github.com/sunset1995/py360convert, accessed on 23 January 2019) is employed to project the best-matched GSV patch P^*, i.e., a patch of the panorama (spherical), and obtain the corresponding perspective image, where the details are introduced in [59]. To perform the projection, we need to estimate the viewing angles and know the fields of view of the GSV patch. Let x_c and y_c be the pixel coordinates of the center of the best-matched patch P^* in the GSV image. The horizontal and vertical viewing angles are $u_P = 360^\circ \cdot \frac{x_c}{W}$, $v_P = 180^\circ \cdot \frac{y_c}{H}$, where W and H represent the image size of the GSV image. Since the projected GSV patch shares a similar capture scene as the query image, we presume that both the fields of view and image size of the projected GSV patch align with those of the query image. Consequently, the focal length of the projected GSV patch is identical to the focal length of the query image. Taking the geometric easting direction as x-axis, the rotation angles of the projective patch can be formulated as: $\omega = 90^\circ - 180^\circ \cdot \frac{y_c}{H}$, $\varphi = 0^\circ$, $\kappa = A + 360^\circ \cdot \frac{x_c}{W} - 180^\circ$, where ω, φ, κ respectively represent the rotation angles along X, Y, and Z axis, and A is the northing angle of the GSV image center. Together with such calculated rotation angles, the 3D location of the reference GSV image is assigned to the projected GSV patch as its exterior orientation parameters.

For a query block, we select the top three best-matched GSV blocks for the subsequent photogrammetric triangulation. As a nominal practice, both the query block size and GSV block size are chosen as three. The images in the query block and the projected GSV images in the selected GSV blocks are used as the input for photogrammetric triangulation with Agisoft Metashape 2.0. This photogrammetric triangulation procedure starts with routine tie point extraction and matching, followed by a bundle adjustment, in which the known location and orientation of the GSV images are essentially used as reference and treated as direct observations with 1 m location uncertainty and 10° orientation uncertainty. The interior orientation of the query images (saved in the EXIF metadata) and the projected GSV patches are treated as known in the bundle adjustment. The pose of the query image and the 3D coordinates of the tie points are the results of the bundle adjustment.

4. Experimental Datasets and Evaluation

4.1. Datasets

A total of three datasets were employed in this study to address the variations in lighting, temporal consistency, and long-term visual changes in order to demonstrate the effectiveness and robustness of our proposed workflow, with each having mobile images and GSV panoramic images. Figure 3 shows several examples of the three datasets. It should be noted that all the GSV images, captured through Ricoh Theta S, were collected using a commercial application: Street View Download 360 (https://svd360.istreetview.com/, accessed on 3 July 2023). The first dataset is a subset of the University of Central Florida (UCF) mobile car-based dataset [60], which originally consisted of 62,058 high-

resolution GSV images captured from multiple side views and one upward view. We specifically chose a subset from the UCF dataset located in the central area of Pittsburgh, PA. This subset comprises 300 query images distributed across a 0.6 km^2 area. Each query image is accompanied by two sequentially captured images, creating a query block with a size of 3. This region shares similarities with permanent structures in other sprawling cities. Consequently, the matching results from the UCF dataset serve as an illustrative example of how our workflow performs in a large city context, emphasizing the importance of considering the approximate areas. The second dataset, the Málaga Streetview Challenge dataset (MSV) [61–63], documents urban changes in Málaga City, Spain, over a span of six years (2014–2020). This dataset was collected using a vehicle equipped with various sensors, including a stereo camera (Bumblebee2, Teledyne FLIR LLC, Wilsonville, OR, USA) and five laser scanners. In total, it includes 436 monocular images and 3411 GSV images, covering an urban area of nearly 0.8 km^2. Finally, it should be noted that the query images in the UCF and MSV datasets only provide latitude and longitude information, lacking altitude and camera orientation data.

Figure 3. Query image and GSV panoramic image examples of the used datasets.

While the above two datasets are public ones, the third one is our recent collection. The Purdue University Vision-Based Navigation dataset (PUVBN) was collected in a moving vehicle on the Purdue University campus. It consists of geo-referenced smartphone images that are to be used as query images for visual localization tasks. PUVBN images were collected using off-the-shelf smartphones (iPhone 13 Pro, Apple Computer Inc., Cupertino, CA, USA) mounted in a moving vehicle at a fixed sampling rate of 30 frames per minute along a pre-planned 30-min driving route. The total route length is about 10 km, covering the same area of 1 km^2 with a driving speed range from 20 km/h to 65 km/h. During image collection, the horizontal location (latitude, longitude), altitude, and orientation of the sensor, as well as their corresponding accuracy, were simultaneously recorded using a commercial iOS application SensorLog with a 30 Hz sampling rate (https://sensorlog.berndthomas.net/, accessed on 14 March 2023). Based on the SensorLog record, the accuracy of the horizontal location is approximately 3.3 m, the vertical accuracy is about 4.8 m, and the orientation accuracy is about 2.4°. In total, PUVBN has 714 smartphone images and 1820 corresponding reference GSV images.

It should be noted that baseline data for the PUVBN dataset were also gathered for potential alternative, broader applications. These applications encompass tasks like building model reconstruction, mobile 3D mapping, autonomous driving, and many more. Other collected baseline data consist of the following components: 85 geo-referenced 3D building models, a layer of road data from OpenStreetMap, digital elevation models at a resolution of 0.76 m (representing bare ground), digital surface models at a resolution of 1.52 m, and a high-resolution mosaic orthoimage at 0.3 m resolution.

Table 1 summarizes the specifications of the datasets utilized to test and evaluate our proposed workflow. Notably, the GSVs included in the databases were collected between 2018 and 2019, guaranteeing a meter-level positioning accuracy [7]. It is important to acknowledge that this timeframe differs from that of the query images, resulting in a temporal gap. Consequently, changes in lighting conditions and urban settings can occur, effectively emulating the challenges encountered in practical visual localization scenarios.

Table 1. Summary of the UCF, MSV, and PUVBN datasets.

Datasets	Coverage	Query Images		GSV Images		Ratio
		Counts	Date	Counts	Date	Query:GSV
UCF	0.8 km^2	300	2012–2014	1291	2018–2019	1:4.3
MSV	0.6 km^2	436	2014–2020	3411	2018–2019	1:7.8
PUVBN	1.0 km^2	714	2022	1820	2018–2019	1:2.5

Figure 4 lists the spatial distribution of the used datasets. The GSVs obtained along the routes cover areas of approximately 0.6 km^2, 0.8 km^2, and 1 km^2 for the UCF, MSV, and PUVBN datasets, respectively. This calculation assumes that each GSV can observe a circle with a radius of 50 m, i.e., a coverage area around 7850 m^2. It should be noted that the count of the reference images in the three datasets ranges from 1291 to 3411, meaning that the search space of a query image for image matching is 1000 to 3000 times more. The ratio of query images to the GSV images is also listed to illustrate how large the reference GSV dataset is to cover the query images for reliable matching results. Additionally, the average distance between two images is approximately 10 m for the GSV images and 20 m for the query images. The spatial distribution of GSV images is denser than query images, which can ensure that each query block has a corresponding GSV block nearby. As for the reference pose for evaluation, the two public datasets (i.e., the UCF and MSV datasets) only provide horizontal locations (latitude and longitude) recorded by GNSS, while the PUVBN dataset not only records the latitude, longitude, and altitude, but also the orientation information of the mobile images. The goal of our effort is to determine the pose of the query images by using the corresponding GSV images to achieve a quality equivalent to the GNSS and IMU records.

Figure 4. Spatial distribution of the three test datasets. Query images are labeled red circles, and GSV images are labeled green triangles. The total vehicle travel length and the total road length are respectively 3 and 6; 5 and 8; and 8 and 10 km for the UCF, MSV, and PUVBN datasets.

4.2. Experimental Results

We evaluated the effectiveness of our approach by employing a range of performance metrics, including matching rates and positioning accuracy represented by the mean and standard deviation of the Euclidean distances between the estimated locations and the GNSS recorded locations of the query images. As noted earlier, when dealing with query images of the UCF and MSV datasets, the reference location of the query images only includes latitude and longitude, with no altitude information available. Consequently,

we could only assess the estimated horizontal positions of the query images. To be more specific, for each query block, we select the top three GSV blocks with the highest block-wise similarity. These selections are then utilized in the photogrammetric triangulation process to estimate the pose of the query images. Before processing in the Metashape 2.0 software, the GSV patches corresponding to the query templates are projected as perspective images. The estimated poses of the query images through bundle adjustment are then compared with the corresponding reference records from GNSS and IMU. Since the query images in UCF and MSV datasets lack reference records for altitude and orientation information, our evaluation focused solely on 2D positioning accuracy at this time. But for the newly proposed PUVBN dataset, both 3D positioning and orientation error are evaluated.

As indicated in Table 2, employing solely image-wise similarity for matching and estimating the pose of query images, where each query image and its top nine matched GSV images are input into the photogrammetric triangulation, yields a mean horizontal positioning accuracy of 2.54 to 4.51 m, accompanied by a standard deviation ranging between 7.97 to 12.75 m across the testing datasets. In contrast, employing the block-wise matching strategy, where each query block includes three query images and three best-matched GSV blocks with each consisting of three GSV images, enhances the horizontal positioning accuracy, resulting in an improved mean error of 1.09 to 2.12 m with a smaller standard deviation from 5.77 to 9.01 m. The distribution of the horizontal positioning accuracy is visualized and compared in Figure 5. It is worth noting that, with image-wise matching, the maximum horizontal positioning accuracy can vary from 50 to 120 m for the UCF, MSV, and PUBNM datasets. However, the adoption of the block-wise matching approach within the photogrammetric triangulation leads to more robust pose estimation results, yielding a maximum horizontal positioning accuracy in the range of 40 to 90 m. This is because image-wise matching results may contain a few mismatches that are not the closest one to the query image, even though high similarity is achieved. This would provide fewer valid correspondences in the photogrammetric triangulation, leading to a less reliable positioning quality. In comparison, the block-wise matching results in robust and more accurate matches, as the GSVs in each block observe the same objects in the query images multiple times and mitigate the influence of mismatches. The outcomes also demonstrate that pose estimation using the block-wise matching strategy attains positioning accuracy comparable to that of GSV. This alignment is notable, given that the positioning accuracy of GSV images is approximately one meter, as detailed in Section 4.1.

Table 2. Mean and standard deviation of the horizontal positioning for image-wise and block-wise matching (block size: 3) with reference to the GNSS records for the three testing datasets.

	UCF	**MSV**	**PUVBN**
Image-wise	4.51 ± 12.75 m	2.79 ± 9.33 m	2.54 ± 7.97 m
Block-wise	2.12 ± 9.01 m	1.35 ± 7.29 m	1.09 ± 5.77 m

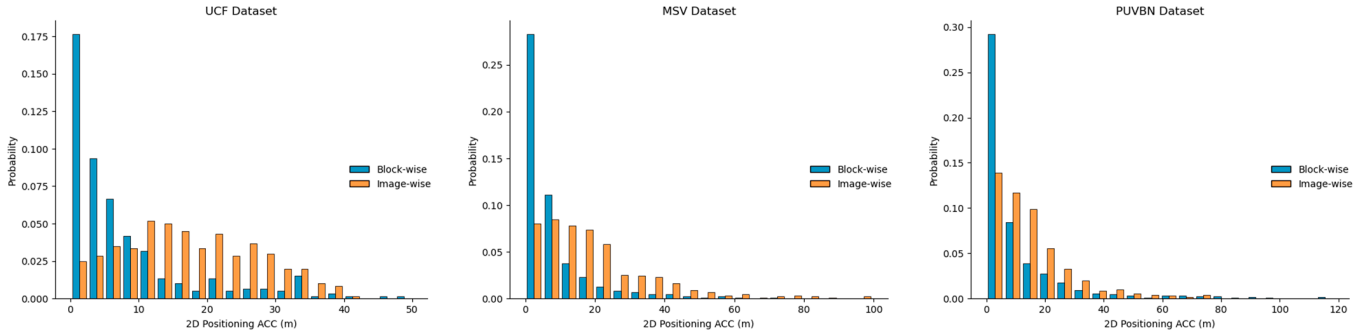


Figure 5. Distribution of horizontal positioning accuracy for image-wise and block-wise matching on the three testing datasets.

In contrast to the UCF and MSV datasets, the PUVBN dataset captures orientation and GNSS altitude information concurrently during the acquisition of the query images. Subsequent assessments reveal a vertical positioning accuracy of 1.85 m and an orientation accuracy of 5.01° when utilizing image-wise matching results for pose estimation. When employing block-wise matching results, the vertical positioning accuracy is improved to 1.24 m, while the orientation accuracy remains nearly unchanged at 4.88°.

Aside from the absolute evaluation metric, which is the horizontal positioning accuracy, we also assess our matching performance using relative metrics, namely Top-1 Recall @D and Recall @N. These metrics gauge the matching rates by considering the distance between a GSV image block and a query image block. If the query image is close to the GSV image, the same permanent objects should be captured, and the query image should match the GSV image with high similarity. The Top-1 Recall @D evaluates the level of correspondence by assessing the distance between the location of the highest-ranked GSV image and the query image. In the context of block-wise matching, we determine the distance by calculating the centroid of the coordinates for both the query images within the query block and the GSV images within the involved GSV block. Matches are deemed satisfactory if the distances between block centroids fall below a specified threshold. As illustrated in Figure 6, we calculate the percentage of good matches as we increase the threshold from 5 to 150 m. In contrast, Recall @N is a metric used to determine the percentage of well-matched queries, those with less than a 50-m distance error, concerning N returned candidates. By calculating Recall @N, we aim to demonstrate that the block-wise approach can yield more robust matches, leading to more precise pose estimation, thanks to the continuous spatial information provided by GSV images. The results of the Recall @N metric illustrate that our method can substantially decrease the search space size, reducing it from a range of 10 km to less than 100 m in relation to the reference location of the query image.

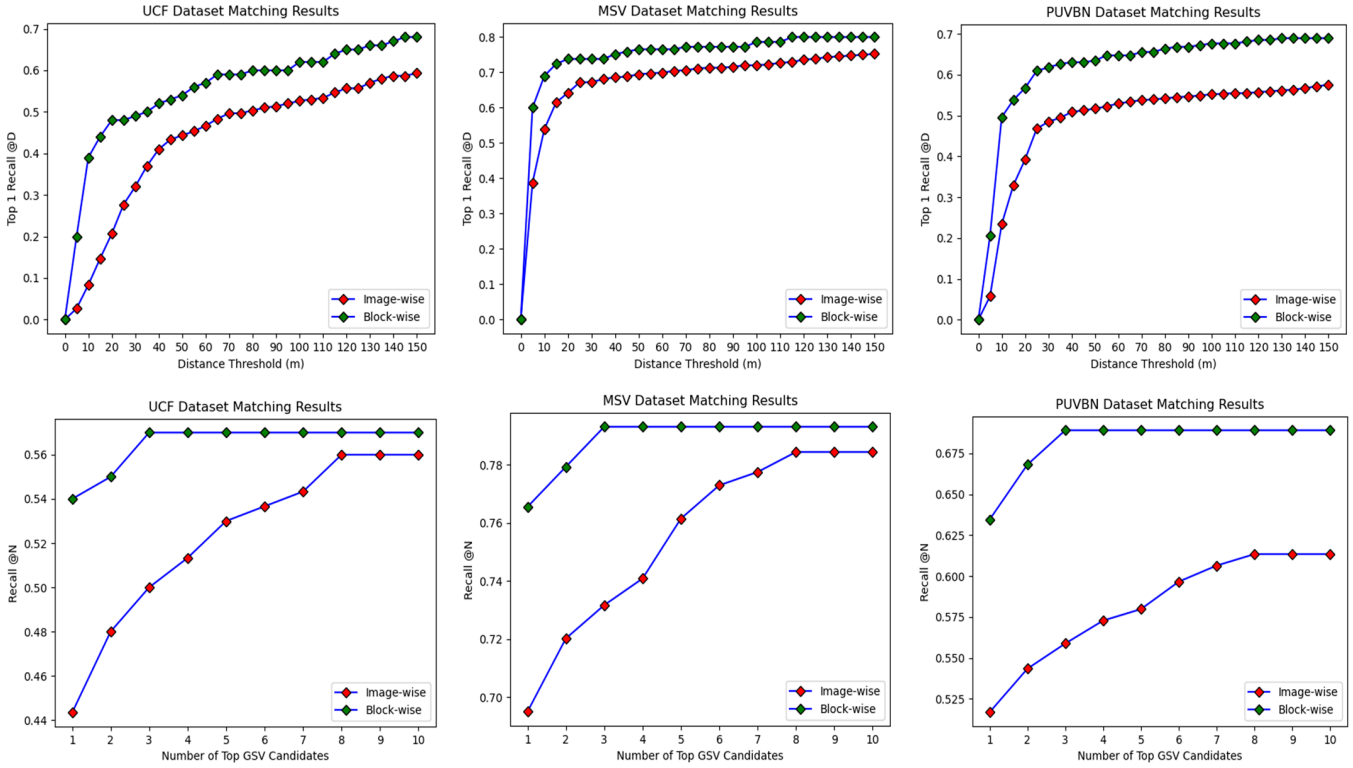
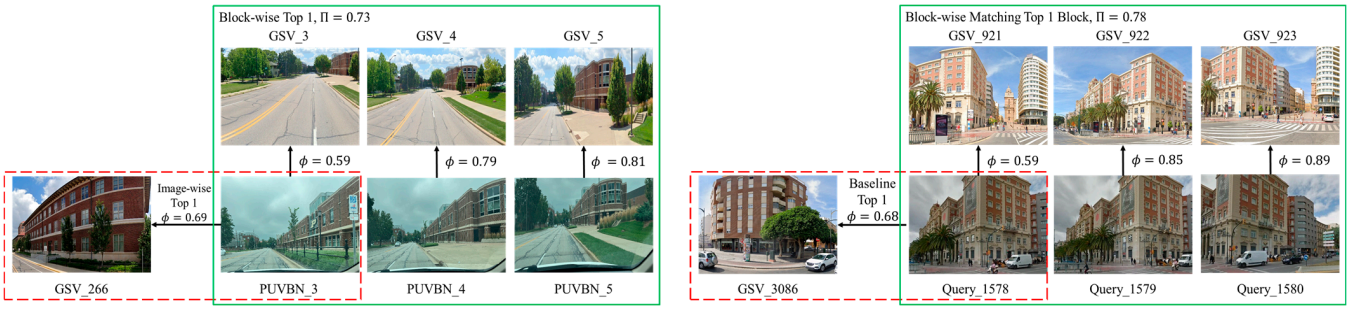


Figure 6. Performance of the proposed workflow in terms of Top-1 Recall @D (1st row) and Recall @N (2nd row) of image-wise (red diamond) and block-wise (green diamond) matching evaluation metrics on the testing datasets. From left to right: UCF, MSV, and PUVBN.

To demonstrate the superiority of our workflow, Figure 6 illustrates the results of both image-wise and block-wise similarity comparisons, along with the evaluation metrics (Top 1 Recall @D and Recall @N). Concerning image-wise matching, the UCF, MSV, and PUVBN datasets attain respectively Top 1 Recall @150 percentages of 59.33%, 75.29%, and 57.56%. These numbers represent the proportion of successful matches between query and GSV images or blocks within 150 m. However, when employing the block-wise matching strategy, the successful matching percentages increase to 68%, 80%, and 68.91%, respectively, showing significant improvements of 8.67%, 4.77%, and 11.4%, respectively. These enhancements in successful matching rates underscore the notable effectiveness of the block-wise matching approach.

The presented plots in Figure 6 demonstrate that utilizing a block-wise matching methodology yields GSV images in closer proximity to the query images, which indicates superior matching outcomes. By examining the recall values displayed in the plots, it becomes apparent that increasing the value of N leads to higher recall rates across all datasets. While block-wise matching achieves the highest Recall @N more quickly, the image-wise approach ultimately reaches a comparable Recall @N. This implies that the image-wise approach struggles to differentiate between true and false matches when their similarities are very close. However, this challenge can be resolved by introducing the block-wise similarity into the matching process.

From the significant improvement of the matching rate, it can be concluded that the incorrect image-wise matching result could be corrected by leveraging block-wise similarity. Figure 7 illustrates a series of visual examples from the PUVBN dataset where the image-wise method fails, but correct matches are retrieved using block-wise matching. For a specific query image (PUVBN_3) in Figure 7a, it is noticeable that GSV_266 receives the highest matching score ($\phi = 0.69$) in the image-wise matching (indicated by the red dashed rectangle), although it is not the correct match. However, the block-wise matching (represented by the green solid rectangle) produces a correct match with the highest block-wise similarity ($\Pi = 0.73$) by including the query image within a block and averaging the matching scores. It can be found that although the best-matched GSV image within the block of PUVBN_3 is GSV_3 with a 0.59 image-wise similarity, its neighboring matches: PUVBN_4 and GSV_4, and PUVBN_5 and GSV_5 result in the Top1 block-wise similarity (0.73). This example demonstrates that the proposed workflow enhances the matching results by considering the continuity of neighboring images. Another example on MSV dataset is shown in Figure 7b. Similarly, image-wise matching conducted on MSV_1578 gives a wrong match GSV_3086 with the highest score ($\phi = 0.68$). However, correct matching GSV_921 is found with a block-wise similarity of ($\Pi = 0.78$) through the block-wise matching strategy.



(**a**) PUVBN example (**b**) MSV example

Figure 7. Two visual comparisons on PUVBN and MSV datasets of image-wise versus block-wise matching methods. The incorrect image-wise matching result is shown in the red dash rectangle, while the corrected result by block-wise matching is shown in green solid rectangle. Only the three GSV images from the best-matched block are shown.

5. Discussion

5.1. Significance of Permanent Objects

Section 3.1 illustrates the identification of different types of objects based on their temporal characteristics. Time-invariant objects like buildings, traffic signs, and light poles are recognized as permanent objects, whereas objects like vegetation, cars, pedestrians, and clouds that change over time and with weather conditions are categorized as inconspicuous objects. Since permanent objects (label ID from 0 to 7) play a significant role in the matching process, it becomes necessary to use a metric to describe the significances (or contributions) of different kinds of permanent objects. In this paper, we employ the Pearson correlation coefficients (represented as r) as a quantitative measure for each enduring object. This measure is based on the correlation between the object's percentage in the query image and the reciprocal distance of the query image to the centroid of the best-matched GSV block.

The permanent object percentage is the ratio of the number of permanent object pixels to the total number of pixels in the image. A positive r occurs when the query images with more permanent objects are matched to the GSV blocks closed to the query images, i.e., larger inversed distance. Conversely, a negative r means that query images with more permanent objects are matched to the distant GSV blocks. When the absolute value of r is close to zero, it implies that the object has little impact on the matching process. On the other hand, if the absolute value of r is significantly different from zero, it indicates that the object has a significant influence on the matching process, i.e., more significance. The Pearson correlation coefficients are calculated for each permanent object in all three datasets, and the results can be found in Table 3.

Table 3. Pearson correlation coefficients (r) between object percentage with the reciprocal distance of the query image to the centroid of the best-matched GSV block. Numerically insignificant values are indicated in gray, while the most significant category, such as 'buildings', is labeled in bold.

Datasets	r (Pearson Correlation Coefficients)								
	Road	Sidewalk	Building	Wall	Fence	Pole	Traffic Light	Traffic Sign	Total
UCF	−0.063	−0.0186	**0.184**	−0.058	−0.103	0.026	−0.021	0.015	0.153
MSV	0.122	−0.002	**0.154**	0.172	0.116	−0.187	−0.016	−0.052	0.217
PUVBN	0.142	0.072	**0.234**	0.064	0.009	0.132	0.014	0.080	0.281

It should be noted that in Table 3, any significance lacking statistical significance at a 95% confidence level is shown in gray. Considering the overall significance of permanent objects, it can be inferred that a greater number of permanent objects result in a more accurate matching. Among all the permanent objects, buildings hold the highest significance (0.15~0.23) and labeled bold in Table 3, as evidenced by their high values of r. In other words, accurately locating the query images is more likely when there are more buildings present. That may be because that buildings are often larger than other permanent objects and easily distinguishable due to their rich textures. As a comparison, other permanent objects such as wall, fence, pole, traffic light and sign show less significance since they are usually too similar and small to make a difference.

5.2. Size of the GSV Block

While the block-wise matching is superior to image-wise matching, its performance may be dependent on the block size. To investigate the effect of GSV block size, we conduct additional tests with the PUVBN dataset. In our previous block-wise matching experiment, we set the block size to 3, which means that each query block and GSV block comprise three consecutive images. Given that the average distances among query images and among GSV images are approximately 20 and 10 m, respectively, each query and GSV block covers a route of 60 and 30 m respectively. To further investigate the impact of the block size, we maintained the size of the query block while increasing the GSV block size to 6 and 9. It should be noted that we still choose three best-matched GSV blocks, i.e., 18 or 27 GSV

images, for pose estimation using the photogrammetric triangulation process described in Section 3.4. The corresponding results, including horizontal positioning, vertical positioning, and orientation, are presented in Table 4. It is evident that despite doubling or even tripling the block size, there is no significant improvement (from 1.09 m to 1.21 m) in the overall pose estimation accuracy. However, it is worth noting that the pose estimation exhibits better stability with reduced standard deviations in the horizontal direction (from 5.77 m to 2.60 m) and the vertical direction (from 3.22 m to 1.22 m). Regarding orientation accuracy, the angle error improved from 4.88° to 3.98° with smaller deviations (from 3.21° to 2.67°). These results indicate that increasing the GSV block size does not lead to a substantial enhancement in pose estimation accuracy, but it does result in improved robustness and stability by significantly reducing the number of large mis-matching outcomes. The more GSVs in the block, the more probability of finding the GSV that best matches the query image. Nevertheless, the enhanced robustness is accompanied by a greater demand for computational resources and time, averagely ranging from 0.18 s to 0.54 s when changing the block size from 3 to 9.

Table 4. Horizontal, vertical, and orientation accuracy of block-wise matching with different block sizes on the PUVBN dataset in terms of mean and standard deviation.

GSV Block Size	Horizontal	Vertical	Orientation
3	1.09 ± 5.77 m	1.24 ± 3.22 m	$4.88 \pm 3.21°$
6	1.18 ± 3.91 m	1.33 ± 1.55 m	$5.02 \pm 3.19°$
9	1.21 ± 2.60 m	1.17 ± 1.20 m	$4.18 \pm 2.67°$

6. Conclusions

We proposed a novel approach for visual localization in urban areas. The solution is based on state-of-the-art machine learning techniques. The use of SegFormer assures the generation of a high-quality query template consisting of permanent objects from the query image, while the adoption of QATM allows us to reliably obtain the corresponding GSV patches from the GSV images. The pre-trained CLIP model was employed to extract deep feature representation for the query template and corresponding GSV patches for the image-wise similarity calculation. In addition, we took advantage of the spatial or temporal continuity of sequential images to provide a reliable image query workflow. Our method utilizes a block-wise matching strategy that involves grouping the query and reference GSV images into blocks, and block-wise similarity computation.

In comparison to image-wise matching, our approach demonstrates an average improvement of 10% in the successful matching rate. Specifically, we could achieve a matching rate of 68%, 80%, and 68.91% for the UCF, MSV, and PUVBN datasets, respectively. A notable aspect of our approach is its proficiency in accurately identifying the correct match from a set of similar matches, achieved by incorporating considerations of spatial continuity.

With our overarching objective in sight, we leverage the three best-matched GSV image blocks to determine the poses of images in a query block through photogrammetric triangulation. Considering the pose estimation results with the PUVBN dataset, when contrasted with the GNSS records of smartphone images, our approach yields a horizontal and vertical positioning accuracy of 1.09 and 1.24 m, accompanied by a standard deviation from 5.77 and 3.22 m. In addition, the orientation accuracy can be achieved around 4.88° with a 3.21° standard deviation. Given the large search space of possibly matched GSV images, i.e., 1820 images spreading over an area of up to 1 km^2 and taking into account the quality of the geo-referenced information associated with GSV images, the achieved meter-level accuracy underscores the effectiveness and dependability of our proposed block-wise matching strategy in comparison to the original meter-level positioning accuracy of GSV.

As for computational cost, our workflow was evaluated on a computer featuring an Intel(R) Xeon(R) E-2186G CPU and an NVIDIA GeForce GTX 1080 GPU. The typical computational time required for a complete search of one single image within a reference dataset comprising 1820 images is approximately one minute and can be improved to a

stable time around a few seconds after optimizing the workflow with parallel computation or using the first few matching results to narrow down the search space for the following matching procedures. An acceptable level of precision falls within a few meters, aligning well with the standard of GNSS measurements in dynamic scenarios or instances where no GNSS data are available. This is particularly relevant in applications relying on pre-stored street view image databases and real-time captured images, such as vehicle location-based services, tracking, and smart city applications.

For future research, we recommend performing experiments wherein a single GSV block is chosen with an increased block size. This block should encompass more neighboring GSVs and offer broader spatial coverage, aiming to enhance the robustness of block-wise matching and improve pose estimation. Additionally, we propose the integration of image patches depicting consistent objects observed in query images from the designated query block. This fusion is intended to harness the complete potential of image continuity, thereby achieving heightened accuracy in matching.

Author Contributions: Conceptualization, J.S., Z.L. and J.A.; methodology, Z.L., J.S. and J.A.; software, Z.L.; validation, Z.L., S.L. and J.S.; formal analysis, Z.L., J.S. and S.L.; investigation, Z.L. and S.L.; resources, Z.L. and J.S.; data curation, Z.L.; writing—Z.L., J.S. and S.L.; writing—review and editing, Z.L., J.S., S.L. and J.A.; visualization, Z.L. and J.S.; supervision, J.S.; project administration, J.S.; funding acquisition, J.S. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: Data are contained within the article.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Usman, M.; Asghar, M.R.; Ansari, I.S.; Granelli, F.; Qaraqe, K.A. Technologies and Solutions for Location-Based Services in Smart Cities: Past, Present, and Future. *IEEE Access* **2018**, *6*, 22240–22248. [CrossRef]
2. Burgard, W.; Brock, O.; Stachniss, C. Map-Based Precision Vehicle Localization in Urban Environments. In *Robotics: Science and Systems III*; MIT Press: Cambridge, MA, USA, 2008; pp. 121–128. ISBN 9780262255868.
3. Xiao, Z.; Yang, D.; Wen, T.; Jiang, K.; Yan, R. Monocular Localization with Vector HD Map (MLVHM): A Low-Cost Method for Commercial IVs. *Sensors* **2020**, *20*, 1870. [CrossRef]
4. Agarwal, P.; Burgard, W.; Spinello, L. Metric Localization Using Google Street View. In Proceedings of the 2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Hamburg, Germany, 28 September–3 October 2015.
5. Pauls, J.-H.; Petek, K.; Poggenhans, F.; Stiller, C. Monocular Localization in HD Maps by Combining Semantic Segmentation and Distance Transform. In Proceedings of the 2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Las Vegas, NV, USA, 24 October 2020–24 January 2021; pp. 4595–4601.
6. Stenborg, E.; Toft, C.; Hammarstrand, L. Long-Term Visual Localization Using Semantically Segmented Images. In Proceedings of the 2018 IEEE International Conference on Robotics and Automation (ICRA), Brisbane, Australia, 21–25 May 2018; pp. 6484–6490.
7. Zamir, A.R.; Shah, M. Accurate Image Localization Based on Google Maps Street View. In Proceedings of the Computer Vision—ECCV 2010, Heraklion, Greece, 5–11 September 2010; pp. 255–268.
8. Qu, X.; Soheilian, B.; Paparoditis, N. Vehicle Localization Using Mono-Camera and Geo-Referenced Traffic Signs. In Proceedings of the 2015 IEEE Intelligent Vehicles Symposium (IV), Seoul, Republic of Korea, 28 June–1 July 2015; pp. 605–610.
9. Senlet, T.; Elgammal, A. A Framework for Global Vehicle Localization Using Stereo Images and Satellite and Road Maps. In Proceedings of the 2011 IEEE International Conference on Computer Vision Workshops (ICCV Workshops), Barcelona, Spain, 6–13 November 2011; pp. 2034–2041.
10. De Paula Veronese, L.; de Aguiar, E.; Nascimento, R.C.; Guivant, J.; Auat Cheein, F.A.; De Souza, A.F.; Oliveira-Santos, T. Re-Emission and Satellite Aerial Maps Applied to Vehicle Localization on Urban Environments. In Proceedings of the 2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Hamburg, Germany, 28 September–3 October 2015; pp. 4285–4290.
11. Chu, H.; Mei, H.; Bansal, M.; Walter, M.R. Accurate Vision-Based Vehicle Localization Using Satellite Imagery. *arXiv* **2015**, arXiv:1510.09171.
12. Dogruer, C.U.; Koku, B.; Dolen, M. Global Urban Localization of Outdoor Mobile Robots Using Satellite Images. In Proceedings of the 2008 IEEE/RSJ International Conference on Intelligent Robots and Systems, Nice, France, 22–26 September 2008; pp. 3927–3932.

13. Bresson, G.; Yu, L.; Joly, C.; Moutarde, F. Urban Localization with Street Views Using a Convolutional Neural Network for End-to-End Camera Pose Regression. In Proceedings of the 2019 IEEE Intelligent Vehicles Symposium (IV), Paris, France, 9–12 June 2019; pp. 1199–1204.
14. Gruen, A. Everything Moves: The Rapid Changes in Photogrammetry and Remote Sensing. *Geo Spat. Inf. Sci.* **2021**, *24*, 33–49. [CrossRef]
15. Zhang, W.; Kosecka, J. Image Based Localization in Urban Environments. In Proceedings of the Third International Symposium on 3D Data Processing, Visualization, and Transmission (3DPVT'06), Chapel Hill, NC, USA, 14–16 June 2006; pp. 33–40.
16. Yu, L.; Joly, C.; Bresson, G.; Moutarde, F. Improving Robustness of Monocular Urban Localization Using Augmented Street View. In Proceedings of the 2016 IEEE 19th International Conference on Intelligent Transportation Systems (ITSC), Rio de Janeiro, Brazil, 1–4 November 2016; pp. 513–519.
17. Yu, L.; Joly, C.; Bresson, G.; Moutarde, F. Monocular Urban Localization Using Street View. In Proceedings of the 2016 14th International Conference on Control, Automation, Robotics and Vision (ICARCV), Phuket, Thailan, 13–15 November 2016; pp. 1–6.
18. Xie, E.; Wang, W.; Yu, Z.; Anandkumar, A.; Alvarez, J.M.; Luo, P. SegFormer: Simple and Efficient Design for Semantic Segmentation with Transformers. In Proceedings of the Advances in Neural Information Processing Systems 34 (NeurIPS 2021), Virtual, 6–14 December 2021; pp. 12077–12090.
19. Cheng, J.; Wu, Y.; AbdAlmageed, W.; Natarajan, P. QATM: Quality-Aware Template Matching for Deep Learning. In Proceedings of the 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 15–20 June 2019.
20. Radford, A.; Kim, J.W.; Hallacy, C.; Ramesh, A.; Goh, G.; Agarwal, S.; Sastry, G.; Askell, A.; Mishkin, P.; Clark, J.; et al. Learning Transferable Visual Models From Natural Language Supervision. In Proceedings of the 38th International Conference on Machine Learning, Virtual, 18–24 July 2021; Meila, M., Zhang, T., Eds.; Volume 139, pp. 8748–8763.
21. Ali, N.; Bajwa, K.B.; Sablatnig, R.; Chatzichristofis, S.A.; Iqbal, Z.; Rashid, M.; Habib, H.A. A Novel Image Retrieval Based on Visual Words Integration of SIFT and SURF. *PLoS ONE* **2016**, *11*, e0157428. [CrossRef] [PubMed]
22. Rublee, E.; Rabaud, V.; Konolige, K.; Bradski, G. ORB: An Efficient Alternative to SIFT or SURF. In Proceedings of the 2011 International Conference on Computer Vision, Barcelona, Spain, 6–13 November 2011; pp. 2564–2571.
23. Lowe, D.G. Distinctive Image Features from Scale-Invariant Keypoints. *Int. J. Comput. Vis.* **2004**, *60*, 91–110. [CrossRef]
24. Bay, H.; Tuytelaars, T.; Van Gool, L. SURF: Speeded Up Robust Features. In Proceedings of the Computer Vision—ECCV 2006, Graz, Austria, 7–13 May 2006; pp. 404–417.
25. Karakasis, E.G.; Amanatiadis, A.; Gasteratos, A.; Chatzichristofis, S.A. Image Moment Invariants as Local Features for Content Based Image Retrieval Using the Bag-of-Visual-Words Model. *Pattern Recognit. Lett.* **2015**, *55*, 22–27. [CrossRef]
26. Jégou, H.; Perronnin, F.; Douze, M.; Sánchez, J.; Pérez, P.; Schmid, C. Aggregating Local Image Descriptors into Compact Codes. *IEEE Trans. Pattern Anal. Mach. Intell.* **2012**, *34*, 1704–1716. [CrossRef]
27. Torii, A.; Sivic, J.; Okutomi, M.; Pajdla, T. Visual Place Recognition with Repetitive Structures. *IEEE Trans. Pattern Anal. Mach. Intell.* **2015**, *37*, 2346–2359. [CrossRef] [PubMed]
28. Torii, A.; Arandjelović, R.; Sivic, J.; Okutomi, M.; Pajdla, T. 24/7 Place Recognition by View Synthesis. In Proceedings of the 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Boston, MA, USA, 7–12 June 2015; pp. 1808–1817.
29. Krizhevsky, A.; Sutskever, I.; Hinton, G.E. ImageNet Classification with Deep Convolutional Neural Networks. *Commun. ACM* **2017**, *60*, 84–90. [CrossRef]
30. Razavian, A.S.; Azizpour, H.; Sullivan, J.; Carlsson, S. CNN Features Off-the-Shelf: An Astounding Baseline for Recognition. In Proceedings of the 2014 IEEE Conference on Computer Vision and Pattern Recognition Workshops, Columbus, OH, USA, 23–28 June 2014; pp. 806–813.
31. Tolias, G.; Sicre, R.; Jégou, H. Particular Object Retrieval with Integral Max-Pooling of CNN Activations. *arXiv* **2015**, arXiv:1511.05879.
32. Jogin, M.; Mohana; Madhulika, M.S.; Divya, G.D.; Meghana, R.K.; Apoorva, S. Feature Extraction Using Convolution Neural Networks (CNN) and Deep Learning. In Proceedings of the 2018 3rd IEEE International Conference on Recent Trends in Electronics, Information & Communication Technology (RTEICT), Bangalore, India, 18–19 May 2018; pp. 2319–2323.
33. Liu, Y.H. Feature Extraction and Image Recognition with Convolutional Neural Networks. *J. Phys. Conf. Ser.* **2018**, *1087*, 062032. [CrossRef]
34. Chen, Z.; Jacobson, A.; Sünderhauf, N.; Upcroft, B.; Liu, L.; Shen, C.; Reid, I.; Milford, M. Deep Learning Features at Scale for Visual Place Recognition. In Proceedings of the 2017 IEEE International Conference on Robotics and Automation (ICRA), Singapore, 29 May–3 June 2017; pp. 3223–3230.
35. Arandjelovic, R.; Gronat, P.; Torii, A.; Pajdla, T.; Sivic, J. NetVLAD: CNN Architecture for Weakly Supervised Place Recognition. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016; pp. 5297–5307.
36. Cinaroglu, I.; Bastanlar, Y. Long-Term Image-Based Vehicle Localization Improved with Learnt Semantic Descriptors. *Eng. Sci. Technol. Int. J.* **2022**, *35*, 101098. [CrossRef]
37. Orhan, S.; Baştanlar, Y. Efficient Search in a Panoramic Image Database for Long-Term Visual Localization. In Proceedings of the 2021 IEEE/CVF International Conference on Computer Vision Workshops (ICCVW), Montreal, QC, Canada, 11–17 October 2021; pp. 1727–1734.

38. Goedemé, T.; Nuttin, M.; Tuytelaars, T.; Van Gool, L. Omnidirectional Vision Based Topological Navigation. *Int. J. Comput. Vis.* **2007**, *74*, 219–236. [CrossRef]
39. Murillo, A.C.; Singh, G.; Kosecká, J.; Guerrero, J.J. Localization in Urban Environments Using a Panoramic Gist Descriptor. *IEEE Trans. Rob.* **2013**, *29*, 146–160. [CrossRef]
40. Hansen, P.; Browning, B. Omnidirectional Visual Place Recognition Using Rotation Invariant Sequence Matching. 2015. Available online: https://kilthub.cmu.edu/ndownloader/files/12039332 (accessed on 15 September 2023).
41. Lu, H.; Li, X.; Zhang, H.; Zheng, Z. Robust Place Recognition Based on Omnidirectional Vision and Real-Time Local Visual Features for Mobile Robots. *Adv. Robot.* **2013**, *27*, 1439–1453. [CrossRef]
42. Wang, T.-H.; Huang, H.-J.; Lin, J.-T.; Hu, C.-W.; Zeng, K.-H.; Sun, M. Omnidirectional CNN for Visual Place Recognition and Navigation. In Proceedings of the 2018 IEEE International Conference on Robotics and Automation (ICRA), Brisbane, Australia, 21–25 May 2018; pp. 2341–2348.
43. Cheng, R.; Wang, K.; Lin, S.; Hu, W.; Yang, K.; Huang, X.; Li, H.; Sun, D.; Bai, J. Panoramic Annular Localizer: Tackling the Variation Challenges of Outdoor Localization Using Panoramic Annular Images and Active Deep Descriptors. In Proceedings of the 2019 IEEE Intelligent Transportation Systems Conference (ITSC), Auckland, New Zealand, 27–30 October 2019; pp. 920–925.
44. Hashemi, N.S.; Aghdam, R.B.; Ghiasi, A.S.B.; Fatemi, P. Template Matching Advances and Applications in Image Analysis. *arXiv* **2016**, arXiv:1610.07231.
45. Hisham, M.B.; Yaakob, S.N.; Raof, R.A.A.; Nazren, A.B.A.; Wafi, N.M. Template Matching Using Sum of Squared Difference and Normalized Cross Correlation. In Proceedings of the 2015 IEEE Student Conference on Research and Development (SCOReD), Kuala Lumpur, Malaysia, 13–14 December 2015; pp. 100–104.
46. Yoo, J.-C.; Han, T.H. Fast Normalized Cross-Correlation. *Circuits Syst. Signal Process.* **2009**, *28*, 819–843. [CrossRef]
47. Briechle, K.; Hanebeck, U.D. Template Matching Using Fast Normalized Cross Correlation. In Proceedings of the Optical Pattern Recognition XII, Orlando, FL, USA, 19 April 2001; Volume 4387, pp. 95–102.
48. Felzenszwalb, P.F.; Girshick, R.B.; McAllester, D.; Ramanan, D. Object Detection with Discriminatively Trained Part-Based Models. *IEEE Trans. Pattern Anal. Mach. Intell.* **2010**, *32*, 1627–1645. [CrossRef]
49. Talmi, I.; Mechrez, R.; Zelnik-Manor, L. Template Matching with Deformable Diversity Similarity. In Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017; pp. 175–183.
50. Luo, W.; Schwing, A.G.; Urtasun, R. Efficient Deep Learning for Stereo Matching. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016; pp. 5695–5703.
51. Wu, Y.; Abd-Almageed, W.; Natarajan, P. Deep Matching and Validation Network: An End-to-End Solution to Constrained Image Splicing Localization and Detection. In Proceedings of the 25th ACM International Conference on Multimedia, New York, NY, USA, 19 October 2017; pp. 1480–1502.
52. Cordts, M.; Omran, M.; Ramos, S.; Rehfeld, T.; Enzweiler, M.; Benenson, R.; Franke, U.; Roth, S.; Schiele, B. The Cityscapes Dataset for Semantic Urban Scene Understanding. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016; pp. 3213–3223.
53. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, Ł.; Polosukhin, I. Attention Is All You Need. In Proceedings of the Advances in Neural Information Processing Systems 30 (NIPS 2017), Long Beach, CA, USA, 4–9 December 2017.
54. Bai, H.; Wang, P.; Zhang, R.; Su, Z. SegFormer: A Topic Segmentation Model with Controllable Range of Attention. *Proc. AAAI Conf. Artif. Intell.* **2023**, *37*, 12545–12552. [CrossRef]
55. Simonyan, K.; Zisserman, A. Very Deep Convolutional Networks for Large-Scale Image Recognition. *arXiv* **2014**, arXiv:1409.1556.
56. Singh, P.K.; Sinha, S.; Choudhury, P. An Improved Item-Based Collaborative Filtering Using a Modified Bhattacharyya Coefficient and User–User Similarity as Weight. *Knowl. Inf. Syst.* **2022**, *64*, 665–701. [CrossRef]
57. Rathee, N.; Ganotra, D. An Efficient Approach for Facial Action Unit Intensity Detection Using Distance Metric Learning Based on Cosine Similarity. *Signal Image Video Process.* **2018**, *12*, 1141–1148. [CrossRef]
58. Dubey, V.K.; Saxena, A.K. A Sequential Cosine Similarity Based Feature Selection Technique for High Dimensional Datasets. In Proceedings of the 2015 39th National Systems Conference (NSC), Greater Noida, India, 14–16 December 2015; pp. 1–5.
59. Araújo, A.B. Drawing Equirectangular VR Panoramas with Ruler, Compass, and Protractor. *J. Sci. Technol. Arts* **2018**, *10*, 15–27. [CrossRef]
60. Zamir, A.R.; Shah, M. Image Geo-Localization Based on MultipleNearest Neighbor Feature Matching UsingGeneralized Graphs. *IEEE Trans. Pattern Anal. Mach. Intell.* **2014**, *36*, 1546–1558. [CrossRef]
61. Çinaroğlu, İ.; Baştanlar, Y. Image Based Localization Using Semantic Segmentation for Autonomous Driving. In Proceedings of the 2019 27th Signal Processing and Communications Applications Conference (SIU), Sivas, Turkey, 24–26 April 2019; pp. 1–4.
62. Cinaroglu, I.; Bastanlar, Y. Training Semantic Descriptors for Image-Based Localization. *arXiv* **2022**, arXiv:2202.01212.
63. Blanco-Claraco, J.-L.; Moreno-Dueñas, F.-Á.; González-Jiménez, J. The Málaga Urban Dataset: High-Rate Stereo and LiDAR in a Realistic Urban Scenario. *Int. J. Rob. Res.* **2014**, *33*, 207–214. [CrossRef]