

## Article

# A State-Based Language for Enhanced Video Surveillance Modeling (SEL)

Selene Ramirez-Rosales <sup>1</sup>, Luis-Antonio Diaz-Jimenez <sup>1</sup>, Daniel Canton-Enriquez <sup>1</sup>,  
Jorge-Luis Perez-Ramos <sup>1</sup>, Herlindo Hernandez-Ramirez <sup>2</sup>, Ana-Marcela Herrera-Navarro <sup>1</sup>,  
Gabriela Xicotencatl-Ramirez <sup>1</sup> and Hugo Jimenez-Hernandez <sup>1,\*</sup>

- <sup>1</sup> Facultad de Informatica, Universidad Autonoma de Queretaro, Av. de las Ciencias S/N, Juriquilla 76230, Mexico; selene.ramirez@uaq.mx (S.R.-R.); luis.diaz@uaq.edu.mx (L.-A.D.-J.); daniel.canton@uaq.edu.mx (D.C.-E.); jorge.luis.perez@uaq.edu.mx (J.-L.P.-R.); gabyxico@uaq.mx (G.X.-R.)  
<sup>2</sup> Centro de Ingeniería y Desarrollo Industrial (CIDESI), Av. Pie de la Cuesta No. 702, Desarrollo San Pablo, Santiago de Querétaro 76125, Mexico; hhernandez@posgrado.cidesi.edu.mx  
\* Correspondence: hugojh@gmail.com or hugo.jimenez@uaq.edu.mx

**Abstract:** SEL, a State-based Language for Video Surveillance Modeling, is a formal language designed to represent and identify activities in surveillance systems through scenario semantics and the creation of motion primitives structured in programs. Motion primitives represent the temporal evolution of motion evidence. They are the most basic motion structures detected as motion evidence, including operators such as sequence, parallel, and concurrency, which indicate trajectory evolution, simultaneity, and synchronization. SEL is a very expressive language that characterizes interactions by describing the relationships between motion primitives. These interactions determine the scenario's activity and meaning. An experimental model is constructed to demonstrate the value of SEL, incorporating challenging activities in surveillance systems. This approach assesses the language's suitability for describing complicated tasks.

**Keywords:** event description language; inference of activities; surveillance system; motion primitives; formal language



**Citation:** Ramirez-Rosales, S.; Diaz-Jimenez, L.A.; Canton-Enriquez, D.; Perez-Ramos, J.L.; Hernandez-Ramirez, H.; Herrera-Navarro, A.M.; Xicotencatl-Ramirez, G.; Jimenez-Hernandez, H. A State-Based Language for Enhanced Video Surveillance Modeling (SEL). *Modelling* **2024**, *5*, 549–568. <https://doi.org/10.3390/modelling5020029>

Received: 8 March 2024  
Revised: 17 May 2024  
Accepted: 21 May 2024  
Published: 24 May 2024



**Copyright:** © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

## 1. Introduction

Automatic surveillance systems specialized in video action recognition have shown significant growth in addressing issues such as detecting abnormal events, recognizing activities and actions, and understanding scenes. They aim to use technical means to determine a person's patterns or types of activity. This is crucial for a wide range of applications, such as long-term health monitoring, active and assisted living systems, monitoring and surveillance systems, and smart homes [1,2].

Traditional surveillance systems use a set of resources (cameras) implemented over a communication topology (network architecture) that sends the data to a concentrator (computational storage and computing resources) that interprets the data using a human-aided process [3].

Efficient fall detection is crucial in human activity recognition, highlighting the importance of developing databases with actual falls in uncontrolled environments. These databases have the advantage of being non-intrusive, allowing monitoring of specific scenarios without causing disruption. The amount of data generated by the various monitored scenarios is massive, but only some are relevant. Therefore, identifying the critical information in the scenario, detection, and interpretation allow for association with events of interest [4,5].

Activity recognition has attracted significant research attention due to the valuable insights, comprehensive analysis, and wide range of applications it offers. Various techniques, approaches, and methodologies have been used to achieve comprehensive recognition

and analysis of activities over time, tailored to specific applications with their unique characteristics [6].

Vision systems for activity detection can be intricate due to handling vast data and interpreting activities accurately within various constraints and variables of each scenario, enhancing complexity and specificity. Modeling and detecting activities through vision systems can be complex due to the information that is handled, as well as the interpretation that can be given to the activities to express them adequately regarding different restrictions and variables of the scenario of interest [7,8].

Several research initiatives are now defining new approaches to the design of systems that can understand human activities in dynamic scenes [2]. The increasing need for intelligent technologies that can understand and analyze intricate real-world scenarios is what motivates these efforts [9]. Feature extraction, learning, classification, object or region segmentation, and other techniques are frequently used in vision systems' analysis activities to identify and infer activities by modeling their temporal evolution [10,11].

Formal languages, with their structured grammar and rules, provide a robust framework for representing information structures within systems. They simplify the analysis process and enable the precise assignment of meaning to activities across various contexts and scenarios. Additionally, formal languages enhance technical education by enabling the clear expression of complex knowledge structures, ultimately increasing fluency and ensuring the decipherability of information [12,13].

The analysis of activities and behaviors allows us to create information analysis structures that can be used to model and infer activities. Challenges arise when the dynamics of the activities are unknown, and the scenarios are not controlled [14,15]. Movement structures observed in a monitoring system can be represented as primitive elements constituting the lexical and grammatical components of a language. From this viewpoint, movements structured in spatial and temporal dimensions can be modeled as tokens of a formal language, which is a hierarchical structure enabling interpretations of observed patterns as generalizations.

One of the many advantages of using a grammatical method is that it can leverage current data and reduce duplicates, which can lead to the eventual elimination of the requirement for redundant data collecting or extra training [16,17]. Although using formal language to describe and model activities provides remarkable expressiveness, it poses significant challenges when interpreting and translating data from real-world scenarios. A system such as this is devoted to the modeling and inference of activities and behavior within a scenario through a structured framework of grammar, rules, motion primitives, and state variables.

In this context, this paper provides a holistic viewpoint in which motion primitives can be introduced, which when generalized, allow the building of recursive procedures that allow modeling the object dynamics across multiple scenarios with different characteristics.

This article is structured as follows. Section 2 reviews related work, offering brief recommendations for the mode and inference of activities within structured distributed system architectures. In Section 3, we present a modeling language for activities and a methodology for its implementation. Subsequently, we discuss Experimental Analysis and Results in Section 4. Finally, in Section 5, we conclude with our findings and outline avenues for future research.

## 2. Related Work

Video action recognition research is an established and growing area because of the wide range of complex and variable settings and situations. In recent years, specialized automatic surveillance systems tailored for video action recognition have significantly progressed in handling challenges [18,19]. Artificial intelligence-based human action recognition in video sequences has advanced significantly, addressing challenges like detecting abnormal events and recognizing diverse activities in complex scenes using machine learning and deep learning [20].



Methods such as neural networks, Bayesian classifiers, and Hidden Markov Models have been extensively utilized in video recognition [16,21]. In contrast, contemporary approaches to shared recognition have witnessed significant growth but are also met with several formidable challenges. Notably, these challenges encompass the intricate task of processing the meaning of activities within their contextual framework and accounting for the movement and object dynamics [22].

Methods such as neural networks, Bayesian classifiers, and Hidden Markov Models have played a pivotal role in advancing video recognition [21]. One of the most notable hurdles involves processing the contextual meaning of activities while considering the movement and object dynamics. This task not only requires robust algorithms but also a deep understanding of the underlying context and environment in which the activities occur [22]. Additionally, the sheer complexity and variability of real-world scenarios add another layer of challenge, as activities can vary widely in appearance and context, making them difficult to accurately classify and interpret.

Multiple authors have proposed diverse criteria for analyzing and detecting activities using image classifiers. These techniques leverage the capabilities of image classification algorithms to identify motion patterns associated with different activities. However, it is important to note that the significant computational complexity inherent in these theoretical approaches can restrict their applicability in a broader context [23–28].

The objective of researchers is to improve comprehension and identification of activities across different domains through the integration of image classifiers and syntactic approaches. The potential of this interdisciplinary methodology to enhance the precision and effectiveness of activity analysis in intricate situations is considerable [23].

Syntactic approaches have emerged as integral components of different approaches to scrutinize a system's textual data stream and extract valuable insights from specific structures. These methods offer the distinct advantage of striking a balance between relatively low complexity and a high degree of expressiveness [3,29].

The syntactic method employs a set of symbols, each of them representing a sub-activity or atomic activity [25]. The inspiration for the syntactic method originates from the inference of activities through an intuitive procedure, where a person characterizes an action by applying grammatical syntax and rules, allowing the representation and interpretation of activities in a structured format [30].

Within the realm of formal languages for activity description, several noteworthy works have emerged, including ECA (Event–Condition–Action), ADeL (Activity Description Language), VIGILANT, SURVANT, CPNs (Colored Petri Nets), and ILIAD (Interactive Learning from Activity Description) [31–36].

ECA primarily uses grammar to detect conditioned activities based on specific events. In contrast, ADeL introduces an activity model that employs finite automaton to determine activities through a hierarchical language, incorporating roles, events, and sub-activities. ADeL is a language with a synchronization-oriented approach, sharing resemblances with other languages such as Scade, Esterel, Signal, and Luster [32,33]. The VIGILANT model combines object-oriented techniques and Description Logics (DLs) for efficient storage and retrieval of surveillance video content and events, enhancing semantic indexing capabilities. SURVANT is an innovative video archive investigation system that utilizes deep learning technologies for object recognition, tracking, and activity detection, enabling semantic indexing for efficient search and retrieval [35,36]. Conversely, ILIAD represents an interactive learning protocol that enables agents to undergo verbal training by describing their actions. Colored Petri Nets (CPNs) represent a discrete event modeling language that combines Petri nets with the functional programming language Standard ML [31,34,37].

Syntactical approaches, such as Stochastic Context-Free Grammar (SCFG), have been employed to model sequential activities within hierarchical analysis. In this approach, a set of symbols is defined. Each symbol represents a sub-activity or atomic activity. Grammar rules, a high-level activity, can be described as a set of state activations comparable to how a natural language expresses an action [25]. Subsequently, by utilizing grammar syntax and

rules, the syntactical approach facilitates the representation and interpretation of activities in a structured hierarchical manner [30].

Brand [37,38] was a pioneer in the use of a straightforward grammar-based approach for activity recognition, preceding the use of probabilistic models. This early work laid the foundation for incorporating grammar into activity recognition tasks.

Seong-Wook Joo and R. Chellappa proposed a grammar for recognizing activities using labels and syntax rules to describe events [39]. Human action recognition has been the subject of probabilistic approaches in other research. The primary objective of their research was to represent the human body through the extraction of crucial body image features. This allowed us to deduce potential actions by analyzing qualitative postural representations [40].

Conversely, context-free grammar was applied to recognize individual and group activities involving interactions. Their approach extended the use of grammar in activity recognition to incorporate more intricate activity patterns [41].

In addition to grammar-based approaches, other methods have been developed that utilize image segmentation, clustering, classification, and object identification techniques to recognize activities under specific constraints. These approaches leverage visual information and computational techniques to identify and classify activities.

Besides grammar-based methodologies, alternative methods have emerged, utilizing techniques such as image segmentation, clustering, classification, and object identification tailored explicitly for recognizing activities under particular constraints. These approaches harness visual information and computational techniques to identify and classify activities effectively.

Previous studies have emphasized the advantages of employing formal languages as a reliable approach to modeling and inferring activities. The proposal presents an expressive method to model different activities in different scenarios based on motion primitives, where the language has the quality of describing the scenario and the activities in various ways based on programs in the language.

The proposal specifies a grammar based on a hierarchical model that helps to generalize information structures in time and that are associated with a surveillance system associated with spatial areas, such as the description of symbols in time. The generated data sequences can be detected, characterized, and represented by compact structures such as grammars, given the fact that the repetition of evidence of movement in time gives the idea of recurring patterns in objects.

The grammar places emphasis on the use of motion primitives, which represent a set of operators to represent activities based on movement. The idea presented presents an expressive method to model multiple actions in different scenarios.

### 3. Proposal SEL: Language Description and Methodology

In this section, we will explore the different components of our proposal. We will start by assessing how the scenario is represented and selecting the key elements that define our system's environment. Following this, we will delve into the motion detection process, outlining the techniques and algorithms used to identify movement within the scene. Next, we will introduce motion primitives, the fundamental building blocks to describe complex activities. Finally, we will discuss the language and grammar used to model these activities, providing a structured framework for our analysis. We will conclude this section by outlining the methodology for the implementation of the language system and detailing the steps involved in putting our proposal into practice.

#### 3.1. Representation of the Scenario

The representation of the scenario is mainly by state, which is the fundamental unit of the language. These are the areas or segments that the user defines in the surveillance system's image, and we may utilize motion primitives to define actions within them.

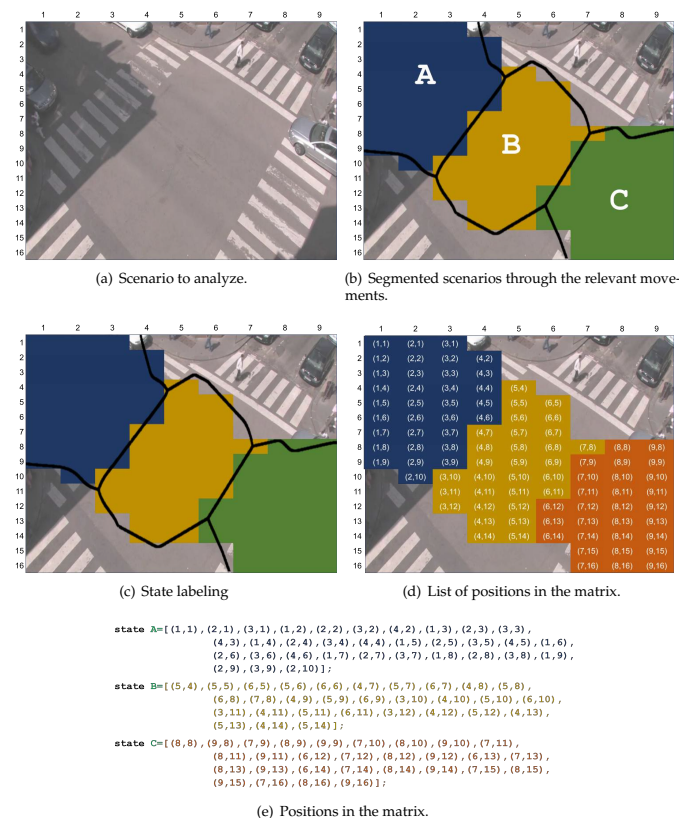
Each state generates a distinct list of scene positions based on the specified matrix. These states are assigned labels or names for ease of use with the motion primitives. Essentially, these states are segmented into spatial regions to confirm the presence or absence of motion.

The user has the flexibility to segment the states and define the information according to their preference. However, this work can be challenging, as it requires a detailed examination of the circumstances. Additionally, there is the potential to employ an automated approach for analyzing and determining the states, thereby enhancing the precision of the information and automatically establishing the states.

Image segmentation is defined as a partition process where homogeneous groups or regions are searched according to the requirements. The way in which image segmentation is performed efficiently depends on how it is interpreted and its application, since it can be very useful in different areas, and it can also be used to determine regions of interest or information of importance.

An image of the scenario is denoted as  $I(\mathbf{x})$ , where  $\mathbf{x}$  indexes the image information of color for a given position. The image dimensions are expressed by  $h$  and  $w$ , for which  $\mathbf{x} \in \chi$  and  $\chi$  is an Euclidean product of all available positions for indexing in the image  $\chi = [1, h] \times [1, w]$ . Image segmentation is expressed by  $S$ , which denotes a partition set of  $I(\mathbf{x})$ . This set has the form  $S = \{s_1, \dots, s_k\}$ , where each  $s_i \subseteq 2^\chi$ . Only consider the disjoint segments to conform  $S$ , and consequently for all given pairs  $s_i, s_j \in S$ , the intersection  $s_i \cap s_j = \emptyset$ .

The user defines the state divisions mentioned, which may apply in the context of Figure 1. This figure illustrates the segmentation and labeling process based on a scenario  $I(\mathbf{x})$ , ultimately producing a list of state positions in the image.



**Figure 1.** (a–e) Segmentation process and list of selected states.

The states can be presented in two forms: active/non-active (motion presence/absence). The activation of a particular set represents an event, and whenever it becomes *true*, it means that motion has been detected in the pixels that conform to the state.

### 3.2. Motion Detection

The motion detection subsystem corresponds to how motion is detected and can implement various approaches, such as background models [42] or optical flow approaches [43], to exchange different methods of motion detection.

The motion detector operator  $M$  is implemented using the temporal differences approach [44]. This approach models movement as a decay time function, gradually exciting it whenever a pixel shows movement. This method is valuable because activation within a short interval produces a peak in the curve, which then decays logarithmically. The advantage of this approach is that the curve reaches its highest activation in a brief interval and then starts to decay in logarithmic form. The speed at which the function decays is related to the temporality of the pixel's excitement. The interval of time before the motion evidence is dismissed represents the motion horizon evidence for the local motion displacement. Temporal differences for an isolated pixel are defined in recursive form by (1).

$$T_t(\mathbf{x}) = \alpha T_{t-1} + (1 - \alpha)d(I_t(\mathbf{x}), I_t(\mathbf{x})) \quad (1)$$

where  $\mathbf{x}$  represents the position of the pixel,  $T$  is the decay function in time  $t$ ,  $\alpha$  is the decay constant (motion dismiss),  $d(I_t(\mathbf{x}), I_t(\mathbf{x}))$  is a binary function that becomes active in motion presence [45,46]. In the practical approach, the frame difference has a Gaussian distribution with zero mean ( $\mu = 0$ ), and  $\sigma$  standard deviation is assumed. Under this assumption, noise effects due to the acquisition or small environment variations are dealt with as a single difference variation. Then, the probability of existing motion for a particular pixel is rewritten in terms of normalized distance to the Gaussian as follows.

$$d(x, \mathbf{G}) = \left( \frac{x - \mu}{\sigma} \right)^2 \quad (2)$$

$$P_\lambda(\mathbf{x}) = d(I_t(\mathbf{x}), \mathbf{G}) > \lambda \quad (3)$$

where  $\mathbf{G} = [\mu, \sigma]$  is a Gaussian expressed as a parameter vector with  $\mu$  mean and  $\sigma$  standard deviation;  $I_t$  is the current image and  $\lambda$  is a statistic value about the certainty of the Gaussian belonging. The return value in  $P_\lambda$  is a logical value referring to the presence/absence of a motion for a particular pixel. Summing up, a binary map computes all elements of the set  $Pb = \{\mathbf{x} \in [1, h] \times [1, w] \mid P_\lambda(I_t(\mathbf{x}))\}$  by testing all available pixel position for a given image. Then, motion operator  $M$  is defined for our purposes as:

$$M(S; \Phi) = \left\{ x \in \{\text{true}, \text{false}\}, s \in S \mid x = \frac{|s \cap Pb|}{|s|} > \rho \right\} \quad (4)$$

where vector parameters are defined by  $\Phi = [\alpha, \lambda, \rho]$ , and  $\alpha$  is the decay constant,  $\lambda$  is the statistical confidence to define the probability of the motion detection, and  $\rho$  indicates the threshold proportion in the area for the motion detection.

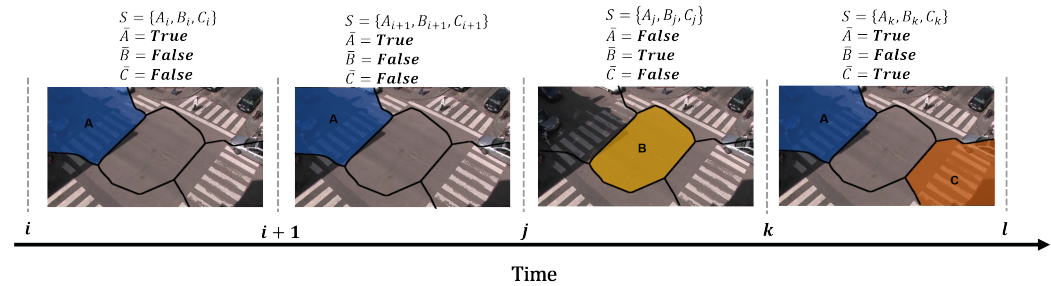
A motion detector operator  $M$  indicates when a state has motion, which is defined as

$$M : 2^X \times \Phi \rightarrow \{\text{true}, \text{false}\}. \quad (5)$$

In terms of notation, it is expressed as  $M(s; \Phi)$ , which represents any motion detection process implemented for testing the motion's presence/absence for the collection of pixels that conform to the state  $S$ , and  $\Phi$  represents the parameters of the process. It will be presented by  $S$  instead of  $M(S; \Phi)$  for notation simplicity. To simplify the syntax, a set of  $S$  is made up of  $n$  regions over a motion detection operator  $S = \{s_1, \dots, s_k\}$ .

A criterion for expressing activities as logical relationships is defined by the sequence of state activation across time, in which time is the frame video represented by  $i, j, k$ , and  $l$ . Figure 2 illustrates activated/non-active (true/false) states based on motion detection over time in 2D, the time sequence from  $I(\mathbf{x})_l^i$  illustrates the activation of the states, where the states are denoted as logical variables, the spatial dependencies are joined by zeroth-order logic, and the motion primitives describe the semantic motion relations in the scene.

The motion evidence performed in a particular scene  $I(\mathbf{x})_i^j$  might represent the relationship between moving objects, and the states represent individual logical variables. The descriptions of the relationships of motion states are expressed with the basic logic operators  $\{\wedge, \vee, \neg\}$  taken from zeroth-order logic as the motion primitive.



**Figure 2.** Scene sequences' 2D view of active states over time in different views (colored areas represent activated states).

Consequently, the temporal activation state has evolved to encompass temporal relations related to motion primitive operators. This advancement has given rise to a formal language that utilizes zeroth-order logic to characterize events based on the spatial relationships between active and non-active states, forming temporal constraint relationships. These primitives utilize zeroth-order logic to delineate spatial events' relationships, effectively encapsulating both time and motion behavior within the scene. The execution of motion primitives is inherently linked to segmentation in instances where movement is observed. Understanding the scenario is crucial for identifying these critical areas. Identifying these key areas stems from thoroughly examining each scenario, which aims to pinpoint critical regions for detection and activity modeling.

The use of the operators aims to create sentences with greater complexity and accuracy. The time relationships consider the time flow as unidirectional with constant increment (acquisition frequency), where for each state, it is possible to test the absence/presence of motion.

This representation shows the video sequences as timeslide images changing over time. The motion primitives add the extensibility to implement temporal relationships to spatial relationships expressed by the states. Generalizing the representation of the current image  $I(\mathbf{x})_i^j$ , the notation is used for  $I^t(\mathbf{x})$  to denote the indexing over time. Two continuous images are denoted by  $I^i(\mathbf{x})$  and  $I^{i+1}(\mathbf{x})$ . Then,  $\vec{s}_i^t \in \vec{S}^t$  denotes a particular state activation for a given motion frame  $\vec{S}^t$ .

### 3.3. Motion Primitives

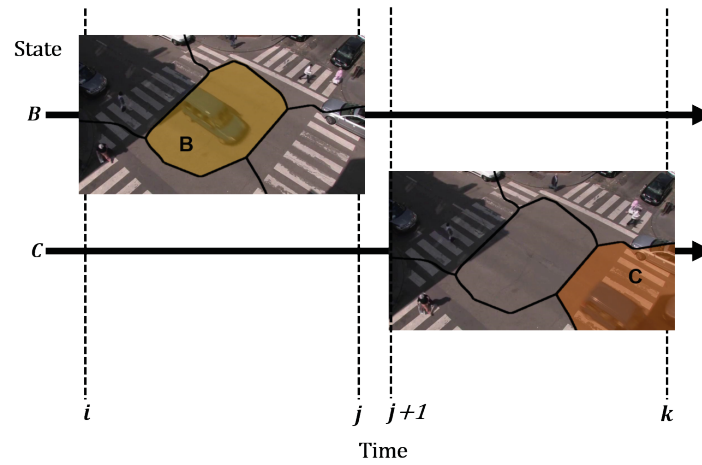
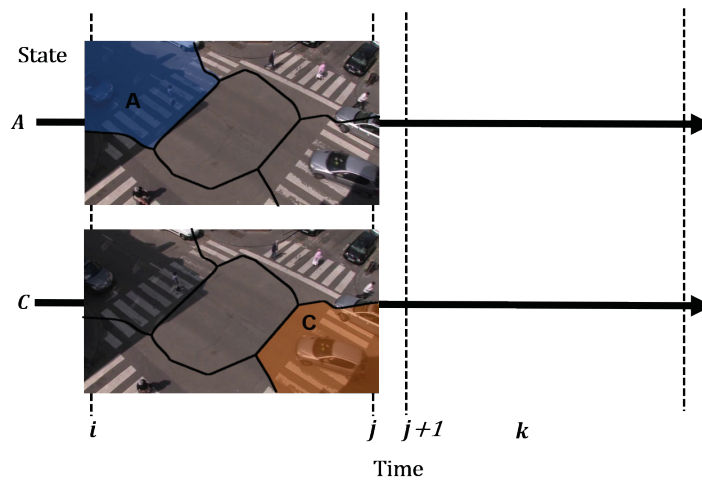
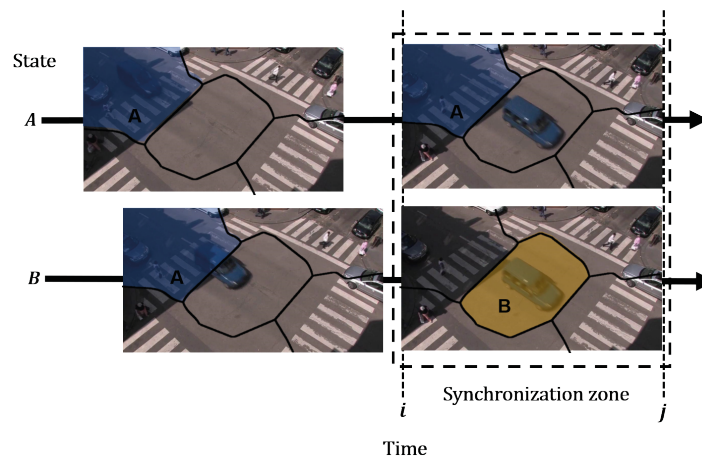
The development of motion primitives to determine and infer activities led to the development of a formal language, which employs zeroth-order logic to describe events based on the spatial relationships between active and non-active states.

SEL proposes that activities can be described through temporal relations in states, which are represented by motion primitives. The motion primitives proposed in this research include sequences, parallels, and concurrency. These operators are designed to represent trajectory evolution, simultaneity, and synchronization, respectively. The three primitives constitute the main foundational methods for representing motion.

The primary objective behind introducing these operators is to construct sentences with enhanced complexity and precision.

Figure 3 illustrates the *motion primitives*, to represent activities, while *sequence*, *parallel*, and *concurrency* impose time constraints on the execution of time order states.



(a) Sequence  $seq(B, C)$ (b) Parallel  $par(A, C)$ (c) Concurrency  $con(A, B)$ **Figure 3.** (a–c) The representation of motion primitives through time and states.

The *sequence primitive*, in the context of activity modeling, refers to the ordered execution of two or more states, where each state is activated after the previous one is completed. There is no time frame associated with the execution order. The time lapse is related to the

duration of the sequence of state activation. All states fall in a loop whenever it becomes active, considering only one event while presenting the motion.

The time state activation for a specific state  $s \in S$  is denoted as

$$\mathbf{s}(s; i, j) = \bigwedge_{t=i}^j \vec{s}^t \quad (6)$$

and this equation consequently returns a true or false.

The *sequence primitive* for a given pair of states  $s_i, s_j \in S$  is defined as  $seq : S \times S \rightarrow \text{true, false}$ . It requires at least two states and verifies that the subsequent state is activated after the previous one is completed, continuing this way until reaching the final state.

$$seq(a, b) \iff \mathbf{s}(a; i, j) \wedge \mathbf{s}(b; j+1, k) \wedge (\vec{a}^{j+1} = \text{false}) \quad (7)$$

for an arbitrary timestamp  $i, j, k$  with the restriction of  $i < j < k$ . The term  $(\vec{a}^{j+1} = \text{false})$  denotes the time order restriction.

The *parallel primitive* implementation involves two or more states are activated simultaneously and can run at the same time. There is no strict order in which the states must be activated, and they can occur concurrently, potentially resulting in faster execution of activities. For a given time interval  $[i, j]$ , the parallel is performed by the following operator:

$$par(a, b) \iff \mathbf{s}(a; i, j) \vee \mathbf{s}(b; i, j) \quad (8)$$

where  $par$  denotes the operator symbol, which is defined as  $par : S \times S \rightarrow \{\text{true, false}\}$ . If appreciated in a particular time, two states become parallel if one or both become active until none are active.

With the *concurrency primitive*, two or more states are also activated simultaneously, but in this case, the states are synchronized and operate cooperatively. Although they are activated simultaneously, they may have dependencies between them and must be coordinated to ensure that interactions between the states are handled correctly.

Then, for a given pair of states, the *concurrency primitive* is defined as follows:

$$con(a, b) \iff par(a, b) \wedge (\mathbf{s}(a; i, j) \wedge \mathbf{s}(b; i, j)) \quad (9)$$

where  $par(a, b)$  stands for time interval calculation,  $[h, i-1]$ , and both states ending in  $\mathbf{s}(a; i, j) \wedge \mathbf{s}(b; i, j)$  synchronize the join activation in a time interval  $[i, j]$ .

The differentiation between concurrent and parallel primitives is based on the following aspect. In parallel primitives, states can be active or non-active independently, regardless of the time interval, allowing a state to become active without considering the status of other states. In contrast, the concurrent operator permits simultaneous activation, but only if all states are activated until the final activation timestamp.

This language assumes that activity might be described as a set of active/non-active (true/false) states based on the presence/absence of motion over time using a logic description approach based on zeroth-order logic operators and a set of operators known as motion primitive operators.

The computational complexity of the SEL programming language is defined by the number of the motion primitives and states utilized to define. (see Equation (10)).

$$\Sigma^* = \{\{\emptyset, seq^*, par^*, con^*\} \times \{s_1, \dots, s_k\}\}^* \quad (10)$$

In conjunction, using the combination of the motion primitive operators makes it possible to define a formal language for describing the local and temporal relations between objects in a given particular scene, instead of considering a classical encode scheme based on the analysis of the dynamics of a given object.

The implementation of the motion primitive enables the creation of a formal language for describing relationships between objects in a scene, differing from traditional encod-

ing techniques that analyze specific object dynamics. The computational complexity of modeling with these operators depends on the complexity of the compositional recursive expression, which is compounded by the individual complexities of each operator.

### 3.4. Language Grammar

The set of primitives previously defined are grouped into complex expressions by the recursively compositional operator. To facilitate the expression writing, these operators can be implemented as a formal grammar to define an expressive language to code source scripts to represent and model complex processes.

In practical terms, whenever a string/word is verified by a language/automaton by an acceptance process, an acceptance process is performed by the analysis of the internal state/rule transitions matched with the rules/grammars that represent the language. The motion operators previously described become represented as orders in the language, such that any program encoded using this grammar represents recursively primitive operators that become associated with motion activities represented over a camera.

Figure 4 presents the grammar, detailing the definitions of the grammar rules associated with the proposed language. It also describes the identifiers, keywords, and symbols for the structures, along with the syntactic structure for programming in the language.

```
[program] ::= [header][body]
[header] ::= module [id];
[body] ::= [states] begin; [statements] end;
[activities] ::= [activity] | [activity] [activities]
[activity] ::= [primitive]([id],[id],[id]); [activityname]
[states] ::= [state] | [state] [states]
[state] ::= state [id] = [[area]];
[area] ::= ([number],[number]), [area] | ([number],[number])
[activityname] ::= %[id];
[primitive] ::= seq | par | con
[id] ::= [letra] | [letra][id]
[letra] ::= [A-Za-z]
[number] ::= [0-9]
```

**Figure 4.** Basic definition and grammar rules for the proposed formal language.

Afterward, the semantics for each reserved word and operator is defined as follows. The *state* operator defines all logical variables on the scene. Each state yields a single list of scene positions and the identifiers that label each state. The identifier represents the spatial area for testing the existence/absence of motion. The number of states defines the set of IDs labeled to each scene state. The semantics for  $\langle State \rangle$  is defined as:

$$S = \{ \langle Id_1 \rangle, \dots, \langle Id_k \rangle \} \text{ and} \\ \langle Id_i \rangle = \bigcup \{ (' \langle number \rangle', \langle number \rangle') \}$$

where  $\langle Id_i \rangle$  is the identifier and the list of positions that conform to the state is taken from the closure expression of  $\langle NL \rangle$ . Similarly, for the operators *sequence*, *concurrent*, and *parallel*:

$$\begin{aligned} \text{seq}(\langle Id \rangle \{, \langle Id \rangle \}+) \\ \text{con}(\langle Id \rangle \{, \langle Id \rangle \}+) \\ \text{par}(\langle Id \rangle \{, \langle Id \rangle \}+) \end{aligned}$$

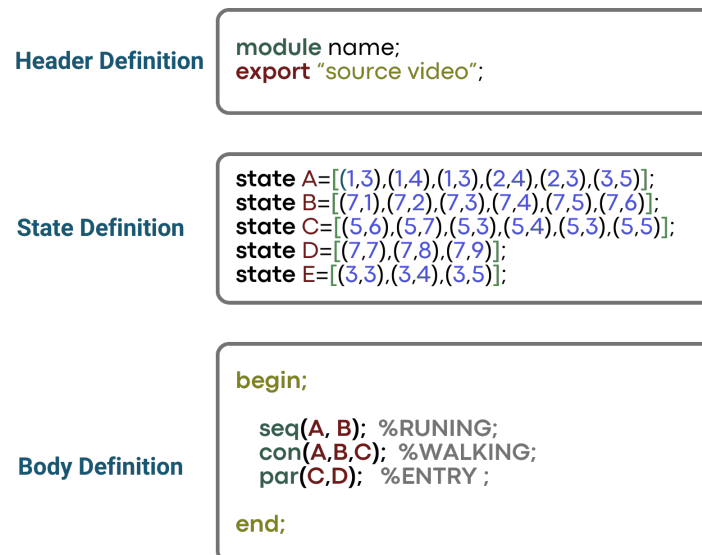
The expression  $\langle Id \rangle \{, ' \langle Id \rangle \}+)'$  represents the list of tested states and corresponds to the following expression, respectively.

$$\begin{aligned}
& seq(< Id_1 >, seq (< Id_2 >, seq(< Id_{n-1} >, < Id_n >)) \\
& con(< Id_1 >, con (< Id_2 >, con(< Id_{n-1} >, < Id_n >)) \\
& par(< Id_1 >, par (< Id_2 >, par(< Id_{n-1} >, < Id_n >))
\end{aligned}$$

Each program has three main parts: header definition, state definition, and body definition, as illustrated in Figure 5. The *header definition* specifies and identifies a process within the proposed SEL formal language and selects the data information source, document name, and video path to analyze.

The *state definition* focuses on defining the states according to the defined grammar according to the defined segmentation and labeling of the scene.

Finally, the body definition defines the application of motion primitives in relation to the defined states for modeling the activities and naming the activity. The output obtained by the interpreter generates a record each time the line of code evaluates to true in the system.



**Figure 5.** The proposed formal language SEL source file structure.

All computable processes written in the SEL language express a possible computable function that describes a specific motion dynamic. Once this process reaches an acceptance state, the complex dynamic expressed itself by the function represents an activity.

### 3.5. Methodology for Implementation

According to the surveillance system's standards, rigorous adherence to the following procedures is necessary, including using movement and language primitives. The methodology is detailed in Figure 6. This provides the systematic and accurate application of the SEL, permitting successful surveillance video analysis and interpretation.

1. **Scenario Analysis:** This step entails a thorough analysis of the scenario, concentrating on pertinent areas and activities, as depicted in Figure 6(1), which illustrates some frames of the stage.
2. **Segmentation:** The scenario is systematically divided into sections using a proposed matrix. The complexity of these segments can be precisely adjusted to match the demands of the activities they are modeling, i.e., in Figure 6(2), multiple segmentation options are presented as needed.
3. **State Listing:** A list of states and names is generated after segmentation, i.e., Figure 6(3) shows the matrix-based structure of the naming and state list.

4. Activity Modeling: After obtaining the states, we proceed to assess the appropriate motion primitives that will be employed for the purpose of modeling activities and constructing a statement, i.e., in Figure 6(4), a graphical representation of three alternative activities developed utilizing our recommended motion primitives is provided. The sequence primitive is utilized for activities such as tracking, while the concurrency primitive becomes vital in scenarios that demand synchronization, and finally, the parallelism primitive is intended specifically to detect movement inside designated zones.
5. Activity Script Creation: The process begins with the creation of one or more scripts for various activities. Each script is segmented, which is crucial for activities modeled using motion primitives. Once the script is completed with the grammar and syntax defined in SEL, the activities are modeled for later inference. For example, Figure 6(5) illustrates an instance of a script based on previous examples.



Figure 6. Methodology for SEL.



#### 4. Experimental Analysis and Results

The experimental section presents the modeling and interpretation of multiple activities, from simple to complex, for activity inference, comparing the expressiveness and simplicity of SEL with other methodologies available in the literature. Most approaches reported in the literature generally involve knowledge of activity detection, probability, activity analysis, and motion collection to produce the desired results.

The proposal enables the video interpretation of motion dynamics by creating scripts in SEL. The experimental model introduces four different types of scenarios (see Figures 7–10), corresponding to everyday video tasks. The objective is to implement multiple procedures in the proposed formal language SEL, utilizing various motion primitives for activity modeling and inference. The activities represented in the scenarios range from the simplest to the most complex. A description of the scenarios used is provided below.

- (a) **Single counting objects.** The first scenario involves passage states or access zones. Its goal is to track the trajectory of moving objects. The complexity arises from various external factors that the motion detector might detect, such as changes in lighting and multiple access points. In this scenario, each motion state represents an individual moving object, such as a person or a car, and must be counted.
- (b) **Activity inference through displacement trajectories.**  
Standard surveillance videos for traffic analysis are essential tasks in the context of an intelligent vehicle's density flow monitoring system, particularly for outdoor zones where illumination conditions are uncontrolled.  
The scenarios encompass various activities that vehicles perform as behaviors that can be recognized and labeled as allowed or not. However, similar to the first scenario, the detection of correct motion zones becomes challenging due to the uncontrollable outdoor conditions.
- (c) **Complex interaction between two or more objects.**  
This scenario represents one of the most complex dynamics of recognition. It involves object interactions, situations where two or more objects interact with a specific semantic interpretation, such as a handshake, where the motions of two states are synchronized. In this context, precise detection and tracking of moving objects are fundamental for correctly interpreting these complex interactions. Recognizing these interactions can have significant applications in advanced surveillance, human–robot interaction, and security monitoring.
- (d) **Complex activity detection.**  
In a complex scenario, multiple activities occur in different areas. These activities are repetitive, meaning that several state sequences can represent the same activity's activation. In this context, accurately identifying the activities and understanding their interactions are crucial for precise analysis and interpretation of the scenario's dynamics. Accurate detection and tracking of moving objects are essential for capturing the complexity of these simultaneous activities. This type of scenario presents unique challenges for surveillance and activity analysis systems, and a detailed understanding is crucial for developing effective monitoring and security solutions.

SEL language implementation describes activities as a state path-time activation in motion primitives. In the context of SEL language, the initial scenario describes the depiction of movement counting within specific areas. SEL operates assuming that the number of objects does not influence state activation. Therefore, the parallel primitive is the most suitable choice for this scenario. Consequently, the specific states situated along the path where objects move freely, in this context, indicate individuals crossing particular areas, which could be indicative of a potential counting statistic. The order of the path defines the direction of counting.

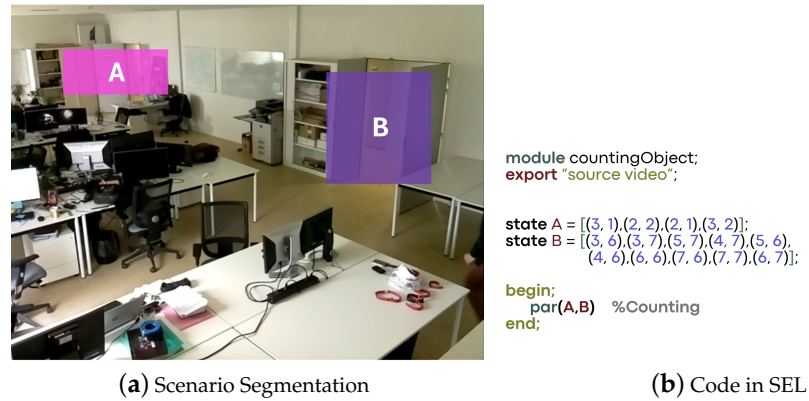


Figure 7. (a,b) Counting scenario.

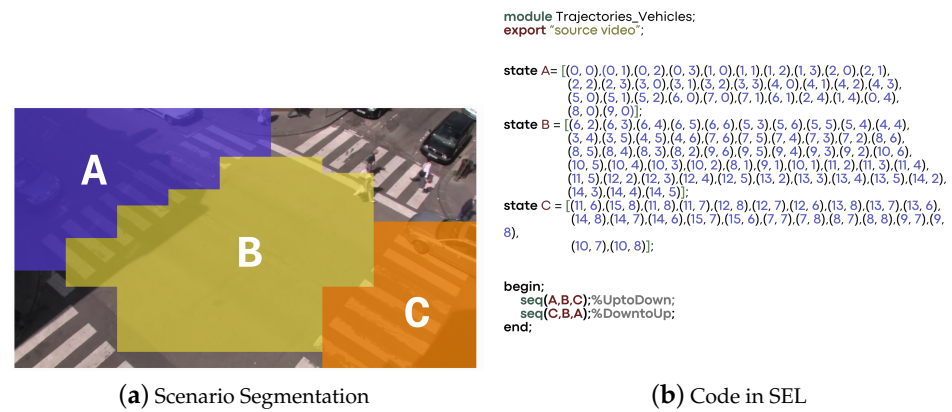


Figure 8. (a,b) Surveillance scenario.

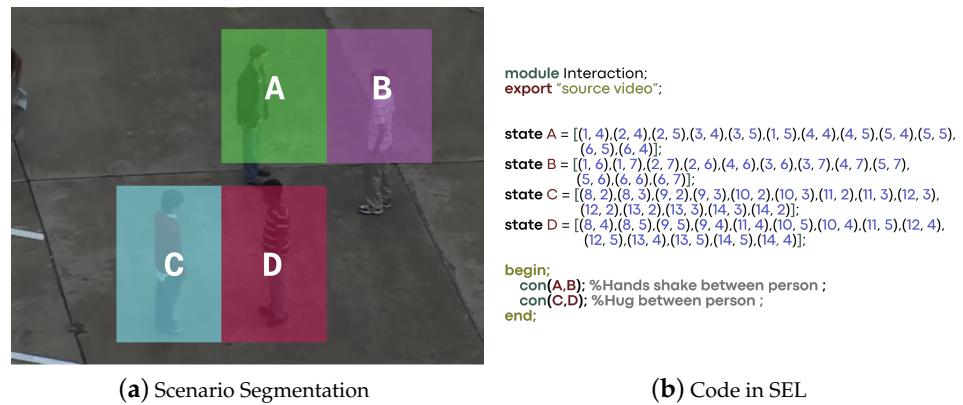


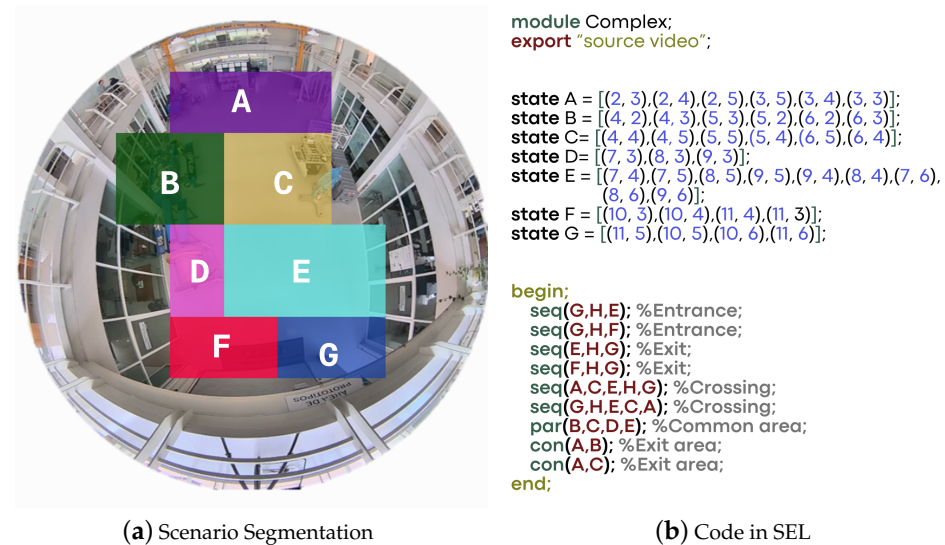
Figure 9. (a,b) Interaction scenario.

The following scenario illustrates the behavior as a sequence of states. Here, the primitive sequence is the optimal selection for a situation where the activation sequence portrays the spatial evidence of motion detected by a particular object. This scenario represents a behavior-tracking detector showcasing activity as a state sequence. The order of state activation establishes the directionality for analyzing the detected activities.

The resulting statistics pertain to an effective path detector, highlighting the recurring use of motion primitives to describe state relations. In this instance, the outputs of these operators are adequate for modeling the trajectory detector.

The inference of activity behavior in video sequences is a complex task. However, this work introduces a proposal based on state machines defined by grammar and a set of atomic motion detection events. The concatenation of all elements allows the detection

and interpretation of activities in video sequences through a set of programs written in the proposal of work entitled SEL.



**Figure 10.** (a,b) Complex scenario and code.

The interaction structure considers a synchronization process involving two or more parallel activation states. In these terms, concurrency represents the primitive for introducing a time constraint for state activation.

The complex scenario uses three motion primitives; some defined activities involve concurrency or parallelism primitives, while others, such as entry, exit, or stage crossing, follow a sequence primitive.

On the one hand, the scripts in the SEL language are compact due to state segmentation. However, over-segmentation of states can prevent situations from losing the specificity required to recognize the actions of a specific object. This means that actions involving articulations may take longer to detect. The balance between segmentation and specificity is influenced by prior environmental knowledge and the action being recognized.

Using a specialized programming language, such as SEL, facilitates modeling activities in surveillance systems by providing explicitly designed grammar and syntax for that purpose. SEL simplifies the development of models and algorithms that can detect and analyze activities in surveillance environments expressively and efficiently.

Motion primitives are one of the most essential components of the language because they simplify expressiveness for modeling and inference. In this case, SEL has three essential elements for modeling tasks, each with its own properties. The computational complexity of each motion primitive is shown in Table 1, the most complex being concurrency, which is the basis for the synchrony of the named states.

**Table 1.** Computational complexity of motion primitives.

| Motion Primitive | Symbol                            | Computational Complexity |
|------------------|-----------------------------------|--------------------------|
| Sequence         | $\text{seq}(\{s_1, \dots, s_k\})$ | $\mathcal{O}(k)$         |
| Parallelism      | $\text{par}(\{s_1, \dots, s_k\})$ | $\mathcal{O}(k)$         |
| Concurrency      | $\text{con}(\{s_1, \dots, s_k\})$ | $\mathcal{O}(k^2)$       |

$k$  = Number of states.

Considering the complexity of employing motion primitives, it is contrasted with other approaches provided for modeling and inferring activities shown in Table 2.

**Table 2.** Computational complexity of the approaches.

| Approach                      | Computational Complexity         |
|-------------------------------|----------------------------------|
| Hidden Markov Models          | $\mathcal{O}(k^2 * n)$           |
| Gaussian Mixture Model        | $\mathcal{O}(n * k^3)$           |
| Tracking optical flow         | $\mathcal{O}(n * (k^2) + (k^3))$ |
| Convolutional Neural Networks | $\mathcal{O}(k)$                 |
| PCA                           | $\mathcal{O}(k^3)$               |
| Bayes classifier              | $\mathcal{O}(k * n)$             |

$k$  = Number of states.  $n$  = Image size.

In Table 3, you can see different scenarios; some of them are easy and others challenging, where, based on a study of the scenario and its characteristics, a segmentation is carried out to be employed with the SEL motion primitives. It also shows how activities could be modeled differently using a more expressive and easy-to-use tool.

The motion sequence primitive is used to model activities especially in temporal sequences or trajectories; normally, this primitive may be compared with the Hidden Markov Model (HMM) approach, which is a powerful tool for identifying activity. HMM employ states and transitions to depict how activities progress chronologically. Furthermore, usage activity tracking is the process of continuously identifying and monitoring things or individuals over a period of time in sequential order. In Table 3, the motion sequence primitive mostly depicts normal trajectory activities or a specific input that fills rules in time and space.

On the other hand, modeling activities with motion parallelism primitive has a broader application; typically, this primitive can be used for count, alarm, prevention, and other applications. Because its primary purpose is to determine whether or not a movement state exists, as shown in Table 3, the activities of abnormal activity and crossing detection are modeled; however, both activities can be performed using other approaches, for example, the Bayes classifier, which is used to detect activity as part of a probability-based detection or classification approach, and Gaussian Mixture Model (GMM), which is based on previous training of normal activities, can also assist us with abnormal activities in the literature, because this model is effective in scenarios where the activities follow different statistical patterns and overlap on the input data.

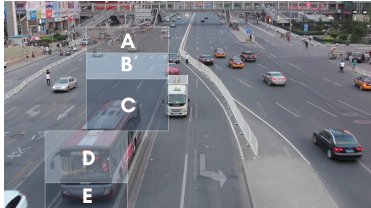
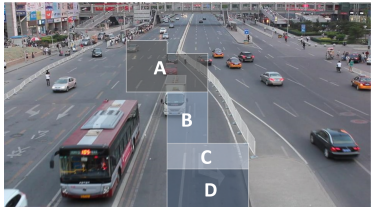
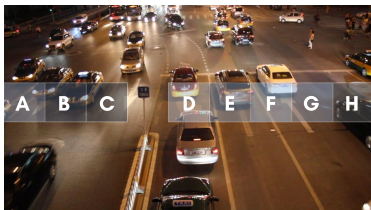
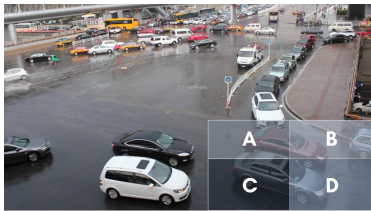
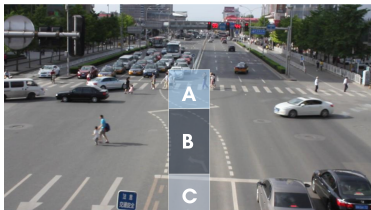
Finally, the motion concurrency primitive is used to model more complex activities, such as validating interactions or synchrony in specific areas; the activities modeled in Table 3 can validate a two-rail intersection, in which the union of a rail or another action joins the rail. For this purpose, it is usually possible to use neural networks that are effective at modeling and identifying patterns and changes in behavior. Principal Component Analysis (PCA) allows activities to be represented more concisely by selecting the most relevant principal components, making it easier to identify similarities, differences, and critical factors that influence activities.

Table 3 assumes that the segmentation is given arbitrarily and subjectively to demonstrate that activities can be modeled with more or fewer states depending on the scenario. Many methods mentioned or compared require prior training, information, or knowledge about activity detection; however, using SEL only requires language knowledge and motion primitives.

The use of motion primitives according to the activity that is desired to be modeled, using a language based on motion primitives, is an expressive and straightforward way to implement modeling for activity inference, particularly for a public inexperienced in managing surveillance systems.



Table 3. Modeling activities with SEL.

| Scenario                                                                            | States                                                                                                                                                                                                                                                                                                                                               | Activity Name          | SEL                           |
|-------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------|-------------------------------|
|    | <b>state A</b> =[(5, 0),(6, 0)];<br><b>state B</b> =[(7, 0),(8, 0),(9, 0),(10, 0),(11, 0)];<br><b>state C</b> =[(10, 6),(11, 6)];<br><b>state D</b> =[(12, 6),(12, 7),(13, 7),(13, 6),<br>(14, 6),(14, 7),(15, 7),(15, 6)];                                                                                                                          | Abnormal activity      | $par(A, B, C, D)$             |
|    | <b>state A</b> =[(2, 2),(3, 2),(3, 3),(2, 3)];<br><b>state B</b> =[(4, 3),(4, 2),(5, 2),(5, 3)];<br><b>state C</b> =[(6, 2),(6, 3),(7, 3),(8, 3),(9, 3),<br>(9, 2),(8, 2),(7, 2)];<br><b>state D</b> =[(10, 1),(10, 2),(11, 2),(11, 1),<br>(12, 1),(12, 2),(13, 2),(13, 1)];<br><b>state E</b> =[(15, 1),(14, 1),(14, 2),(15, 2)];                   | Normal trajectory flow | $seq(A, B, C, D, E)$          |
|   | <b>state A</b> =[(3, 3),(4, 3),(4, 4),(5, 4),(6, 4),<br>(6, 3),(5, 3)];<br><b>state B</b> =[(7, 4),(8, 4),(9, 4),(10, 4)];<br><b>state C</b> =[(11, 5),(11, 4),(12, 4),(12, 5)];<br><b>state D</b> =[(13, 4),(13, 5),(14, 5),(14, 4),<br>(15, 4),(15, 5)];                                                                                           | Lateral road entry     | $seq(A, B, C, D)$             |
|  | <b>state A</b> =[(6, 0),(7, 0),(8, 0)];<br><b>state B</b> =[(6, 1),(7, 1),(8, 1)];<br><b>state C</b> =[(6, 2),(7, 2),(8, 2)];<br><b>state D</b> =[(6, 4),(7, 4),(8, 4)];<br><b>state E</b> =[(6, 5),(7, 5),(8, 5)];<br><b>state F</b> =[(6, 6),(7, 6),(8, 6)];<br><b>state G</b> =[(6, 7),(7, 7),(8, 7)];<br><b>state H</b> =[(6, 8),(7, 8),(8, 8)]; | Double traffic lanes   | $con(A, B, C, D, E, F, G, H)$ |
|  | <b>state A</b> =[(9, 5),(9, 6),(10, 6),(11, 6),(11, 5),<br>(10, 5)];<br><b>state B</b> =[(9, 7),(9, 8),(10, 8),(11, 8),(11, 7),<br>(10, 7)];<br><b>state C</b> =[(12, 5),(12, 6),(13, 6),(14, 6),<br>(14, 5),(13, 5),(15, 5),(15, 6)];<br><b>state D</b> =[(12, 7),(12, 8),(14, 8),(13, 7),<br>(13, 8),(14, 7),(15, 7),(15, 8)];                     | Join the highway       | $con(A, B, C, D)$             |
|  | <b>state A</b> =[(5, 4),(6, 4),(7, 4)];<br><b>state B</b> =[(8, 4),(9, 4),(10, 4),(11, 4),(12, 4)];<br><b>state C</b> =[(13, 4),(14, 4),(15, 4)];                                                                                                                                                                                                    | Crossing detection     | $par(A, B, C, D)$             |

## 5. Conclusions

The suggested formal language, SEL, aims to automate the modeling and inference of activities by encoding a source file to describe activities logically. Motion primitives are a fundamental aspect of language, as actions are modeled from them.



The experimental area contains a spectrum of scenarios that depict numerous complexities and standard tasks of the vision system. These scenarios range from the simplest, such as counting objects, to the most complicated, like dealing with sophisticated human interactions. These tests show that SEL is an adequate and precise formal language for modeling activities as long as you know the scenario. This contrasts with other approaches in the literature, which are generally focused on specialized models based on theoretical assumptions.

The fundamental purpose is to rationally characterize the suggested activity discovery through logical explanations of spatial and temporal correlations. Meanwhile, other research employs different methodologies, including Bayesian classification, incremental learning, activation criteria, automatic learning, and unsupervised learning. SEL involves scenario analysis, activity determination, and script construction using grammar and given motion primitives.

The segmentation criteria are determined by prior knowledge of the scenario. However, future research will include an automated segmentation strategy based on motion detection. Future work will concentrate on methods for calculating the number of states of a scene using input from moving objects and improving the size of the matrix for state definition, which will become increasingly scene-specific. The motion primitives describe motion detection and synchronization in specific scenario states. Furthermore, these motion primitives include a temporal constraint on the motion's execution path.

The scenarios demonstrate that the SEL language can be used as an alternative to specific approaches used by other authors in the literature to interpret model activity. As a result, rather than specific models based on theoretical approaches, the goal is to express the proposal in terms of a logical description of spatial and temporal relationships to discover activities.

The current version of SEL is built for centralized systems, but future advancements may allow its adaptation to multi-camera dispersed systems. Finally, the next stage entails automatically producing code depending on specific camera settings. The simplicity and grammatical expressiveness inherent to SEL facilitate automatic code generation to implement the proposed language.

**Author Contributions:** Conceptualization, S.R.-R. and H.J.-H.; methodology, S.R.-R. and H.J.-H.; software, S.R.-R. and D.C.-E.; validation, H.H.-R., A.-M.H.-N. and G.X.-R.; formal analysis, H.J.-H. and H.H.-R.; investigation, J.-L.P.-R., D.C.-E. and L.-A.D.-J.; writing—review and editing, H.J.-H., A.-M.H.-N. and S.R.-R.; supervision, H.J.-H. All authors have read and agreed to the published version of the manuscript

**Funding:** This research received no external funding.

**Data Availability Statement:** The raw data supporting the conclusions of this article will be made available by the authors on request.

**Conflicts of Interest:** The authors declare that there are no conflicts of interest regarding the publication of this paper and declare that all materials and methods presented are original and the result of the research field of the group.

## References

1. Fan, C.; Gao, F. Enhanced Human Activity Recognition Using Wearable Sensors via a Hybrid Feature Selection Method. *Sensors* **2021**, *21*, 6434. [[CrossRef](#)] [[PubMed](#)]
2. Kong, Y.; Fu, Y. Human Action Recognition and Prediction: A Survey. *arXiv* **2018**, arXiv:1806.11230.
3. Abu-Bakar, S.A.R. Advances in human action recognition: An updated survey. *IET Image Process.* **2019**, *13*, 2381–2394. [[CrossRef](#)]
4. Eraso Guerrero, J.C.; Muñoz España, E.; Muñoz Añasco, M. Human Activity Recognition via Feature Extraction and Artificial Intelligence Techniques: A Review. *Tecnura* **2022**, *26*, 213–236. [[CrossRef](#)]
5. Ke, S.R.; Thuc, H.; Lee, Y.J.; Hwang, J.N.; Yoo, J.H.; Choi, K.H. A Review on Video-Based Human Activity Recognition. *Computers* **2013**, *2*, 88–131. [[CrossRef](#)]
6. Shakya, S.; Zhang, C.; Zhou, Z. Comparative Study of Machine Learning and Deep Learning Architecture for Human Activity Recognition Using Accelerometer Data. *Int. J. Mach. Learn. Comput.* **2018**, *8*, 577–582. [[CrossRef](#)]

7. Ravipati, A.; Kondamuri, R.K.; Posonia, M. Vision Based Detection and Analysis of Human Activities. In Proceedings of the 2023 7th International Conference on Trends in Electronics and Informatics (ICOEI), Tirunelveli, India, 11–13 April 2023; pp. 1542–1547. [\[CrossRef\]](#)
8. Morris, B.T.; Trivedi, M.M. A survey of vision-based trajectory learning and analysis for surveillance. *IEEE Trans. Circuits Syst. Video Technol.* **2008**, *18*, 1114–1127. [\[CrossRef\]](#)
9. Vu, V.T.; Brémond, F.; Thonnat, M. Automatic Video Interpretation: A Novel Algorithm for Temporal Scenario Recognition. In Proceedings of the International Joint Conference on Artificial Intelligence, Acapulco, Mexico, 9–15 August 2003.
10. Lou, J.; Liu, Q.; Tan, T.; Hu, W. Semantic interpretation of object activities in a surveillance system. In Proceedings of the Object Recognition Supported by User Interaction for Service Robots, Quebec City, QC, Canada, 11–15 August 2002; Volume 3, pp. 777–780. [\[CrossRef\]](#)
11. Morris, B.; Trivedi, M. Learning trajectory patterns by clustering: Experimental studies and comparative evaluation. In Proceedings of the 2009 IEEE Conference on Computer Vision and Pattern Recognition, Miami, FL, USA, 20–25 June 2009; pp. 312–319. [\[CrossRef\]](#)
12. Lewis, H.R.; Papadimitriou, C.H. Elements of the Theory of Computation. *SIGACT News* **1998**, *29*, 62–78. [\[CrossRef\]](#)
13. Lowry, E. Formal Language as a Medium for Technical Education. In Proceedings of the ED-MEDIA 96, Boston, MA, USA, 17–22 June 1996.
14. Kim, E.; Helal, S.; Cook, D. Human Activity Recognition and Pattern Discovery. *IEEE Pervasive Comput.* **2010**, *9*, 48–53. [\[CrossRef\]](#)
15. Yao, B.; Jiang, X.; Khosla, A.; Lin, A.L.; Guibas, L.; Fei-Fei, L. Human Action Recognition by Learning Bases of Action Attributes and Parts. In Proceedings of the 2011 International Conference on Computer Vision, Barcelona, Spain, 6–13 November 2011.
16. Chmiel, W.; Kwiecień, J.; Mikrut, Z. Realization of Scenarios for Video Surveillance. *Image Process. Commun.* **2012**, *17*, 231. [\[CrossRef\]](#)
17. Lee, J.; Ahn, B. Real-Time Human Action Recognition with a Low-Cost RGB Camera and Mobile Robot Platform. *Sensors* **2020**, *20*, 2886. [\[CrossRef\]](#) [\[PubMed\]](#)
18. Tran, D.; Bourdev, L.D.; Fergus, R.; Torresani, L.; Paluri, M. C3D: Generic Features for Video Analysis. *arXiv* **2014**, arXiv:1412.0767.
19. Devanne, M.; Wannous, H.; Berretti, S.; Pala, P.; Daoudi, M.; Del Bimbo, A. 3-D Human Action Recognition by Shape Analysis of Motion Trajectories on Riemannian Manifold. *IEEE Trans. Cybern.* **2015**, *45*, 1340–1352. [\[CrossRef\]](#) [\[PubMed\]](#)
20. Kumar, R.; Kumar, S. Survey on artificial intelligence-based human action recognition in video sequences. *Opt. Eng.* **2023**, *62*, 023102. [\[CrossRef\]](#)
21. Kim, B.; Lee, J. A Bayesian Network-Based Information Fusion Combined with DNNs for Robust Video Fire Detection. *Appl. Sci.* **2021**, *11*, 7624. [\[CrossRef\]](#)
22. Simonyan, K.; Zisserman, A. Two-Stream Convolutional Networks for Action Recognition in Videos. *arXiv* **2014**, arXiv:1406.2199.
23. Poppe, R. A survey on vision-based human action recognition. *Image Vis. Comput.* **2010**, *28*, 976–990. [\[CrossRef\]](#)
24. Turaga, P.; Chellappa, R.; Subrahmanian, V.S.; Udrea, O. Machine Recognition of Human Activities: A Survey. *IEEE Trans. Circuits Syst. Video Technol.* **2008**, *18*, 1473–1488. [\[CrossRef\]](#)
25. Aggarwal, J.; Ryoo, M. Human Activity Analysis: A Review. *ACM Comput. Surv.* **2011**, *43*, 16. [\[CrossRef\]](#)
26. Candamo, J.; Shreve, M.; Goldgof, D.; Sapper, D.; Kasturi, R. Understanding Transit Scenes: A Survey on Human Behavior-Recognition Algorithms. *IEEE Trans. Intell. Transp. Syst.* **2010**, *11*, 206–224. [\[CrossRef\]](#)
27. Chaudhary, A.; Raheja, J.L.; Das, K.; Raheja, S. A survey on hand gesture recognition in context of soft computing. *Commun. Comput. Inf. Sci.* **2011**, *133 CCIS*, 46–55. [\[CrossRef\]](#)
28. Hosler, B.C.; Zhao, X.; Mayer, O.; Chen, C.; Shackleford, J.A.; Stamm, M.C. The Video Authentication and Camera Identification Database: A New Database for Video Forensics. *IEEE Access* **2019**, *7*, 76937–76948. [\[CrossRef\]](#)
29. Malgiredy, M.R.; Nwogu, I.; Govindaraju, V. Language-Motivated Approaches to Action Recognition. *J. Mach. Learn. Res.* **2013**, *14*, 2189–2212.
30. Yang, Z.; Kay, A.; Li, Y.; Cross, W.; Luo, J. Pose-based Body Language Recognition for Emotion and Psychiatric Symptom Interpretation. *arXiv* **2020**, arXiv:2011.00043.
31. Alferes, J.; Banti, F.; Brogi, A. An Event-Condition-Action Logic Programming Language. In *Logics in Artificial Intelligence (JELIA 2006)*; Lecture Notes in Computer Science; Springer: Berlin, Heidelberg, 2006; Volume 4160. [\[CrossRef\]](#)
32. Sarray, I.; Ressouche, A.; Moisan, S.; Rigault, J.; Gaffe, D. An activity description language for activity recognition. In Proceedings of the 2017 International Conference on Internet of Things, Embedded Systems and Communications (IINTEC), Gafsa, Tunisia, 20–22 October 2017; pp. 177–182. [\[CrossRef\]](#)
33. Nguyen, N.T.; Phung, D.Q.; Venkatesh, S.; Bui, H. Learning and detecting activities from movement trajectories using the hierarchical hidden Markov model. In Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05), San Diego, CA, USA, 20–25 June 2005; Volume 2, pp. 955–960. [\[CrossRef\]](#)
34. Jensen, K.; Kristensen, L.M.; Wells, L. Coloured Petri Nets and CPN Tools for modelling and validation of concurrent systems. *Int. J. Softw. Tools Technol. Transf.* **2007**, *9*, 213–254. [\[CrossRef\]](#)

35. Vella, G.; Dimou, A.; Gutierrez-Perez, D.; Toti, D.; Nicoletti, T.; La Mattina, E.; Grassi, F.; Ciapetti, A.; McElligott, M.; Shahid, N.; et al. SURVANT: An Innovative Semantics-Based Surveillance Video Archives Investigation Assistant. In Proceedings of the Pattern Recognition. ICPR International Workshops and Challenges, Virtual Event, 10–15 January 2021; Del Bimbo, A., Cucchiara, R., Sclaroff, S., Farinella, G.M., Mei, T., Bertini, M., Escalante, H.J., Vezzani, R., Eds.; Springer: Cham, Switzerland, 2021; pp. 611–626.
36. Zerzour, K.; Frazier, G. VIGILANT: A semantic Model for Content and Event Based Indexing and Retrieval of Surveillance Video. In Proceedings of the Knowledge Representation Meets Databases, Berlin, Germany, 21 August 2000.
37. Lei, Q.; Du, J.; Zhang, H.; Ye, S.; Chen, D.S. A Survey of Vision-Based Human Action Evaluation Methods. *Sensors* **2019**, *19*, 4129. [\[CrossRef\]](#)
38. Brand, M. Understanding manipulation in video. In Proceedings of the Second International Conference on Automatic Face and Gesture Recognition, Killington, VT, USA, 14–16 October 1996; pp. 94–99. [\[CrossRef\]](#)
39. Joo, S.W.; Chellappa, R. Attribute Grammar-Based Event Recognition and Anomaly Detection. In Proceedings of the 2006 Conference on Computer Vision and Pattern Recognition Workshop (CVPRW'06), New York, NY, USA, 17–22 June 2006; p. 107. [\[CrossRef\]](#)
40. Duckworth, P.; Hogg, D.C.; Cohn, A.G. Unsupervised human activity analysis for intelligent mobile robots. *Artif. Intell.* **2019**, *270*, 67–92. [\[CrossRef\]](#)
41. Ryoo, M.S.; Aggarwal, J.K. Recognition of Composite Human Activities through Context-Free Grammar Based Representation. In Proceedings of the 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06), New York, NY, USA, 17–22 June 2006; Volume 2, pp. 1709–1718. [\[CrossRef\]](#)
42. Schuldt, C.; Laptev, I.; Caputo, B. Recognizing human actions: A local SVM approach. In Proceedings of the 17th International Conference on Pattern Recognition, ICPR 2004, Cambridge, UK, 26 August 2004; Volume 3, pp. 32–36. [\[CrossRef\]](#)
43. Ikizler-Cinbis, N.; Sclaroff, S. Object, Scene and Actions: Combining Multiple Features for Human Action Recognition. In Proceedings of the ECCV, 11th European Conference on Computer Vision, Heraklion, Crete, Greece, 5–11 September 2010.
44. Richard, A.; Gall, J. Temporal Action Detection Using a Statistical Language Model. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016; pp. 3131–3140. [\[CrossRef\]](#)
45. García-Huerta, J.M.; Jiménez-Hernández, H.; Herrera-Navarro, A.M.; Hernández-Díaz, T.; Terol-Villalobos, I. Modelling dynamics with context-free grammars. In Proceedings of the IS&T/SPIE Electronic Imaging, San Francisco, CA, USA, 2–6 February 2014. [\[CrossRef\]](#)
46. Bobick, A.F.; Davis, J.W. The recognition of human movement using temporal templates. *IEEE Trans. Pattern Anal. Mach. Intell.* **2001**, *23*, 257–267. [\[CrossRef\]](#)

**Disclaimer/Publisher's Note:** The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.