Expert Systems With Applications 126 (2019) 246–264

Contents lists available at ScienceDirect

Expert Systems With Applications

journal homepage: www.elsevier.com/locate/eswa

A multi-agent dynamic system for robust multi-face tracking

Lada Maleš[a,*], Darijan Marčetić[b], Slobodan Ribarić[b]

[a] Faculty of Humanities and Social Sciences, University of Split, Poljička cesta 35, 21000 Split, Croatia
[b] Faculty of Electrical Engineering and Computing, University of Zagreb, Unska 3, 10000 Zagreb, Croatia

ARTICLE INFO

Article history:
Received 28 July 2018
Revised 5 December 2018
Accepted 4 February 2019
Available online 8 February 2019

Keywords:
Multi-agent dynamic system
BDI agent
Multi-face tracking
De-identification pipeline

ABSTRACT

The paper presents a new architecture framework in the field of expert and intelligent systems which is based on four paradigms: a novel multi-agent dynamic system architecture (MADS), an extended Belief Desire Intention (EBDI) agent community, autonomy-oriented entities (AOEs), and deep learning concepts. The main impact of the proposed framework is a new approach, or even a new way of thinking, which enables integration of the concepts of deep learning, a conventional approach to solving the domain problem, cognitive agents with mental attitudes, and concepts of nature-inspired computing. All these allow the effective use of the framework in the field of intelligent and expert systems. The significance of the framework lies in its flexibility and adaptability based on the formal logical description of EBDI agents, the definition of the behaviour of AOEs, the use of classical modules for domain problem solving, and modules based on deep-learning concepts. We believe that the example of the adaptation of the proposed architecture framework to robust multi-face tracking illustrates the significance of the proposed framework. In this paper, MADS is adapted to the first two stages of a face de-identification pipeline: robust face detection and multi-face tracking. The proposed architecture of MADS has a two-level hierarchical organization. At the first level there is a manager designed as an Extended Belief Desire Intention (mEBDI) agent. The extension of a manager BDI agent consists of a convolutional neural network-based face detector, a set of autonomous-oriented entities for the elimination of false positive face detections, and a trajectory memory. At the second level, there are many tracking agents (trEBDIs) which consist of a basic BDI agent extended with a face tracker based on position and scale correlation filters, a visual appearance memory, and a trajectory memory. The mEBDI and trEBDI agents are defined by the modal logic and are described at the implementation level. The proposed architecture for a robust multi-face tracking system was tested on a subset of YouTube music videos. The qualitative results, as well as the preliminary quantitative results expressed by the standard testing metrics, demonstrate the effective adaptation of the proposed multi-agent dynamic architecture to a robust multi-face tracking system.

1. Introduction

Recent advances in cameras, recording devices, web technology and signal processing have improved the effectiveness of video surveillance, primarily for the benefit of security and law enforcement. This technology is now widely exploited in a variety of scenarios to capture video recordings of people in public, semi-public and even in private environments, either for immediate inspection or for storage, and for subsequent data analysis and sharing (Ribarić, Ariyaeeinia, & Pavešić, 2016). Whilst in many situations, such as law enforcement, forensics, bioterrorism surveillance, and disaster prediction, there are justifiable reasons for recording, storing and analysing video data acquired in such ways, there is also a strong need to protect the privacy of individuals who are inevitably captured in the recordings. De-identification is one of the basic methods for protecting privacy. It is defined as the process of removing or concealing personal identifiable information, i.e. personal identifiers, or replacing them with surrogate personal identifiers, to prevent the recognition of an individual directly or indirectly by human and/or machine (Ribarić & Pavešić, 2017).

An individual may be recognized (identified) based on biometric physiological (face, ear, iris, fingerprint) and/or behavioural (voice, gait, gesture, lip-motion, style of typing) identifiers, but also based on a combination of both, or additionally with the help of soft-biometric identifiers, such as body silhouette, age, gender, race, moles, and tattoos.

The face is, without doubt, the main biometric personal identifier used for biometric-based identification at distance (Jain & Li, 2011). Face-based identification is used in various application scenarios – from identification of a person based on a still image

* Corresponding author.
E-mail addresses: lada.males@ffst.hr (L. Maleš), darijan.marcetic@fer.hr (D. Marčetić), slobodan.ribaric@fer.hr, slobodan.ribaric@zemris.fer.hr (S. Ribarić).

https://doi.org/10.1016/j.eswa.2019.02.008
0957-4174/

in a passport or on an identity card, through identification of persons in photographs of crowded scenes, to identification based on face images captured either overtly or covertly by a video surveillance system (Ribarić & Pavešić, 2017). In many application scenarios, especially in video surveillance, privacy can be compromised.

Face de-identification is a way to protect the individual's privacy in video surveillance systems (Ribarić & Pavešić, 2015) and it is performed by a so-called de-identification pipeline which consists of the following three main pipeline stages: face detection, face tracking, and face masking by applying different privacy filters (Ribarić et al., 2016).

The first stage in a face de-identification pipeline is face detection. Due to large variances in poses of the face, sizes, bad lighting conditions, the face affected by partial occlusion, the presence of structural components (e.g., glasses, sunglasses, beards) and cluttered scenes, face detection has to be robust. Face detection is a critical step in the process of de-identification. Note that privacy might be compromised in video sequences if face detection fails in a single frame (and consequently it is not de-identified), so one of the directions of research is the development of robust and effective face detectors. Use of state-of-the-art detectors, such as face detectors for unconstrained face detection based on features called Normalized Pixel Difference (NPD) (Liao, Jain, & Li, 2016), detectors based on deep convolutional networks (Simonyan & Zisserman, 2015; Zhang, Zou, He, & Sun, 2016), hierarchical dense structures (Wen, Lei, Lyu, Li, & Yang, 2016), and social context (Qin & Shelton, 2016) can guarantee very low false negative detection, approaching zero. On the other hand, due to the requirement for the naturalness of de-identified videos, there is an additional demand for low false positive detections. A combination of face detection and tracking, i.e., a combination of spatial and temporal correspondence among frames, can improve the effectiveness of detection and localization of faces, and can help remove false positive detections.

In this paper, a novel multi-agent approach to the first two stages of pipeline de-identification (multi-face detection and tracking) in video sequences is described. Based on the concepts of multi-agent architecture for dynamic systems, a two-level hierarchical organization of a multi-face tracking system is proposed. It is built based on extended BDI (Belief Desire Intention) agents, i.e. extended agents with mental attitudes. It consists of a manager Extended BDI (mEBDI) agent at the first level and multiple tracker Extended BDI (trEBDI) agents at the second level.

The following might be considered the main contributions of this work: (i) a new multi-agent dynamic system architecture based on extensions of the BDI concepts for intelligent applications such as complex computer vision tasks; (ii) a logical representation model of relations and mental attitudes of agents involved in a multi-agent dynamic system; (iii) the integration of an agent with mental attitudes and autonomy-oriented entities for specific problem solving; (iv) adaptation of the proposed architecture to online multi-face tracking in video sequences; (v) the integration of state-of-the-art methods for face detection based on deep learning and online robust tracking with BDI agents; (vi) experimental verification of the proposed approach.

The paper is organized as follows. Section 2 describes the background of the BDI model agency and related work in the field of object detection and tracking based on multi-agent concepts. Section 3 includes a formal definition of a model of MADS and its adaptation to face detection and multi-face tracking tasks. Section 4 provides details of the architecture of a MADS for face detection and multi-face tracking and the logical representation of relations and mental attitudes of the extended BDI agents. Section 5 describes the functions of the extended BDI agents in MADS during initialization, regular face tracking, and exception. Section 6 gives the implementation details related to specific functions as extensions of the BDI agents – face detection, elimination of false positive detections, and face loss detection based on state-of-the-art methods. Section 7 describes an implementation experimental setup and presents the results of the experiments and their evaluation. Section 8 deals with adaptation of MADS to different classes of problems. In the conclusion, some final remarks and further directions for research are given.

2. Related work

2.1. Background

The multi-agent system presented in this paper is built on the BDI (Belief Desire Intention) model of agency, more precisely on the extended BDI agent with autonomous entities (Maleš & Ribarić, 2016). In the first part of this section, the foundations of BDI agents are given, as well as the idea behind autonomy-oriented entities. In the second part, an overview is presented of the research work related to agent and/or multi-agent approaches to object detection and tracking.

The BDI model was initially introduced by Bratman (1987) and then refined by Rao and Georgeff (1991) and Rao and Georgeff (1995) for real implementation in agent-based systems. A BDI agent is a cognitive agent and has mental attitudes: beliefs, desires and intentions. Beliefs represent the informational state of a BDI agent, i.e. what it knows about itself and its environment. Desires (or goals) are its motivational state, that is, what the agent prefers and wants to achieve. Intentions represent the deliberative state of the agent and they tend to lead to action. Logical foundations of agency have been established since the late 1980s, but nevertheless multi-agent system models built on logic are still popular. Since then, a number of logical theories of agency have been developed (Dix & Fisher, 2011; Fisher, Bordini, Hirsch, & Torroni, 2007). The reason is that logic can be a powerful tool for reasoning about multi-agent systems (van der Hoek & Wooldridge, 2012). Van der Hoek and Wooldridge present three reasons for this. The first is that logics provide a language for specification of the properties of an agent, a group of agents, and of the environment. The second reason is that such properties are expressed as a logical formula that forms part of some inference system and can be used to deduce other properties. The third reason is that logics provide a formal semantics in which sentences from the language are assigned a precise meaning.

The BDI agents balance pro-active goal seeking behaviour and reactive responses to changes in the environment (Singh, Padgham, & Logan, 2016). They have a hierarchical plan library of pre-defined plans. Each plan has conditions under which certain circumstances will be executed to fulfil a goal in a given situation (Singh, Sardina, Padgham, & Airiau, 2010). When an unexpected event occurs, often because the environment has changed, agents backtrack and implement a different strategy for the new situation. In this sense, BDI agents are adaptive. A limitation of BDI agents is their lack of learning capabilities, i.e. they are unable to learn new behaviours from their experience (Airiau, Padgham, Sadrina, & Sen, 2008). The BDI agents are inspired by the human concept of knowledge and deliberation which makes them easy to understand for humans. On the other hand, the process of logical computation is often intractable, undecidable and brittle (Van Dyke Parunak, Nielsen, & Brueckner, 2006). At the implementation level, a BDI agent can be represented by the following structure (Meng, 2009):

```
<BDI Agent> {<Beliefs>
            Constrains; Data Structures;
            <Desires>
            Values; Conditions; Functions;
            <Intentions>
            Methods; Procedures}
```

Completely different types of agents are autonomy-oriented entities. They are inspired by swarm intelligence (Dorigo & Stützle, 2010; Garnier, Gautrais, & Theraulaz, 2007; Kennedy, Eberhart, & Shi, 2001). They emulate animal behaviour where the emphasis is not on individual behaviour but on society as a whole, which might be described as intelligent. An internal representation is numerical so they use optimization methods for exploring parameter space. This approach makes them computationally efficient on one hand, and on the other they are difficult to understand from the human standpoint; for example, knowledge is hidden on the weight between two nodes in neural networks or in the strength of the pheromone of an ant trail (Dorigo & Stützle, 2010). Systems based on nature-inspired computing (NIC) utilize autonomous entities that self-organize to achieve the goals of system modelling and problem solving (Liu & Tsui, 2006). Besides the characteristic of self-organization, NIC-based systems are autonomous, distributed, emergent, and adaptive. Autonomy-oriented computing (AOC) (Liu, Jin, & Tsui, 2004) is a concrete manifestation of the NIC paradigm applied in the field of computer science that explores metaphors and models of autonomy offered in nature. The AOC system is a multi-agent system (MAS). It has the characteristics of self-organization, self-organized computability, interactivity, and computational scalability in solving large-scale computationally hard problems or modelling complex systems (Yang, Liu, & Liu, 2010). In AOC, computation is based on autonomy-oriented agents or entities (AOEs). They spontaneously interact with their local environments, self-organize their structural relationships, and operate based on their behavioural rules. This process is known as self-organization and it is the core of AOC (Liu, 2008).

The integration of different types of agents such as BDI agents and autonomy-oriented entities in a single system might be a challenge, since they possess different cognitive levels. In the proposed approach, these two types of agents are integrated in a single architecture of a manager EBDI agent.

2.2. Use of agents and multi-agent systems for computer vision applications (object detection and tracking)

Graf and Knoll (2000) proposed a multi-agent system architecture dedicated to the model of computer vision systems. The main aim of the proposed architecture was to provide a decentralized computer vision system with a high degree of flexibility. The basic idea of the architecture is to model a vision system as a society of autonomous agents, where each of them is responsible for a specific vision task. The authors defined two types of agents: (i) a master agent which contains an inference engine, general and individual knowledge, working memory, and a communication module; and (ii) a slave agent with very simple architecture consisting of processing functions, a communication module, and rudimentary mechanisms for interpreting messages. As a test bed for the proposed approach, the authors adapted an object recognition system to multi-agent architecture.

The experiment was performed only for the static scene consisting of the simple overlapping two-dimensional objects (ledges and rims), and the recognition task for the multi-agent system was to recognize the objects that match the object specification (3-hole-ledges and two red rims). The system failed to detect one red rim (among six rims), but the authors claimed that this was not a problem of the multi-agent architecture but rather of an inaccurate feature extraction.

The idea to model a vision system as a society of two types of autonomous agent was promising but the distribution of responsibility between the master agent with an inference engine and simple slave agents with only processing and communication functions represents a drawback of the approach. The complex computer vision tasks require not only a distribution of functions but also a distribution of intelligence.

Kipčić and Ribarić (2005) described a multi-agent-based approach to face detection and localization in colour images. The assembly of agents consists of autonomous behaviour agents which are randomly distributed throughout the image. Based on the innate agents' behaviour functions, such as diffusion, self-reproduction and dying, the agents detect and mark skin-colour pixels using a combination of HSI and RGB colour models. The marked skin-colour pixels are candidates of face-like regions. Each of these regions is represented by an agent's family. Using the information about the shape of the agent's families, the final decision is made about regions which represent a human face. The approach was tested using both the XM2VTS database of frontal face images and images containing people in natural situations.

The experiments were performed on the dataset consisting of more than 580 images. The authors reported 94.5% of correct face detection and localization with 6% of the false acceptance rate. Using a set of autonomy-oriented agents based on nature-inspired computing can be effective but only for specific and relatively simple computer vision tasks (e.g., low-level image processing, the detection of predefined objects). The assembly of autonomous behaviour agents dedicated to specific functions can be a component of a system for solving complex tasks.

A multi-agent system for people detection and tracking using stereo vision in mobile robots is presented in Muñoz-Salinas, Aguirre, García-Silvente, Ayesh, and Góngora (2009). The multi-agent system provides a basic set of perceptual-motor skills useful for mobile robotic applications that are required to interact with human users. The multi-agent system architecture is hierarchical, based on a functional design, and has three levels: hardware managers, behaviours, and skills. Each level comprises a set of agents with different tasks. For example, the behaviour level comprises a set of agents that implement behaviours which are combined to create more complex ones (named skills). People detection and tracking are performed using stereo vision (obtained by a StereoCamera agent), a plan-view map representation of the data, and an agent which uses position and colour information.

The authors defined the experimental setup of five sets of experiments performed on their private database. The first three sets test the ability to detect persons, while the fourth and fifth evaluate person tracking. The authors report that the success of person tracking was between 74.2% and 92.4% (depending on the number of persons in the scene).

The proposed multi-agent system is tailored for specific mobile robot tasks which also include interaction with human users. The proposed architecture is robot oriented and is not generic enough to be implemented for wider computer vision applications.

BDI-based multi-agent reconfigurable architecture for real-time object tracking is proposed in Meng (2009). The author proposes reconfigurable computing, i.e. a reconfigurable system-on-chip (rSoC) platform with microprocessors and field-programmable gate arrays (FPGAs). To simplify the hardware/software interface, BDI multi-agent architecture is proposed as the unified framework. The system is decomposed into agents based on system specifications, where agents can accomplish some specific tasks (task graph partitioning, tracking) independently and can communicate with each other.

To evaluate the proposed approach, a tracking video experiment was conducted on the video sequence (frame resolution 320×240) in an office environment. The scene included one person, occlusion and situations where a person disappears, and reappears. The author did not give the quantitative results but only concluded that the "proposed algorithm showed the robustness to keep tracking the object all over the cases".

The approach of using BDI-based reconfigurable multi-agent architecture for real-time object tracking and its hardware implementation are promising, but the author did not provide quantitative results of the experiments.

Multi-agent architecture based on the BDI model for data fusion in visual sensor networks is shown in Castanedo, García, Patricio, Miguel, and Molina (2010). The proposed architecture performs tracking, data fusion, and coordination. The authors focus on how to fuse the tracks from different agents which are applied to the same object. The proposed Cooperative Sensor Agent architecture is based on the Procedural Reasoning System computational model, specifically on JADEX. It is composed of three different types of agents: surveillance-sensor agents, fusion agents, and interface agents. Surveillance-sensor agents and fusion agents possess beliefs, desires and intentions, while interface agents receive the fused data and show them to the final user.

The suitability of the proposed multi-agent system was evaluated in two different scenarios. The first one is given by indoor tracking in the laboratory using a recorded video file with three different cameras. The second one is performed using the APIDIS dataset (APIDIS basketball dataset) composed of 1500 frames of a basketball game acquired by seven 2-Megapixels cameras mounted above a basketball court. In the first scenario, the focus was on evaluating the tracking continuity of the target obtained using the fused values, and the mean absolute error of the fused values against manually annotated ground-truth values. In the second scenario, the performance of the system was evaluated when more visual sensors were used. The results of the exhaustive experiments are shown in a number of tables and diagrams (position/time instant) in which the distances between the positions of the tracked objects and the ground truth positions were expressed in cm.

The research was primarily oriented to fuse the tracks obtained from different agents which are applied to the same object. An approach which combines BDI agents at two levels (surveillance-sensor and fusion) and "ordinary" inference agent leads to the embryonic idea of a hybrid architecture.

In Gascuena and Fernandez-Cabalero (2011), the authors describe the use of agent technology in intelligent, multisensory and distributed surveillance. They list multisensory distributed surveillance systems which did not use agent technology and surveillance systems based on agent technology in the period 1997–2009.

A path-planning method for multi-human tracking by multiple agents based on long-term prediction is presented in Takemura, Nakamura, Matsumoto, and Ishiguro (2012). The aim is to obtain detailed information about human behaviours and characteristics. The objective of path-planning is to find paths for agents so that they will continue to follow humans at close range. The agent's paths are planned based on the similarity between the predicted positions of humans and the agent's field of view.

The authors conducted three simulation experiments where humans moved constantly, moved following two rules, switching from one to the other, and an experiment using real human movement data observed in a Kyoto subway station. The results of the experiments showed that the performance of human tracking could be kept high even in a changing environment.

A memory-based multi-agent model for tracking a moving object is presented in Wang, Qi, and Li (2013). Agents are randomly distributed near the located object region and mapped onto a 2D lattice-like environment for predicting the new location of an object by their co-evolutionary behaviours (competition, recombination, and migration). The three-stage human brain memory model (ultrashort-term memory, short memory, and long-term memory) is incorporated into a multi-agent co-evolutionary process for finding a best match of the appearance of the object.

The efficacy of the proposed multi-agent system was verified on their own video dataset. The first set of experiments was related to tracking a person with abrupt appearance changes. The second set was aimed at tracking persons who were occasionally occluded. The authors claim that the proposed method could deal with large appearance changes, as well as heavy occlusions.

The approach which integrates behavioral agents which are randomly distributed near the located object region (a similar approach described in Kipčić and Ribarić (2005) and a model of the human brain is interesting and promising but requires additional elaboration of the distribution of the functions of three-level memory organization.

In recent work, an agent-based framework for individual tracking in unconstrained environments is presented in Zaghetto, Aguiar, Zaghetto, Ralha, and de Barros Vidal (2017). The framework is composed of three different types of agents: face detector, face tracker, and manager. The face detector and tracker agents perform fully automatic single-sample face recognition, and track individuals using Viola-Jones and Speeded Up Robust Features (SURF) algorithms. A functional interaction model is based on the Contract Net Protocol and uses FIPA-ACL as the communication protocol. Agents communicate through a shared directory in the cloud.

The preliminary experimental results showed that the framework agents (face detector, face tracker, and manager agent) could adequately execute the tasks they were assigned, considering the detection, identification and tracking of individuals within an environment under surveillance. The quantitative results and metrics were not given.

The strengths of the proposed multi-agent intelligent system are the use of distributed and parallel processing and cloud storage services among detector, tracker and manager agents. The weaknesses are the fact that very simple methods are used for face detection (Viola-Jones), tracking and face recognition are based on SURF, and the testing procedure is performed for few people in an extremely simplified indoor scenario (non-moving camera and target).

Considering the pros and cons of the methods described in the related works, we propose a new architecture framework in the field of intelligent systems which is based on four paradigms: a novel multi-agent dynamic system architecture (MADS), an extended Belief Desire Intention (EBDI) agent community, autonomy-oriented entities (AOEs) and deep learning concepts. The main advantages of the proposed architecture are:

(i) A generic multi-agent dynamic system (MADS) architecture based on different types of agents (BDI and AOE), deep-learning concepts and conventional methods adaptable for a wide range of complex problems in the field of expert and intelligent systems. To illustrate this, we described an application of MADS in the field of computer vision (robust multi-face tracking).
(ii) Integration of extended BDI agents and autonomy-oriented behavioural agents to manage and evaluate outcomes obtained by the CNN module(s). Owning to the mental attitudes of a BDI agent, specified by the modal logic approach, it is possible to represent common-sense and expert knowledge and verify or correct the results (e.g., activation of AOEs when the confidence level of the CNN detector is below the threshold).
(iii) Multiple BDI agents at the lower level and assigning a specific function to each of them enable the simultaneous execution of tasks and supervision of the obtained outcomes (e.g., checking the tracker confidence value based on the Peak to Sidelobe Ratio (PSR)).
(iv) The proposed architecture allows integration of state-of-the-art methods controlled by the mental attitudes of agents (e.g., a CNN-based face detector, a tracker based on multi-

ple discriminative scale and space invariant correlation filters, and a detector of target loss).

The main disadvantages of the proposed architecture are:

(i) It requires both a specific domain knowledge to design the mental attitudes of agents and a training dataset to adjust the number parameters (e.g., the face detection confidence threshold, the PSR threshold, the maximum age for all AOEs ...).
(ii) Potentially intensive communication traffic between the agent(s) at the higher level and agent(s) at the lower levels in an exceptional situation (e.g., suspension of tracker agents when they lose the target, the appearance of new faces, and/or face re-detection).

The problem of multi-face tracking in unconstrained videos is still far from being resolved. The main reasons for this are the inability to detect faces in an unconstrained environment due to the multi-pose appearance of faces, changes of facial expressions, the presence of structural components and significant variations in the face scale and scene illumination in multiple shots. Short- and/or long-term facial occlusions and the multiple entering and exiting of persons in/out of the camera field of view remain challenging problems. Recently, the following developments have appeared: multi-face tracking systems with state-of-the-art performance which use a conventional computer vision approach based on discriminative correlation filters for translation and scale estimation (Danelljan, Häger, Khan, & Felsberg, 2017); a novel model of multi-face tracking and clustering of faces, and a re-identification algorithm which combines the co-occurrence model of multiple body parts to seamlessly create face tracklets, and recursively link tracklets to construct a graph for extracting clusters (Lin & Hung, 2018); and an approach (Marčetić & Ribarić, 2018) based on deep neural networks for face detection and face recognition of normalized detected faces (to minimize the number of ID switches).

3. Model of a multi-agent dynamic system (MADS)

3.1. MADS – general definition

There is no unique definition of the term "multi-agent system (MAS)". According to Weiss (1999), a multi-agent system is simply defined as "a system composed of multiple, interacting agents". Shoham and Leyton-Brown (2008) define MAS as a system which contains autonomy entities (agents) with different information and interests. Stone and Veloso (2000) define MAS as a loosely coupled network of problem-solving entities (agents) that work together to find answers to problems that are beyond the individual capabilities or knowledge of each entity (agent). Ferber defines the concept of MAS as a system which comprises the following elements (Ferber, 1999): an environment E, a set of objects O, an assembly of agents A, an assembly of relations R, an assembly of operations Op, making it possible for the agents of $A \subseteq O$ to perceive, produce, consume, transform and manipulate objects from O, and operators which specify tasks and reactions of the world. Based on Ferber's approach, we define a multi-agent dynamic system, which has the characteristics specified by Stone and Veloso, as follows.

In general, the model of a multi-agent dynamic system is defined as 11-tuple:

$$\text{MADS} = (\text{A}, \text{O}, \text{T}, \Theta, \text{R}, \rho, \alpha, \varepsilon, \chi, \text{R}_\text{T}, \nu),$$

where:

- A is a set of agents;
- O is a set of objects in a dynamic environment, $\text{A} \cap \text{O} = \emptyset$;
- T is a set of time points $\text{t} \in \text{T}$;
- Θ is a set of temporal intervals $\tau \in \Theta$, $\tau = [\text{t}_1, \text{t}_2]$, where $\text{t}_1 \leq \text{t}_2$; $\text{t}_1, \text{t}_2 \in \text{T}$ (when $\text{t}_1 = \text{t}_2$ temporal interval τ is converted in a time point);
- R is a set of relations with temporal constraint.

Other elements of the 11-tuple are defined below:

- ρ is a *temporal constraint relation function*: $\text{A} \cup \text{O} \times \text{A} \cup \text{O} \times \text{A} \cup \text{O} \times \Theta \rightarrow \text{R}$, which describes relations between agents and an object during a temporal interval $\tau \in \Theta$;

The functions α, ε, and χ describe the existence of elements of sets A, O and R, respectively, at each time point:

- α is an *agent existence function* α: $\text{T} \rightarrow \xi(\text{A})$, maps a time point t into one or more agents from the set A, where $\xi(\text{A})$ is a partition set of A, i.e. α defines the presence of the agents in the system at time t;
- ε is an *object existence function* ε: $\text{T} \rightarrow \xi(\text{O})$, maps a time point into one or more objects from the set O, where $\xi(\text{O})$ is a partition set of O, i.e. it defines the presence of the objects in the system at time t;
- χ is a *relation existence function* χ: $\text{T} \rightarrow \xi(\text{R})$, maps a time point into one or more relations from the set R, where $\xi(\text{R})$ is a partition set of R, i.e. it defines the existing relations in the system at time t;
- R_T is the set of relations between two temporal intervals or/and time points. The set R_T consists of the following temporal relations: before, meets, during, overlaps, starts, finishes, equal and their inversions;
- ν is a *temporal relation function*, ν: $\Theta \times \Theta \rightarrow \text{R}_\text{T}$, it maps a pair of temporal intervals or/and time points into a relation from the set R_T.

One of the basic characteristics of a dynamic system is its changing during time. Let $\text{t}_1, \ldots \text{t}_\text{i}, \ldots \in \text{T}$ be time points in which elements of sets A, O and R are observed. $\text{MADS} = (\text{A}, \text{O}, \text{T}, \Theta, \text{R}, \rho, \alpha, \varepsilon, \chi, \text{R}_\text{T}, \nu)$ is *dynamic* when $\exists\ \text{t}_\text{i}, \text{t}_\text{j} \in \text{T}$, $\text{t}_\text{i} \neq \text{t}_\text{j}$ and at least one of the following conditions is satisfied:

(i) $\alpha(\text{t}_\text{i}) \neq \alpha(\text{t}_\text{j})$;
(ii) $\varepsilon(\text{t}_\text{i}) \neq \varepsilon(\text{t}_\text{j})$; and
(iii) $\chi(\text{t}_\text{i}) \neq \chi(\text{t}_\text{j})$.

The model of the multi-agent dynamic system MADS is domain independent. The elements of MADS 11- tuple can be adapted and extended depending on a specific problem.

3.2. MADS adapted to a multi-face tracking system

For the purpose of adapting MADS for multi-face tracking in video sequences, the following modifications and new elements of MADS are introduced.

An agent can be any entity which has mental attitudes and is capable of deliberating, reasoning, making decisions, and acting autonomously. Set A is a finite set of agents where $\text{A} = \text{A}_\text{M} \cup \text{A}_\text{TR}$. There are two types of agents: manager agent $\text{a}_\text{i} \in \text{A}_\text{M}$, $i = 1, \ldots n$, called a manager Extended Belief Desire Intention (mEDBI) agent, and tracker agents $\text{a}_\text{j} \in \text{A}_\text{TR}$, $j = 1, \ldots, m$, called tracker Extended Belief Desire Intention (trEBDI) agents. The basic requirement is the existence of at least one manager agent and one tracker agent. Note that a manager agent mEBDI is extended by a set of autonomy-oriented entities or agents (AOEs) which are used for the elimination of the false positive detection of faces during the detection phase.

The objects $\text{o}_\text{i} \in \text{O}$; $i = 1, 2, \ldots$ are elements of MADS on which the agents act. In the proposed system, the objects are faces (face image patches) which are assigned to the tracker agents.

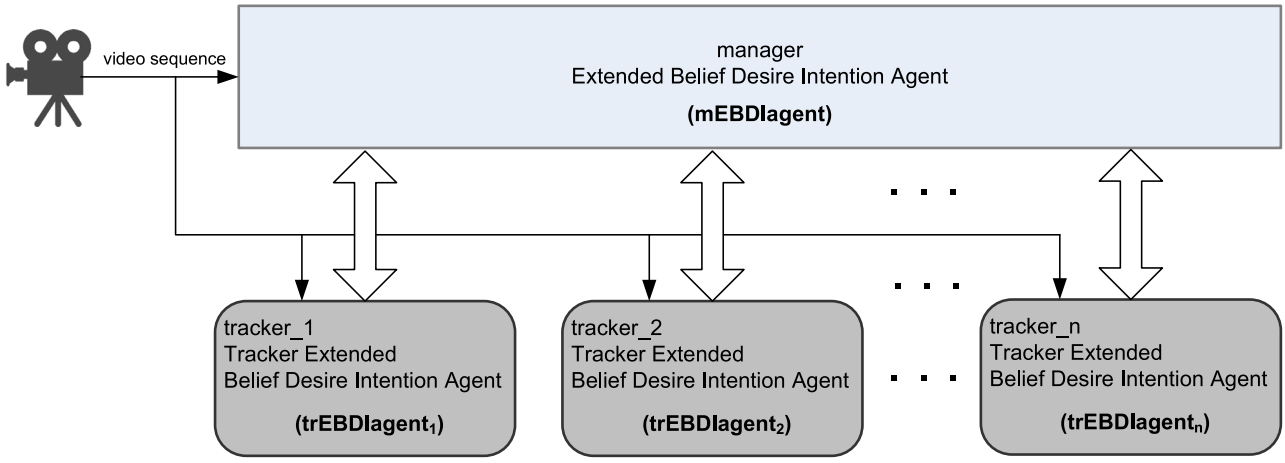

Fig. 1. The multi-agent architecture of a dynamic system.

Table 1
Manager agent, tracker agents, faces and corresponding subsets of relations $R = R_1 \cup R_2 \cup R_3 \cup R_4$.

Relation	Manager	$Face_i$	Tracker $agent_i$	Set
Detect	+	+	–	R_1
Create	+	–	+	R_3
Assign	+	+	+	R_2
Inform_t	+	–	+	R_3
Inform_m	+	–	+	R_3
Activate	+	–	+	R_3
Track	–	+	+	R_4
Validate	–	+	+	R_4
Lost	+	+	+	R_2
Redetect	+	+	+	R_2
Suspend	+	–	+	R_3
Re-track	–	+	+	R_4
Update_m	+	+	–	R_1
Update_t	–	+	+	R_4

Temporal constraint relations from set R describe the relationships among the manager agent, a tracker agent and an object (face) in time. Only relations relevant to a problem domain are taken into consideration as follows: set R, which is defined as $R = R_1 \cup R_2 \cup R_3 \cup R_4$, where the elements of the subset R_1 are relations between the manager agent and a face; elements of the subset R_2 are relations among the manager agent, a face and a tracker agent; elements of the subset R_3 are relations between the manager agent and a tracker agent; and elements of the subset R_4 are relations between a face and a tracker agent. Table 1 represents agents, objects, and corresponding subsets of relations R_i, $i = 1, 2, \ldots, 4$. Note that each relation has a temporal constraint defined by Θ, which means that the relation holds in a specific time interval τ. In this specific context of multi-face tracking in videos, a time interval τ^j which corresponds to a frame j, is

$$\tau^j = [j/fps, (j+1)/fps],$$

where fps denotes frames per second.

The interval τ^j consists of the (sub)intervals $\tau_k{}^j$:

$$\tau^j = \sum_{k=1}^{4} \tau_k^j$$

Four (sub)intervals correspond to procedure steps performed during face tracking.

Consequently, instead of one *temporal constraint relation function* ρ (defined in Section 3.1) for the specific system, four temporal constraint relation functions are modified and defined:

$$\rho_1 : A_M \times O \times \Theta \rightarrow R_1,\ \rho_2 : A_M \times O \times A_{TR} \times \Theta \rightarrow R_2,$$
$$\rho_3 : A_M \times A_{TR} \times \Theta \rightarrow R_3 \text{ and } \rho_4 : O \times A_{TR} \times \Theta \rightarrow R_4.$$

In addition, the MADS model is extended with five functions:

- *face position* fp: $O \rightarrow P$, which maps a set of face image patches O into a set of positions $P \subset \mathbb{N} \times \mathbb{N}$, where the elements of P are positions $(x, y); x, y \in \mathbb{N}$ of face image patches from O in a frame with the resolution m by n; n, $m \in \mathbb{N}$;
- *face image patch size* fs: $O \rightarrow S$, $s \in S \subset \mathbb{N}$ is a side length of a square patch where a face is detected;
- *detector confidence* dc: $O \rightarrow [0, 1]$, which defines face detector assurance that a face image patch contains a face;
- *face label* fl: $O \rightarrow \mathbb{N}$, which maps a face image patch (which contains a face) to a unique identity label (index);
- *tracker confidence* tc: $O \rightarrow [0, 1]$ which defines face tracker assurance that a tracked face image patch contains a face.

The model of a multi-agent dynamic system for multi-face tracking in video sequences is defined as:

$$MADS^* = (A, O, T, \Theta, R, \rho, \alpha, \varepsilon, \chi, R_T, \nu, fp, fs, dc, fl, tc).$$

The dynamic of MADS* is mirrored in the fact that at least one of the conditions (i)–(iii) (see Section 3.1) is satisfied, with an additional changing of a *face position* fp, *face_size* fs, *detector confidence* dc, *face label* fl and *tracker confidence* tc.

4. Architecture of a multi-agent dynamic system for multi-face tracking

The architecture of a multi-agent dynamic system is represented as a two-level hierarchical organization (Fig. 1). At the first level there is a manager designed as an Extended Belief Desire Intention (mEBDI) agent. The extension of BDI consists of a face CNN detection module, a set of autonomous oriented entities (AOEs), and a trajectory memory (mTM).

The mEBDI agent is responsible for the management of a whole multi-agent face tracking system: face detection, elimination of false positive face detections, the initialization of multi-face tracking, the creation and activation of tracking agents (trEBDI), and the maintenance of face trajectories of all tracked faces. When a tracking failure occurs, the mEBDI agent re-initializes face detection, and, based on the results of the detections, assigns a detected face to the corresponding tracking agent trEBDI, or creates a new tracking agent or agents.

At the second level, there are a number of tracking agents (trEBDIs) which consist of a basic BDI agent extended with an integration module. The integration module consists of a tracker based on position and scale correlation filters (DSST), a visual appearance memory (VAM), and a trajectory memory (TM).

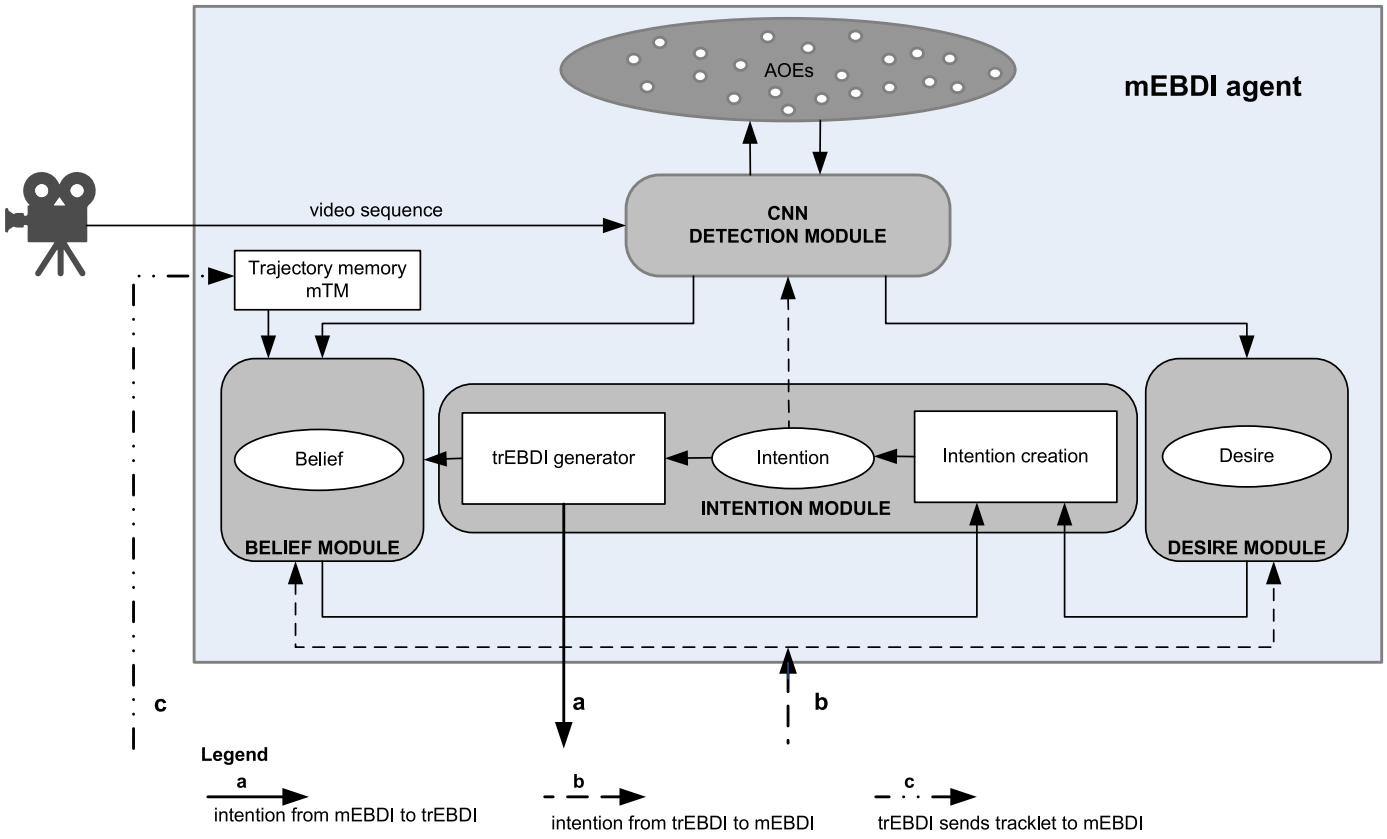

Fig. 2. The architecture of a mEBDI agent.

4.1. Manager agent (mEBDI)

The architecture of the mEBDI agent is shown in Fig. 2. It consists of three basic modules which are common to all types of BDI agents (Belief, Desire and Intention), a convolutional neural network (CNN) detection module, a set of autonomy-oriented entities (AOEs), and a trajectory memory (mTM). The mEBDI acts in the following manner. The detection module based on the CNN detects faces in the frames of a video sequence and then invokes autonomy-oriented entities (AOEs) which eliminate false positive face detections. The behaviour of the AOEs is the same as in Kipčić and Ribarić (2005). After the elimination of false positive face detections, the CNN detection module sends information to the belief and the desire modules. Specifically, the CNN detection module sends the following parameters (face data): the detector confidence value, the position (x, y) of a square image patch where a face is detected in a frame, the side length s of this square patch, a face label, and the time of detection (see the definition of functions in Section 3.2).

The initialization procedure of the system starts with face detection and the elimination of false positive faces, i.e. vague face region candidates for which the CNN score (the detector confidence level) is below the experimentally determined threshold which is 0.72. The threshold for the CNN score is determined by the video training subset consisting of three YouTube music videos (Pussycat Dolls, Bruno Mars, Darling), which are excluded from the process of the experimental evaluation of the proposed system.

Then, the initialization procedure continues with the following activities (see Section 5.1): the mEBDI agent believes that faces are detected in the frame in a video sequence and it desires to create the trEBDI agents, to assign the detected faces to the trEBDI agents, and, finally, to activate them. The intention module uses the beliefs and desires and creates an intention. At the end of the initialization, the corresponding face data are sent to a belief module of each trEBDI agent, and a trEBDI agent is activated. The mEBDI agent stores the list of face trajectories in a trajectory memory (mTM) for all trEBDI agents. The list will is used in the following situations: (i) regular face tracking; (ii) tracking exception when trEBDI agents lose tracked faces; and (iii) when the final results of face tracking are required. Besides initialization, the mEBDI agent supervises all trEBDI agents. When the trEBDI agent has lost the tracked face that is assigned to it, it informs the mEBDI agent by sending a message to the mEBDI agent's belief and desire modules. This message will stipulate the mEBDI agent to create an intention to redetect the lost face.

A component of the mEBDI agent called a trEBDI generator creates a trEBDI agent, assigns a face to it, and activates the trEBDI agent. In the phase of regular tracking and when the trEBDI agents lose their faces, the trEBDI generator transfers messages among the mEBDI agent and the trEBDI agents. A detailed description of the functions of the mEBDI agent is given in Section 5.

4.2. Tracker agent (trEBDI)

The architecture of a trEBDI agent consists of a belief module, a desire module, an intention module, and an integration module as an extension (Fig 3).

The mEBDI agent creates trEBDI agents using information about detected faces (face data). The trEBDI agents store this information in their belief modules. The activated trEBDI agent believes that the face is assigned to it, and it believes in the received face data. The trEBDI agent desires and intends to track the face. During regular tracking, the trEBDI agent believes and desires to track the face and therefore determines a tracker confidence value. The tracker confidence value is determined based on a normalized

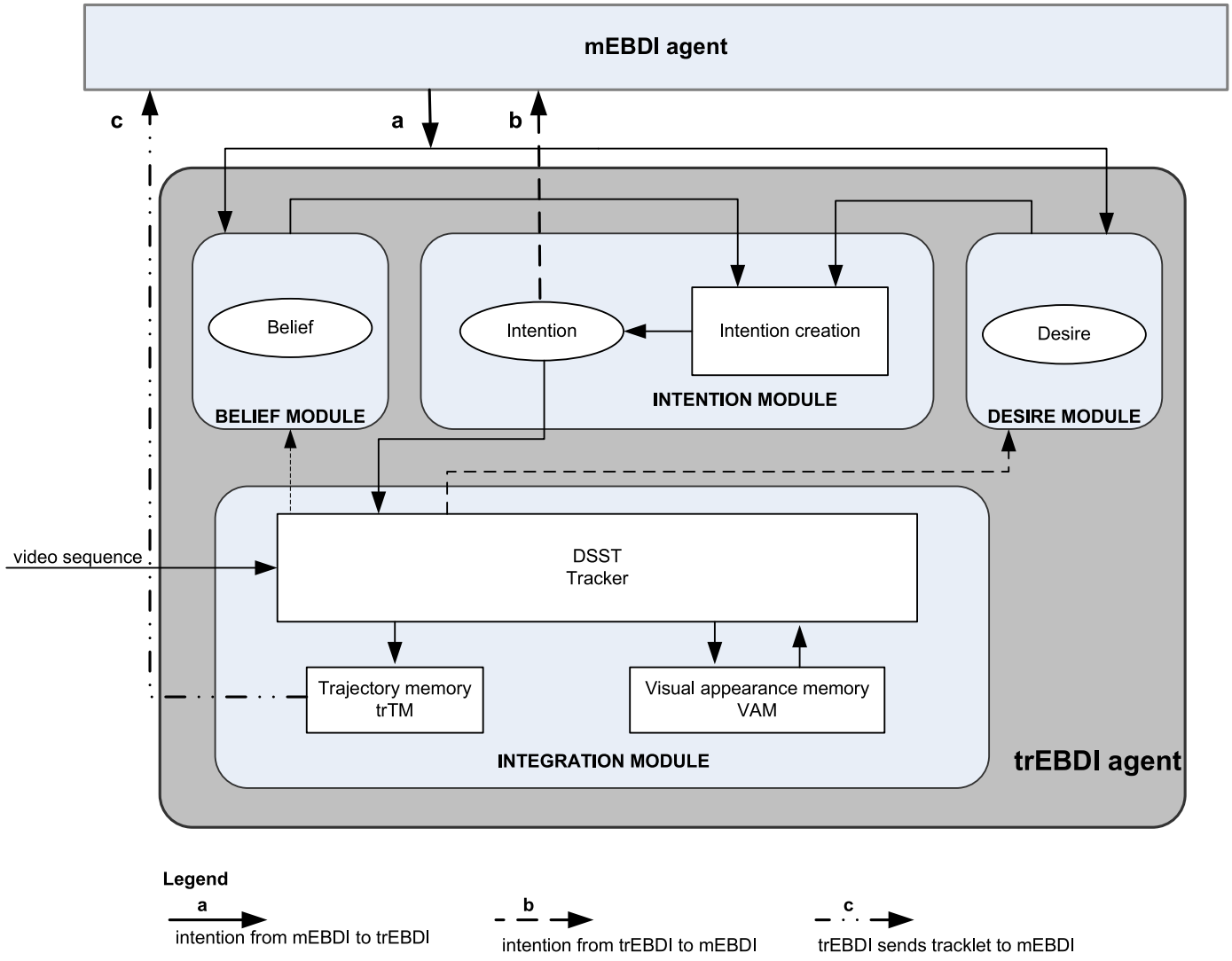

Fig. 3. The architecture of a trEBDI agent.

value of a Peak to Sidelobe Ratio (PSR) (Bolme, Beveridge, Draper, & Lui, 2010) (see Section 6.2). An agent intends to validate the tracker confidence value. When the trEBDI agent believes that the tracker confidence value is above the prescribed threshold, it intends to update its own beliefs and desires with the face data, and then intends to inform the mEBDI agent by sending it the face data (see Section 5.2).

The elements of the integration module are an "off-the-shelf" Discriminative Scale Space Tracking (DSST) tracker (Danelljan et al., 2017), a visual appearance memory (VAM), and a trajectory memory (trTM). A VAM stores visual appearance information about the tracked face represented by F-HOG features (Felzenszwalb, Ross, McAllester, & Ramanan, 2010). The visual appearance information is used: (i) to support regular tracking by using circular correlation between the stored F-HOG features and the features obtained from the current frame; and (ii) to determine a tracker confidence value based on the normalized PSR score. A trTM stores information about recent face positions and sizes. The intention module induces the DSST tracker to operate in the integration module. Every 1/fps (see Section 3.2), a new frame from a video sequence is presented to the DSST tracker that estimates a new face position and the size parameters and sends this information to the mEBDI.

When a trEBDI agent has lost a face (it believes in that), it desires and intends to inform the mEBDI agent about this. It sends its beliefs and desires to the mEBDI's belief and desire modules (face data are a component of the trEBDI agent's beliefs). The trEBDI agent changes its state from an active to a suspended state. It remains in the suspend state until the mEBDI redetects the face and associates it with the trEBDI agent that had previously tracked that face. With these new beliefs and desires, the mEBDI initiates the redetection process. The mEBDI agent initiates CNN face detection and the elimination of false positive face detections by AOEs. A detailed description of the functions of a trEBDI agent is given in Section 5.

4.3. Interaction among the mEBDI and trEBDI agents

Interaction among the mEBDI agent and trEBDI agents is performed through cooperation and communication. The mEBDI agent and trEBDI agents communicate by interchanging information about their beliefs and desires. They create intentions that induce the transfer of beliefs and desires from one agent to another. When an agent (mEBDI or trEBDI) has an intention to inform an agent of another type (mEBDI to trEBDI and/or trEBDI to mEBDI), it sends its beliefs and desires, where the face data are a component of the agent's beliefs. The first transfer of face data is done at the end of the initialization of the multi-face tracking system (see Section 5.1)

when the mEBDI needs to transfer its beliefs about the detected faces to trEBDI agents and activate them to track those faces.

During regular tracking, trEBDI agents send information to the mEBDI agents. The mEBDI agents store information in the trajectory memory (mTM) (see Section 5.2).

If some of the active trEBDI agents have lost their tracked faces, they inform the mEBDI agent. Consequently, the mEBDI agent first starts a new detection procedure and then the elimination of false positive face detections. The results are detected faces. Based on information stored in the trajectory memory (mTM), the mEBDI agent believes which face belongs to which active trEBDI agent, and the remaining faces are declared as unresolved.

The mEBDI agent sends messages about all the unresolved faces to all suspended trEBDI agents. The mEBDI agent and the suspended trEBDI agents exchange a few messages with information about the unresolved detected faces. All suspended trEBDI agents determine a tracker confidence value for all unresolved faces and send a message containing this value to the mEBDI agent. Depending on the received tracker confidence value, the mEBDI agent concludes on the assignment of a face identity label to the suspended trEBDI agents and sends messages to them. The content of the message enables some trEBDI agents to continue tracking faces for which the tracker confidence value is above the threshold. Other trEBDI agents with a tracker confidence value below the threshold remain suspended (see Section 5.3). New trEBDI agents are initialized for the remaining unresolved faces.

All the above-described activities of the mEBDI agent and trEBDI agents are given in the details in Section 5.

4.4. Logical representation of relations and mental attitudes of agents

The mEBDI agent and trEBDI agents have mental attitudes (beliefs, desires and intentions) that are denoted by modal logic operators: *BEL*, *DES* and *INT*. The agent's mental attitudes are represented by two formulas (Maleš & Ribarić, 2016):

(i) φ – the agent mental attitude content formula; and
(ii) ψ – the agent mental attitude formula.

The agent *mental attitude content formula* φ is defined as:

$$\varphi := P(v_1, \ldots, v_n) | \neg\varphi | (\varphi_1 \wedge \varphi_2) | \forall v_i \varphi,$$

where a predicate $P(v_1, \ldots, v_n)$ is an atomic formula. The terms $v_1, \ldots, v_n$ are: (i) constants – mEBDI agent; (ii) variables – trEBDI agents {$trEBDI_1$, $trEBDI_2$, ...}, faces {$face_1$, $face_2$, ...}; (iii) temporal variables (time points t or temporal intervals τ); and iv) functions – fp, fs, dc, tc and fl.

The formula φ defines the content of an agent's mental attitudes, e.g. beliefs, desires and intentions. In order to define the *agent's mental attitude,* the formula ψ is defined as:

$$\psi := BEL\varphi | DES\varphi | INT\varphi | \neg\psi | (\psi_1 \wedge \psi_2) | \forall v_i \psi,$$

where *BEL, DES* and *INT* are modal operators and φ is a formula.

The mEBDI and trEBDI agents create intentions according to their beliefs and desires. The form of the rule for creating an intention is $BEL\,\varphi \wedge DES\,\varphi \rightarrow INT\,\varphi$.

An agent's mental attitudes have temporal constraints, i.e. the temporal duration of an agent's beliefs, desires and intentions are written in the formula φ. The modal operator *BEL* satisfies these properties: the necessity rule and axioms K, D and T (Blackburn & van Benthem, 2007). The modal operator *DES* and *INT* satisfy these properties: necessity rule and axioms K and D.

Interpretation of the formulas φ and ψ are given below.

A predicate $P(v_1, \ldots, v_n)$ is further defined depending on a number of arguments as an *n*-place predicate:

- for $n = 2$ the terms are temporal variables (defined by the set R_T).

The following predicates are defined on the elements of the set R:

- for $n = 3$ the terms are (mEBDI agent, $face_i$, temporal variable), (mEBDI agent, $trEBDI_i$ agent, temporal variable) and ($face_i$, $trEBDI_i$ agent, temporal variable);
- for $n = 4$ the terms are the mEBDI agent, faces, $trEBDI_i$ agents and temporal variables.

The terms of the 6-place predicate are faces, functions fp, fs, dc or tc and fl (see Section 3.2) and a temporal variable.

The elements of the set R_T are relations between the temporal variables (where τ is an interval and t is a time point) and they are written as a 2-place predicate: Meets(τ_1, τ_2), Before(τ_1, τ_2), During(τ_1, τ_2), Overlap(τ_1, τ_2), Starts(τ_1, τ_2), Finishes(τ_1, τ_2) and Equal(τ_1, τ_2); Meets(t, τ), Before(t, τ), During(t, τ), Starts(t, τ) and Finishes(t, τ); Before(t_1, t_2) and Equal(t_1, t_2) (Allen, 1983).

The elements of the set $R = R_1 \cup R_2 \cup R_3 \cup R_4$ are relations among the mEBDI agent, faces, trEBDI agents that hold during the temporal interval τ or time point t and are written as a 3-place or 4-place predicate.

The 3-place predicates are: Detect(mEBDI, $face_i$, τ), Create(mEBDI, $trEBDI_i$, τ), Inform_m(mEBDI, $trEBDI_i$, τ), Inform_t(mEBDI, $trEBDI_i$, τ), Activate(mEBDI, $trEBDI_i$, τ), Suspend(mEBDI, $trEBDI_i$, τ), Track(face, $trEBDI_i$, τ), Validate(face, $trEBDI_i$, τ), Re-track($face_i$, $trEBDI_i$, τ), Update_m(mEBDI, $face_i$, τ) and Update_t($face_i$, $trEBDI_i$, τ). The interpretations of the predicates are:

- Detect(mEBDI, $face_i$, τ) – the mEBDI has detected a $face_i$ during interval τ;
- Create(mEBDI, $trEBDI_i$, τ) – the mEBDI creates a $trEBDI_i$ agent during interval τ;
- Inform_t(mEBDI, $trEBDI_i$, τ) – the mEBDI agent informs a $trEBDI_i$ agent about face data during τ, i.e. it sends information about a face to the trEBDI agent;
- Inform_m(mEBDI, $trEBDI_i$, τ) – a $trEBDI_i$ agent informs the mEBDI agent about face data during τ, i.e. it sends elements of face data (tracklet) to the mEBDI agent;
- Activate(mEBDI, $trEBDI_i$, τ) – the mEBDI agent activates a $trEBDI_i$ agent during τ;
- Suspend(mEBDI, $trEBDI_i$, τ) – a trEBDI agent is suspended during τ and the mEBDI agent is informed
- Track($face_i$, $trEBDI_i$, τ) – a $face_i$ is tracked by $trEBDI_i$ during τ;
- Validate($face_i$, $trEBDI_i$, τ) – a $trEBDI_i$ agent has evaluated a tracker confidence value of a $face_i$ during τ and the degree of belief is above the threshold;
- Re-track($face_i$, $trEBDI_i$, τ) – a $face_i$ is tracked again by a $trEBDI_i$ agent during τ;
- Update_m(mEBDI, $face_i$, τ) – the mEBDI agent updates tracklets during τ, obtained by a $trEBDI_i$ agent for a successfully tracked $face_i$; and
- Update_t($face_i$, $trEBDI_i$, τ) – the $trEBDI_i$ agent updates a tracklet during τ for a successfully tracked $face_i$.

The 4-place predicates are: Assign(mEBDI, $face_i$, $trEBDI_i$, τ), Lost(mEBDI, $face_i$, $trEBDI_i$, τ), and Redetect(mEBDI, $face_i$, $trEBDI_i$, τ). The interpretations are:

- Assign(mEBDI, $face_i$, $trEBDI_i$, τ) – the mEBDI agent assigns a $face_i$ to $trEBDI_i$ agent during τ;
- Lost(mEBDI, $face_i$, $trEBDI_i$, τ) – the mEBDI agent has been informed that the $trEBDI_i$ agent lost the tracked $face_i$ during τ;
- Redetect(mEBDI, $face_i$, $trEBDI_i$, τ) – the mEBDI agent has redetected during τ the $face_i$ that had been assigned to the $trEBDI_i$ agent and that had been lost.

The 6-place predicates are Face_data_d($face_i$, fp, fs, dc, fl, τ) and Face_data_t($face_i$, fp, fs, tc, fl, τ). They relate a $face_i$ (or a face

image patch) with a position, size, detector confidence value or tracker confidence value, and a face identity label during τ (see Section 3.2).

The mEBDI agent obtains a detector confidence value dc from the CNN detector, while the trEBDI agent calculates the tracker confidence value tc. Both types of agents include only own confidence values at the Face_data predicate.

For example, the sentence "The agent mEBDI has detected $face_1$ during interval τ at position fp ($face_1$), with size fs ($face_1$), detector confidence value dc ($face_1$), and a face identity label fl($face_1$)" is written as the formula:

$$\varphi := \text{Detect}(\text{mEBDI}, face_1, \tau) \wedge \text{Face\_data\_d}(face_1, \text{fp}(face_1), \text{fs}(face_1), \text{dc}(face_1), \text{fl}(face_1), \tau).$$

The content of the mEBDI agent's beliefs, desires and intentions are the relations between it and the faces (the set R_1) and between it and the trEBDI agents (the set R_3). The mEBDI agent also believes, desires and intends to achieve relations among it, the faces and the trEBDI agents (the set R_2).

The trEBDI agent believes, desires and intends to achieve relations between it and the mEBDI agent (the set R_3) and between it and a face (the set R_4). The trEBDI agent also believes, desires and intends to achieve relations among it, the mEBDI agent and a face (the set R_2).

The mEBDI agent and the trEBDI agent believe in the temporal order (the set R_T) and in the relation between a face and its position in a video frame, its size and the value of detector confidence or tracker confidence. Both agents have no desires or intentions to achieve the temporal order and the information about face data.

Example 1. The formula for the sentence "The mEBDI agent believes that it has detected a face during interval τ at position fp(face), with size fs(face), detector confidence value dc(face) and face identity label fl(face)" is written as:

$$\psi := BEL(\text{Detect}(\text{mEBDI}, face, \tau) \wedge \text{Face\_data\_d}(face, \text{fp}(face), \text{fs}(face), \text{dc}(face), \text{fl}(face), \tau)).$$

Example 2. The formula for the sentence "The trEBDI_i agent desires to track $face_i$ during interval τ" is written as $\psi :=$ *DES*(Track($face_i$, trEBDI_i, τ)).

Example 3. The formula for the sentence "The mEBDI agent intends to assign $face_i$ to trEBDI_i during interval τ" is written as $\psi :=$ *INT*(Assign(mEBDI, $face_i$, trEBDI_i, τ)).

Example 4. "The mEBDI agent believes that $face_i$ which trEBDI_i has lost is redetected during τ and believes in the information about the $face_i$ position, size and a detector confidence value. It also believes that interval τ_1 meets interval τ_2. The mEBDI agent desires to inform the trEBDI_i agent during τ_4 and intends to inform trEBDI_i during τ_5 about its beliefs". The rule is written as

- *BEL*(Redetect(mEBDI, $face_4$, trEBDI_4, τ_4) ∧ Face_data_d($face_4$, fp($face_1$), fs($face_4$), dc($face_4$), fl($face_1$), τ_4) ∧ Meets(τ_4, τ_5)) ∧ *DES*(Inform_t(mEBDI, trEBDI_4, τ_4) → *INT*(Inform(mEBDI, trEBDI_4, τ_5)).

5. Functions of mEBDI and trEBDI agents – initialization, regular tracking, and exception

In the process of multi-face tracking, the proposed multi-agent-based system can be in the following main states: initialization, regular tracking, and exception.

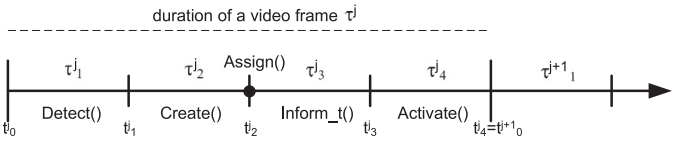

Fig. 4. Intentions of the mEBDI agent during initialization and corresponding time intervals.

5.1. Initialization of the multi-face tracking system

For the first frame of the video sequences ($j = 0$), all trEBDI agents are inactive. The manager agent mEBDI performs the following steps:

1. Face detection – activates the CNN-based face detector;
2. Elimination of false positive face detections – activates the autonomy-oriented entities (AOEs);
3. Assignment of a face identity label to each face image patch;
4. Initialization of the trEBDI agents:
   - creates trEDBI agents;
   - assigns each face to one of the trEBDI agents;
   - sends a message with information about a face image patch (face identity label, position and size of a face image patch) to the created trEBDI agents;
   - activates trEDBI agents (starts tracking).

Note that if there is no face detection in the first frame of a video sequence, then step 1 above is repeated on the next frame(s) – until a face or faces are detected.

The initialization at a frame j is specified by means of logical rules as follows.

mEDBI agent:

- $\bigwedge_{i=1}^{n}$ *BEL*(Detect(mEBDI, $face_i$, τ_1^j) ∧ ¬Assign(mEBDI, $face_i$, trEBDI_i, τ_1^j)) ∧ Face_data_d($face_i$, fp($face_i$), fs($face_i$), dc($face_i$), fl($face_i$), τ_1^j)) ∧ Meets(τ_1^j,τ_2^j)) ∧ *DES*(Create(mEBDI, trEBDI_i, τ_1^j)) → *INT*(Create(mEBDI, trEBDI_i, τ_2^j));
- $\bigwedge_{i=1}^{n}$ *BEL*(Create(mEBDI, trEBDI_i, τ_2^j) ∧ Finishes(t_2^j,τ_2^j)) ∧ *DES*(Assign(mEBDI, $face_i$, trEBDI_i, τ_2^j)) → *INT*(Assign(mEBDI, $face_i$, trEBDI_i, t_2^j));
- $\bigwedge_{i=1}^{n}$ *BEL*(Assign(mEBDI, $face_i$, trEBDI_i, t_2^j) ∧ Face_data_d($face_i$, fp($face_i$), fs($face_i$), dc($face_i$), fl($face_i$), τ_1^j) ∧ Meets(τ_1^j,τ_2^j) ∧ Finishes(t_2^j,τ_2^j) ∧ Meets(τ_2^j,τ_3^j)) ∧ *DES*(Inform_t(mEBDI, trEBDI_i, t_2^j) → *INT*(Inform_t(mEBDI, trEBDI_i, τ_3^j));
- $\bigwedge_{i=1}^{n}$ *BEL*(Inform_t(mEBDI, trEBDI_i, τ_3^j) ∧ Meets(τ_3^j,τ_4^j)) ∧ *DES*(Activate(mEBDI, trEBDI_i, τ_3^j)) → *INT*(Activate(mEBDI, trEBDI_i, τ_4^j)).

where $i = 1, 2, \ldots, n$ is the index of the detected face image patch, and n is the total number of detected faces.

The intentions of the mEBDI agent during initialization of the multi-face tracking system and corresponding time intervals τ_k^j; $k = 1, \ldots, 4$, are shown in Fig. 4.

5.2. Regular tracking

Each active trEDBI agent performs the following steps during regular tracking:

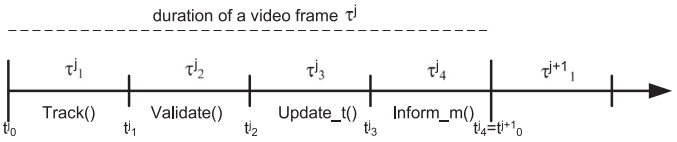

Fig. 5. Intentions of the trEBDI agents during regular tracking and corresponding time intervals.

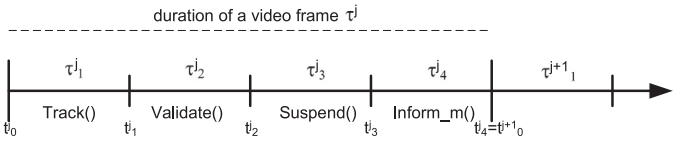

Fig. 6. Intentions of the trEBDI agents when they lose assigned faces and the corresponding time intervals.

1. Activation of the DSST tracker for the current frame – data extraction (a new position and size of a face image patch), and determination of a tracker confidence value;
2. Evaluation of the degree of belief (based on the tracker confidence value) that the face image patch contains the tracked face – the degree of belief is above the threshold for regular tracking;
3. Storing the new elements of a tracklet in its trajectory memory (trTM) – a new position and size of the tracked face;
4. Message sending – a trEDBI sends a message with new elements of a tracklet to the mEDBI agent, and it remains in an active state.

The mEBDI agent waits for messages from active trEBDI agents and updates the corresponding tracklets stored in its trajectory memory (mTM).

The description of the above steps at the logical level is as follows:

– *BEL*(Track(face$_i$, trEBDI$_i$, τ_1^j) ∧ Face_data_t(face$_i$, fp(face$_i$), fs(face$_i$), tc(face$_i$), fl(face$_i$), τ_1^j) ∧ Meets(τ_1^j,τ_2^j)) ∧ *DES*(Track(face$_i$, trEBDI$_i$, τ_1^j)) → *INT*(Validate(face$_i$, trEBDI$_i$, τ_2^j));

– *BEL*(Validate(face$_i$, trEBDI$_i$, t_2^j) ∧ Face_data_t(face$_i$, fp(face$_i$), fs(face$_i$), tc(face$_i$), fl(face$_i$), τ_1^j) ∧ Meets(τ_1^j,τ_2^j) ∧ Finishes(t_2^j, τ_2^j) ∧ Meets(τ_2^j,τ_3^j)) ∧ *DES*(Update_t(face$_i$, trEBDI$_i$, t_2^j)) → *INT*(Update_t(face$_i$, trEBDI$_i$, τ_3^j));

– *BEL*(Validate(face$_i$, trEBDI$_i$, t_2^j) ∧ Face_data_t(face$_i$, fp(face$_i$), fs(face$_i$), tc(face$_i$), fl(face$_i$), τ_1^j) ∧ Meets(τ_1^j, τ_2^j) ∧ Finishes(t_2^j,τ_2^j) ∧ Meets(τ_2^j, τ_3^j) ∧ Meets(τ_3^j, τ_4^j)) ∧ *DES*(Inform_m(mEBDI, trEBDI$_i$, t_2^j)) → *INT*(Inform_m(mEBDI, trEBDI$_i$, τ_4^j)).

The intentions of the trEBDI agents during regular tracking and corresponding time intervals τ_k^j; $k = 1, \ldots, 4$, are shown in Fig. 5.

5.3. Exceptions – lost faces, new faces, re-detection

Activity of the trEBDI agent:

Case 1. An active trEDBI agent has lost its tracked face in the current frame

The first two steps of regular tracking described in Section 5.2 are performed.

1. The trEBDI agent changes its state from an active to suspended state and it informs the mEBDI agent that its tracked face is lost (i.e. the value of tracker confidence is below the threshold). The description of the above-described step at the logical level is as follows:
   – *BEL*(Track(face$_i$, trEBDI$_i$, τ_1^j) ∧ Face_data_t(face$_i$, fp(face$_i$), fs(face$_i$), tc(face$_i$), fl(face$_i$), τ_1^j) ∧ Meets(τ_1^j, τ_2^j)) ∧ *DES*(Track(face$_i$, trEBDI$_i$, τ_1^j)) → *INT*(Validate(face$_i$, trEBDI$_i$, τ_2^j));
   – *BEL*(¬Validate(face$_i$, trEBDI$_i$, t_2^j) ∧ Face_data_t(face$_i$, fp(face$_i$), fs(face$_i$), tc(face$_i$), fl(face$_i$), τ_1^j) ∧ Meets(τ_1^j, τ_2^j) ∧ Finishes(t_2^j, τ_2^j) ∧ Meets(τ_2^j, τ_3^j)) ∧ *DES*(Suspend(mEBDI, trEBDI$_i$, t_2^j)) → *INT*(Suspend(mEBDI, trEBDI$_i$, τ_3^j));
   – *BEL*(¬Validate(face$_i$, trEBDI$_i$, t_2^j) ∧ Lost(mEBDI, face$_i$, trEBDI$_i$, t_2^j) ∧ Face_data_t(face$_i$, fp(face$_i$), fs(face$_i$), tc(face$_i$), fl(face$_i$), τ_1^j) ∧ Meets(τ_1^j, τ_2^j) ∧ Finishes(t_2^j, τ_2^j) ∧ Meets(τ_2^j, τ_3^j) ∧ Meets(τ_3^j,τ_4^j)) ∧ *DES*(Redetect(mEBDI, face$_i$, trEBDI$_i$, t_2^j)) → *INT*(Inform_m(mEBDI, trEBDI$_i$, τ_4^j)).

   (this message invokes the first six steps performed by the mEBDI agent; see Activity of the mEBDI agent).
   The intentions of the trEBDI agents when they lose assigned faces and the corresponding time intervals τ_k^j; k = 1,..., 4, are shown in Fig. 6.
2. Suspended trEBDI agents wait for the mEBDI agent message – containing information about all unresolved faces detected in the current frame. Note that a message with no unresolved faces is also possible.
3. The trEBDI agent determines a degree of belief based on a tracker confidence value for all unresolved faces.
4. The trEBDI sends a message with a tracker confidence value for all unresolved faces to the mEDBI agent
   (this message invokes step 7 performed by the mEBDI agent; see Activity of the mEBDI agent).
5. Waiting for the mEBDI agent's message – containing information about the assignment of a face identity label.
6. Resolution of identity face label assignment. Based on the received message, two situations are possible:
   (i) the trEBDI agent continues to track its assigned face;
   (ii) the trEBDI agent remains in a suspended state (i.e. no unresolved face is assigned to it). The trEBDI agent waits for the mEBDI agent's message at the next frame (go to step 2).

Case 2. A trEDBI agent has lost its face in some previous frame

Typical situations of Case 2 are face tracking under short- and/or long-term face full occlusions (Soldić, Marčetić, Maračić, & Ribarić, 2017).

In Case 2, the procedure described in Case 1 is performed, excluding the change from the active to the suspended state of a trEBDI agent in step 1.

Activity of the mEBDI agent

The mEBDI performs the following steps for lost faces, new faces, and redetection.

Receiving messages from trEBDI agents that have lost a tracked face – each of the suspended trEDBI agents sends a message.

The mEBDI agent is informed that the trEBDI agent has lost a face. The description of the above-described step at the logical level is as follows:

– *BEL*(Lost(mEBDI, face$_i$, trEBDI$_i$, t_0^{j+1}) ∧ Face_data_t(face$_i$, fp(face$_i$), fs(face$_i$), tc(face$_i$), fl(face$_i$), τ_1^j) ∧ Meets(τ_1^j, τ_2^j) ∧ Meets(τ_2^j, τ_3^j) ∧ Meets(τ_3^j, τ_4^j) ∧ Meets(τ_4^j, τ_1^{j+1}) ∧ Starts(t_0^{j+1}, τ_1^{j+1})) ∧ *DES*(Redetect(mEBDI, face$_i$, trEBDI$_i$, t_0^{j+1})) → *INT*(Redetect(mEBDI, face$_i$, trEBDI$_i$, τ_1^{j+1}))

Table 2
Architecture of the CNN for face detection and localization.

Stage *No.*	Layer *No.*	*Type*	Kernels *Number*	*Size*	*Stride*	Number of parameters
1	1	con	16	$5\times5\times3$	2×2	$16\times5\times5\times3=1200$
	2	relu	–	–	–	0
2	3	con	32	$5\times5\times16$	2×2	$32\times5\times5\times16$=12,800
	4	relu	–	–	–	0
3	5	con	32	$5\times5\times32$	2×2	$32\times5\times5\times32$=25,600
	6	relu	–	–	–	0
4	7	con	45	$5\times5\times32$	1×1	$45\times5\times5\times32$=36,000
	8	relu	–	–	–	0
5	9	con	45	$5\times5\times45$	1×1	$45\times5\times5\times45$=50,625
	10	relu	–	–	–	0
6	11	con	45	$5\times5\times45$	1×1	$45\times5\times5\times45$=50,625
	12	relu	–	–	–	0
7	13	con	1	$9\times9\times45$	1×1	$19\times9\times45$=3645
	14	MMOD	–	–	–	0
Total number of parameters						180,495

1–3. These steps are the same as the first three steps in the initialization of the multi-face tracking system (see Section 5.1).
4. Unresolved faces determination – identity labelling is performed based on intersection over union (IoU) (or Jaccard similarity coefficient) between areas belonging to detected faces by the CNN detector and areas of faces tracked by active trEBDI agents in the previous frame. These areas are stored in a trajectory memory mTM of the mEBDI. Remaining face detections with an unassigned identity label are declared as unresolved.
5. Message sending – the mEBDI agent sends a message to all suspended trEDBI agents, containing information on all unresolved faces detected in the current frame
(this message invokes steps 3 and 4 performed by the trEBDI agent; see activity of the trEBDI agent).
6. Waiting for the trEBDI agent's messages – each of the suspended trEDBI agents sends a message with degrees of belief for all unresolved faces to the mEDBI agent.
7. Assignment of a face identity label – this step is performed based on all received PSR scores above the threshold, by using the Hungarian algorithm (Munkres, 1957).
8. Message sending – the mEDBI sends a message containing information about the assignment of a face identity label. Two types of messages are possible: one of the unresolved faces is or is not assigned to a trEBDI agent
(this message invokes step 5 performed by the trEBDI agent; see Activity of the trEBDI agent).
9. Initialization of new trEBDI agents – for all remaining unresolved faces (i.e., new faces), an initialization procedure described in step 4 of the initialization of the multi-face tracking system (see Section 5.1) is performed.

Note that these nine steps are also periodically performed (e.g., every 5 frames). This enables faces that newly appear in a frame to be detected and tracked.

For all the above steps, the descriptions at the logical level with formulas are developed, but due to space limitations only representative steps are given in this paper.

6. Extensions of the BDI agents – detection, elimination of false positive detections, face loss detection

A description of the implementation details related to the extension components of the mEBDI agent and a trEBDI agent is given in this section.

6.1. Description of the implementation details related to the extensions of mEBDI – the detection and elimination of false positive detections

The detection module, as an extension of the mEBDI agent, is based on a deep convolutional neural network (CNN). There are two main reasons for selecting the deep CNN for face detection and localization: (i) robustness (suitable for unconstrained conditions, low false negative and positive detections); and (ii) the CNN detector also returns the value of a confidence level, called detector confidence dc, in a range from 0 to 1, which expresses belief that the image patch at a specific position contains a face. The CNN detector architecture is designed for face detection and localization. The problem of face localization is solved based on the combination of a sliding window and max-margin object detection (MMOD) (King, 2015). The CNN consists of seven stages implemented as convolutional layers with a number of convolutional kernels all having the same size as the first two dimensions, i.e. (5×5), but it does not have pooling layers. Architecture details of the CNN for face detection and localization are given in Table 2. An input to the CNN detector is a frame with resolution $m\times n$ pixels. A resolution image pyramid consisting of six levels is created from the frame to handle the scale invariance problem. Deep CNN architecture is implemented by using the Dlib library (King, 2009).

A dataset of 6975 faces (Dlib C++ Library) obtained from ImageNet, AFLW, Pascal VOC, the VGG dataset, WIDER, and FaceScrub was used for learning the 180,495 parameters of the CNN.

Fig. 7 depicts the two examples of face detections.

The corresponding values of the terms for Face_data_d($face_i$, fp($face_i$), fs($face_i$), dc($face_i$), fl($face_i$), τ) predicates are:

T-ara video frame 3287:

Face_data_d($face_1$, fp($face_1$) = (211, 231), fs($face_1$) = 64, dc($face_1$) = 0.90, fl($face_1$) = 1, [3287/25, 3288/25]);
Face_data_d($face_2$, fp($face_2$) = (391, 200), fs($face_2$) = 78, dc($face_2$) = 0.95, fl($face_2$) = 2, [3287/25, 3288/25]); …,
Face_data_d($face_6$, fp($face_6$) = (906, 186), fs($face_6$) = 76, dc($face_6$) = 0.92, fl($face_6$) = 6, [3287/25, 3288/25]).

Girls Aloud video frame 2288:

Face_data_d($face_1$, fp($face_1$) = (153, 116), fs($face_1$) = 80, dc($face_1$) = 0.87, fl($face_1$) = 1, [2288/25, 2289/25]);
Face_data_d($face_2$, fp($face_2$) = (277, 106), fs($face_2$) = 80, dc($face_2$) = 0.83, fl($face_2$) = 2, [2288/25, 2289/25]); …,
Face_data_d($face_5$, fp($face_5$) = (730, 106), fs($face_5$) = 80, dc($face_5$) = 0.78, fl($face_5$) = 5, [2288/25, 2289/25]).

(a) (b)

Fig. 7. Examples of face detections; (a) T-ara; (b) Girls Aloud YouTube videos.

Note that a position (x, y) for $fp(face_i)$ corresponds to coordinates of the upper left corner of an image patch.

Elimination of false face detections is performed using a set of autonomy-oriented entities / agents (AOEs).

The AOEs are randomly distributed in the face image patches with detector confidence level values (obtained from the CNN detector) which are below the prescribed threshold to detect and eliminate image patches with false positive face detections (Kipčić & Ribarić, 2005). The agent $a_i \in$ AOEs, $i = 1, 2, \ldots, n$ is described by the 4-tuple:

$$a_i = (A, p, fid, a),$$

where:

- A is the age of an agent. The initial age of an agent is zero. An agent becomes older (its age increases by one) by diffusing or moving to another pixel in the image patch;
- p stands for the position of an agent in the image patch. The position is represented by pixel coordinates (x, y);
- fid is the family identifier, which indicates an agent's family membership. During initialization, every initial agent is a seed of a family, and it has a unique fid;
- a is the activity that indicates whether the agent is still searching for skin-like pixels.

Agents becomes inactive after breeding or by dying.

An agent shows the following types of behaviour: (i) diffusion – an agent diffuses (or moves)when it does not find a skin-like pixel in its current position. By diffusing, an agent changes its internal states: its position and age; (ii) Breeding – an agent breeds if it encounters a skin-like pixel at its current location. By breeding, it creates new agents in its neighbourhood that belong to its family. After breeding, the agent deactivates itself; and (iii) Dying – an agent dies when its age exceeds the maximum age determined for all agents.

AOE agents search image pixels that have skin-colour values defined based on combined HSI and RGB models, and form face-like regions. Based on a compound decision function composed of evaluation functions for the characteristics of face-like regions (Kipčić & Ribarić, 2005) that include size, fullness, orientation, and a regularity characteristic estimated on the height-width aspect-ratio, a final decision (the elimination of the image patch without a face) is made.

The AOE agent-based search procedure for skin pixels is formalized as follows:

The search procedure can be divided into four phases (Kipčić & Ribarić, 2005).

6.1.1. Initial agent distribution

The agents are randomly distributed over the face image patch (vague face region candidates).

6.1.2. Agent lifetime

Initialized agents in each vague face region start their lifecycle as follows:

```
repeat
  for each agent
      if agent is active
        if agent is beyond the region edge
            agent dies
        else
        if agent is on skin pixel
            mark pixel
            add agent to family statistic
            agent breeds
        else if agent_age > maximum age
            agent dies
        else agent diffuses
  next agent
while number of agents > 0
```

To determine a skin pixel, a hybrid HSI-RGB color model is used with the following parameters: $0 \le H \le 50$ and $340 \ge H \le 360$; $0.20 \le S \le 0.7$ and $R > 160$.

The initial number of agents and their maximum age is experimentally determined on a set of training regions with and without face.

6.1.3. Post-processing

When all the agents reach the end of their lifetimes, the result of their activities are areas that consist of skin-like pixels in the vague face region candidates. The areas are subjected to post-processing which consists of the elimination of areas smaller than the predefined threshold and/or grouping neighborhood areas with a "weak" border.

During the agents' lifetime, the following statistics were gathered for each vague face region candidate: number of skin-like pixels, the average value of the H, S and R, second moments, the centre of the area, and the rectangular frame containing the area with skin-like pixels.

6.1.4. Decision

The final decision about false positive detection is based on the logical function composed of the conjunction of five logical evaluation functions based on the above-mentioned statistics and the set of threshold values which are experimentally determined (Kipčić & Ribarić, 2005). If the value of the composed logical function is zero, this means that the vague face region candidate is a false positive detection.

Fig. 8 illustrates examples of false positive face CNN-based detections which are eliminated by the autonomy-oriented agents.

The trajectory memory mTM stores all trajectories of all tracked faces. The trajectory contains a face position, size, tracker confidence value, and time stamp for each face in all frames in the video.

Fig. 8. Examples of false positive face CNN-based detections.

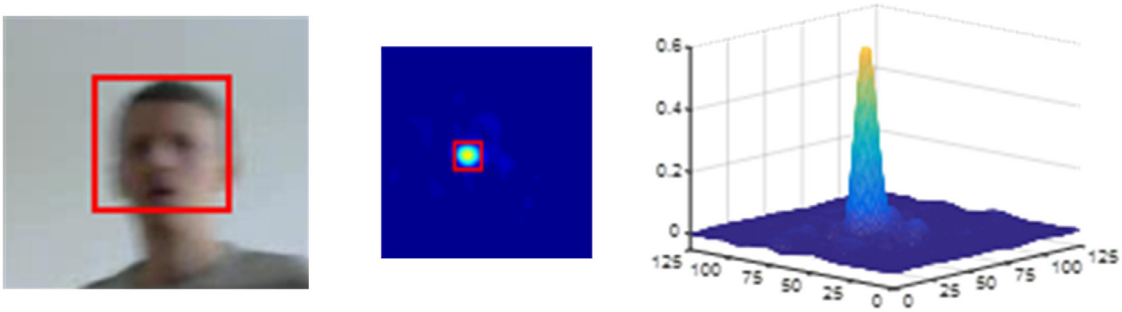

Fig. 9. Illustration of the PSR value = 52.43; (MaxPeak = 0.5887; Mean = 0.0014; Std = 0.0112) when a face is detected. The tracker confidence value tc is 0.655, the maximum value of the PSR is 80; (a) a face image patch (red square); (b) 2D representation of position correlation responses with a denoted peak with its surrounding area and sidelobe; (c) 3D representation of position correlation responses.

6.2. Description of the implementation details related to the extensions of trEBDI – tracking and face loss

The basic component of the tracking module, as an extension of the trEBDI agent, is the Discriminative Scale Space Tracking (DSST) module (Danelljan et al., 2017). There are three main reasons for selecting the DSST for face tracking: (i) suitability for online and real-time object tracking; (ii) robustness to scale and space variations of the visual appearance of the tracked object; and (iii) the ability to detect tracked object loss based on a value of a PSR obtained from the response of the position correlation filter.

A DSST tracker during the initialization procedure uses the position and size of a face image patch (received from the mEBDI) to learn the visual appearance and scale of the tracked faces (both stored in the visual appearance memory (VAM)). Two multichannel correlation filters of HOG features are calculated from a face image patch independently for position and scale. These filters are transformed into the Fourier domain to avoid a computationally intensive exhaustive search of a position-scale space. The circular correlation score is efficiently obtained in the Fourier domain which enables real time performance. Note that the scale correlation filter is obtained by using the sampling of the image patch at 33 different scales (from 0.7284 to 1.3728, with step 1.02 regarding the size of the initial image patch).

In the process of regular tracking, the DSST tracker (Danelljan et al., 2017) first searches for an optimal target position using the discriminative position correlation filter, and then applies the scale correlation filters to find an optimal target scale. The position and scale correlation filters are updated online for each consecutive frame.

The steps performed by the DSST tracker are described in more detail as follows (Danelljan et al., 2017): for each consecutive frame, an image patch centred on the face position estimated in the previous frame is selected as a candidate for a face position search. For the image patch, a 28-channel position correlation filter is obtained, as described for the initialization. Then a new target position is estimated by performing a circular correlation between the 28-channel position correlation filter from the previous frame and the features obtained from the image patch in the current frame. The new position of the target in the current frame is determined based on the maximal correlation response. Patches at 33 scales centred on the target position estimated in the previous step are taken from the current frame. Then, a new target scale is estimated by performing a circular correlation between the scale correlation filter and the features obtained as described above. The new scale of the target in the current frame is determined based on the maximal correlation response which corresponds to one of 33 scales.

Position and scale correlation filters are updated online for position and scale target estimation in the next frame. Both filters are updated separately based on the defined learning rate.

A tracker confidence value, which is used for face loss detection, is determined by using a normalized PSR.

The peak strength called the PSR is obtained from the response of the position correlation filter (Bolme et al., 2010). The position correlation response is divided into two areas (Figs. 9 and 10): the peak with its surrounding area and the area called the sidelobe. The square area surrounding the peak has a square side that is 12% of the size of the position correlation response (which is represented as a 2D square matrix). The sidelobe is the remaining area that excludes the area surrounding the peak. The value of the PSR is computed by the following formula:

$$\mathrm{PSR} = \frac{(MaxPeak - \mu)}{\sigma},$$

where *MaxPeak* is the maximum peak value, and the mean value μ and standard deviation σ are calculated from the sidelobe of the position correlation response. The value of the PSR for

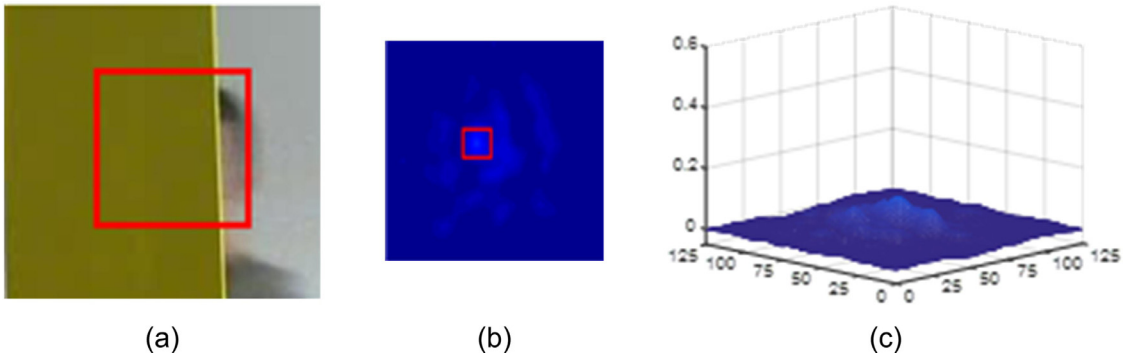

Fig. 10. Illustration of the PSR value = 8, 56 (*MaxPeak* = 0.142; *Mean* = 0.005; *Std* = 0.016) when a face is lost. The tracker confidence value tc = PSR / maxPSR = 0.107: tc < threshold = 0.125 – a face is lost; (a) a face image patch (red square); (b) 2D representation of position correlation responses with denoted peak with its surrounding area and sidelobe; (c) 3D representation of position correlation responses.

Fig. 11. An illustration of face tracking: (a) Apink; (b) Westlife YouTube videos.

so-called good tracking (i.e. manual annotation) is typically between 20 and 60. In the case of tracking failure, the PSR is less than 10 (Bolme et al., 2010). By dividing the PSR value obtained for the current face image patch with its maximum value determined by using a ground truth face annotation for the set of experimental video sequences, the tracker confidence value is obtained. To obtain a tracker confidence value tc, the actual value of the PSR is divided by the maximal value of the PSR which is experimentally determined.

Figs. 9 and 10 illustrate the PSR values for different tracked confidence values.

Fig. 11 illustrates the result of face tracking obtained by corresponding trEBDI agents.

The corresponding values of terms for Face_data_t($face_i$, fp($face_i$), fs($face_i$), tc($face_i$), fl($face_i$), τ) predicates are:

Apink video frame 968:

Face_data_t($face_1$, fp($face_1$) = (73, 160), fs($face_1$) = 131,
tc($face_1$) = 0.28, fl($face_1$) = 1, [968/25, 969/25]);
Face_data_t($face_2$, fp($face_2$) = (268, 201), fs($face_2$) = 115,
tc($face_2$) = 0.64, fl($face_2$) = 2, [968/25, 969/25]); ...;
Face_data_t($face_6$, fp($face_6$) = (1096, 206), fs($face_6$) = 122,
tc($face_6$) = 0.79, fl($face_6$) = 6, [968/25, 969/25]).

Westlife video frame 2391:

Face_data_t($face_1$, fp($face_1$) = (183, 248), fs($face_1$) = 71,
tc($face_1$) = 0.47, fl($face_1$) = 1, [2391/25, 2392/25]);
Face_data_t($face_2$, fp($face_2$) = (502, 209), fs($face_2$) = 70,
tc($face_2$) = 0.46, fl($face_2$) = 2, [2391/25, 2392/25]); ...;
Face_data_t($face_4$, fp($face_4$) = (825, 248), fs($face_4$) = 122,
tc($face_4$) = 0.29, fl($face_4$) = 4, [2391/25, 2392/25])

7. Experimental setup and preliminary results

The demonstration of the proposed approach based on a multi-agent dynamic system for robust multi-face tracking was performed on a subset of YouTube music videos (Zhang, Gong, et al., 2016). The test subset of the YouTube music video sequences consists of five videos (T-ara, Hello Bubble, Apink, Westlife and Girls Aloud) that are challenging to track due to large visual differences caused by face appearance variations (changes in pose, size, make-up, and illumination), and/or rapid camera motion. The remaining three videos (Pussycat Dolls, Bruno Mars, and Darling) are used to determine and adjust the set of parameters for proper system operation. These three videos are excluded from the evaluation of the system.

The parameters have the following values: the tracker confidence threshold is 0.125; the PSR score threshold is 10; the intersection over union (IoU) for identity labelling is 0.5; new faces are periodically detected every 5 frames; the face detection confidence threshold is 0.72; the maximum age for all AOEs is 8; the scale pyramid with 33 levels is spaced at 1.02 up to 1.37 down to 0.73 relative to the original image patch size.

In the experiments for this test subset, we used manual shot change labelling to divide the input videos into non-overlapping shots instead of the method used in (Zhang, Gong, et al., 2016). Note that for every shot a tracking process is restarted.

The qualitative results of the proposed multi-face tracking system are illustrated in Fig. 12.

Fig. 13 illustrates examples of the ground truth of visual appearances and the corresponding visual appearances of tracked faces in an Apink video sequence with perfect tracking results. Each column corresponds to one frame. Each row consists of two sub-rows: the upper sub-row depicts the ground truth and the lower sub-row depicts the corresponding tracked face. Fig. 14 illustrates examples

Fig. 12. The qualitative results of the proposed multi-face tracking system in 1–5 rows are T-ara, Hello Bubble, Apink, Westlife, Girls Aloud, respectively.

Fig. 13. Examples of ground-truth visual appearances and the corresponding visual appearances of tracked faces for an Apink video.

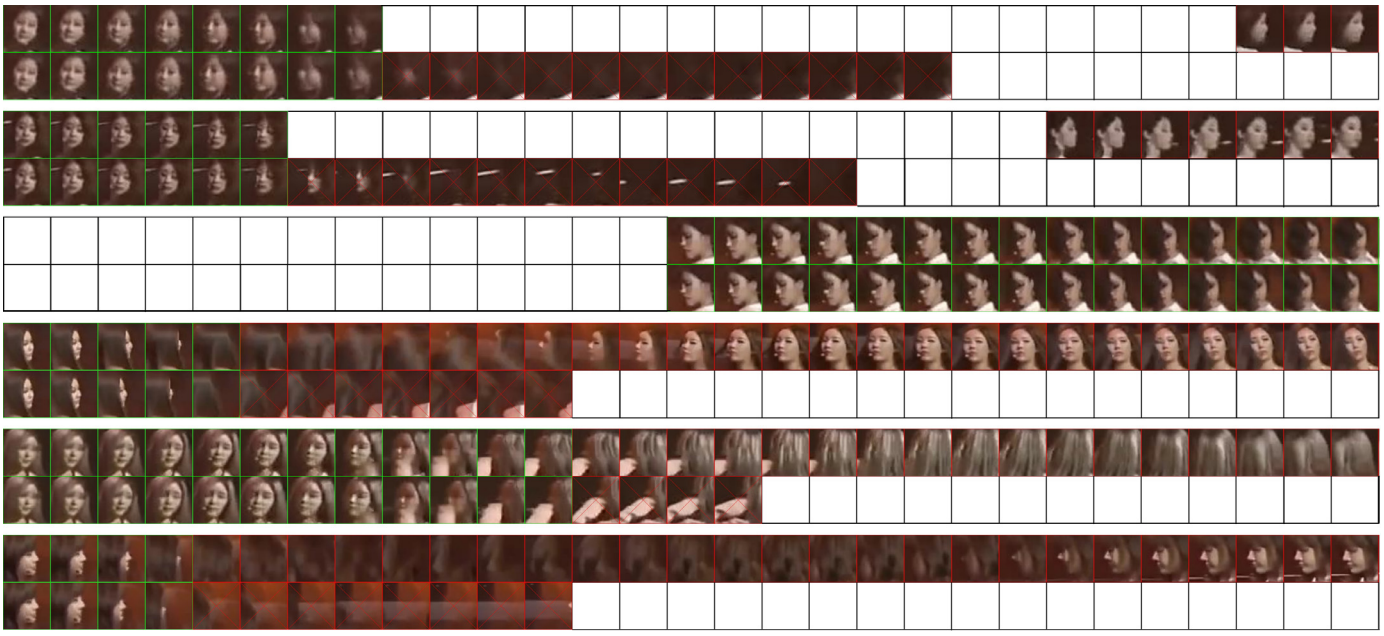

Fig. 14. Examples of ground-truth visual appearances and the corresponding visual appearances of tracked faces for a T-ara video.

Table 3
Quantitative results of the proposed architecture for multi-face tracking.

Video\Metric:	FP	FN	GT	IDS	MOTP	MOTA
T-ara (4547 frames)	1262	2014	14,321	18	0.77	0.77
Hello Bubble (3764 frames)	447	483	5169	5	0.76	0.82
Apink (5275 frames)	405	474	7177	6	0.77	0.88
Westlife (5736 frames)	1608	1841	11,289	4	0.68	0.69
Girls Aloud (5531 frames)	2427	2711	16,048	23	0.75	0.68

of the ground truth of visual appearances and the corresponding visual appearances of tracked faces in a T-ara video sequence with false positive and false negative tracking results. A white box in the first sub-row of a row denotes that a face is not present in the frame, while a white box in the second sub-row means that a tracker is not tracking a face in a frame. Note that poor ground-truth annotations for faces in a few frames have resulted in tracking failures for the last three faces.

The preliminary quantitative results are expressed by the testing metrics (Kasturi et al., 2009): MOTA (*Multiple Object Tracking Accuracy*), MOTP (*Multiple Object Tracking Precision*), and IDS (*Identity Switch*). The MOTA measure combines false negatives (FN), false positives (FP), mismatches or identity switches (IDS), and ground truth (GT) obtained from each frame from $j=0$ to N-1, where N is the total number of frames in a video sequence:

$$\mathrm{MOTA} = 1 - \frac{\sum_{j=0}^{N-1} (\mathrm{FN}_j + \mathrm{FP}_j + \mathrm{IDS}_j)}{\sum_{j=0}^{N-1} \mathrm{GT}_j}$$

The MOTP is the average dissimilarity between all true positives and their corresponding ground truth targets (faces). For bounding box overlap, MOTP is computed as

$$\mathrm{MOTP} = \frac{\sum_{n=1}^{M} \sum_{j=0}^{N-1} d_{n,j}}{\sum_{j=0}^{N-1} c_j},$$

where M denotes the number of different objects (faces) in the entire video sequence, N is the number of frames in a video sequence, $d_{n,j}$ is the bounding box overlap of tracked face n with its assigned ground truth face in the frame j, and c_j denotes the number of visible objects (faces) in the frame j.

IDS (*Identity switches*) counts how many switches in object labelling occurred during the tracking when compared to the ground truth.

The quantitative results of the experiments are given in Table 3.

The obtained experimental results (Table 3) show that the performance of the proposed system is comparable with state-of-the-art performance Zhang, Gong, et al. (2016) and Lin and Hung (2018). The experimental outcomes show that the proposed MADS system architecture is capable of multi-face tracking in unconstrained videos.

Note that the quantitative results in Table 3 are mainly dependent on the characteristics of the components of the mEBDI and trEBDI agents: a CNN-based face detector, the efficiency of the AOE agents and the DSST tracker, but also dependent on the testing procedure.

Better results can be achieved by using more advanced methods, for example a deep residual CNN-based approach (He, Zhang, Ren, & Sun, 2016), Siamese network and/or triplet network-based approaches (Zhang, Gong, et al., 2016), an approach based on a so-called prior-less framework and co-occurrence model that can continue tracking partially visible multiple faces (Lin & Hung, 2018), and using a CNN-based face detector and the ResNet face recognition network to minimize the number of identity switches (Marčetić & Ribarić, 2018).

The main aim of the experiments was to demonstrate the effective adaptation of MADS to the architecture of a multi-face tracking system.

8. Adaptation of MADS to other classes of problems

In this section, two short examples are given as an illustration of the adaptation of the generic MADS architecture to different classes of problems (stock trading and automatic image annotation).

An adaptation of MADS architecture to a decision-making system for a stock market is described as follows. Many researchers have concluded that the dynamics of networked market systems are better understood as complex adaptive systems, in which independent software components interact without centralized control or oversight (Paulin, Calinescu, & Wooldridge, 2018). A process of buying and selling financial assets is guided by complex and efficient algorithms. The MADS architecture should consist of two types of EBDI agents: (i) investor agents and (ii) broker agents. These agents act in an environment which is a real-time trading platform. Inside the environment, agents should have intentions toward objects, i.e. financial assets. A reasoning process is based on relations between both types of agents and objects and should have a temporal constraint. The investor agents are assigned to real investors. These agents model investor's beliefs and desires with data such as financial targets, a defined risk level and investment horizons. The investor agents are extended with a stock market analysis function which enables the creation of the agents' intention e.g., to buy or sell stocks. Using these types of agent, some typical investors' mistakes which can happen due to irrational decisions in crisis situations (e.g., panic selling) or during a reasoning process under the influence of emotions (mostly fear and greed) are avoided. An intention of broker agents is to execute intentions defined by investing agents. The broker agents should have real time access to trade platforms to perform their intentions.

In another example, in order to design a MADS-based system for automatic image annotation as a two-tier annotation model (Ivasic-Kos, Pobar, & Ribaric, 2016), a hierarchical structure of extended BDI (EBDI) agents should be introduced: at the first level there are EBDI agents which are specialized in image annotation at the object level, and EBDI agents at the second level which use inference-based algorithms to handle the recognition of scenes and higher-level concepts. Both sets of EDBI agents should be supported by knowledge representation schemes with specific knowledge about low-level image features and higher-level concepts, respectively.

9. Conclusion

The paper has presented a novel multi-agent dynamic system called MADS. The MADS model can be easily adapted for a specific problem domain. In this paper, MADS is adapted to robust multi-face tracking in video sequences. The proposed MADS architecture, based on an extended BDI-based agent paradigm, is represented as a two-level hierarchical organization. At the first level, there is a manager agent designed as an Extended Belief Desire Intention (mEBDI) agent. The mEBDI agent is a hybrid agent consisting of a BDI agent extended by a set of autonomy-oriented entities (behaviour agents), by a trajectory memory, and by a deep convolutional neural network-based face detection module. At the second level, there are a number of tracking agents (trEBDIs) which consist of a basic BDI agent extended with an integration module. The integration module contains a tracker based on position and scale correlation filters, a visual appearance memory, and a trajectory memory. The mEBDI agent's main tasks are initialization and face detection in a video sequence and the management of a number of tracking agents (trEBDI) at the second level. The trEBDI agents track faces detected by the mEBDI agent. Every trEBDI agent is responsible for tracking one face initially assigned to it by the mEBDI agent. During the multi-face tracking process, the proposed multi-agent system can operate in the following states: initialization, regular tracking, and exception. The mEBDI agent's and the trEBDI agent's activities in all three states have been described in detail. The mEBDI agent and trEBDI agents interact through cooperation and communication, i.e. they exchange messages with information about faces that are necessary to maintain the dynamics in MADS. Note that the trEBDI agents are wholly autonomous during the state of regular tracking. The activities of the agents and their interaction are explained with suitable rules and are written in formulas in modal logic.

The proposed robust multi-face tracking system based on MADS architecture was tested on a subset of YouTube music videos. The qualitative results of the proposed multi-face tracking system, as well as the preliminary quantitative results expressed by the testing metrics MOTA, MOTP and IDS, demonstrate the effective adaptation of MADS to the architecture of a multi-face tracking system.

Future research work will be directed to include:

(i) The integration of deep learning concepts, conventional approaches and knowledge from the problem domain (e.g., intelligent multisensory, distributed surveillance, crowd analysis) based on an extension of BDI agents in MADS by management abilities supported by a knowledge representation scheme and common-sense reasoning;
(ii) The development of an efficient interface between agents with mental attitudes and autonomy-oriented entities (AOEs) in MADS which are oriented to the solution of highly parallel local oriented tasks;
(iii) The adaptation of MADS to classes of problems such as automatic image annotation, internet search, stock trading, modelling real-life dynamic situations, etc.

In the near future our research efforts will be directed towards:

(iv) The integration of a robust multi-face tracking system into a face de-identification pipeline and to the adaptation of the MADS model to a third de-identification pipeline stage (i.e., face masking by applying different privacy filters);
(v) The adaptation of MADS for crowd analysis at a hybrid level (e.g., the combination of a microscopic and macroscopic approach).

Credit authorship contribution statement

Lada Malеš: Conceptualization, Formal analysis, Investigation, Methodology, Writing - original draft. **Darijan Marčetić:** Investigation, Software, Validation, Writing - original draft. **Slobodan Ribarić:** Conceptualization, Supervision.

Acknowledgments

This work has been supported by the Croatian Science Foundation under project 6733 De-identification for Privacy Protection in Surveillance Systems (DePPSS) and project 7619 A Knowledge-based Approach to Crowd Analysis in Video Surveillance (KACAVIS).

References

Airiau, S., Padgham, L., Sadrina, S., & Sen, S. (2008). Incorporating learning in BDI agents. In *Workshop AAMAS: adaptive and learning agents and MAS (ALAMAS+ALAg)* (pp. 49–56). Estoril: ACM.

Allen, J. F. (1983). Maintaining knowledge about temporal intervals. *Communications of the ACM, 26*(11), 832–843.

APIDIS basket ball dataset. (n.d.). Retrieved 11 21, 2018, from ISP Group: https://sites.uclouvain.be/ispgroup/Softwares/APIDIS.

Blackburn, P., & van Benthem, J. (2007). Modal Logic: A semantic perspective. In P. Blackburn, J. van Benthem, & F. Wolter (Eds.), *Handbook of modal logic* (pp. 1–84). Amsterdam: Elsevier.

Bolme, D. S., Beveridge, R. J., Draper, B. A., & Lui, Y. M. (2010). Visual object tracking using adaptive correlation filters. In *IEEE conference on computer vision and pattern recognition (CVPR)* (pp. 2544–2550). San Francisco: IEEE. doi:10.1109/CVPR.2010.5539960.

Bratman, M. E. (1987). *Intentions, plans and practical reason*. Cambridge: Harvard University Press.

Castanedo, F., García, J., Patricio, G., Miguel, Á., & Molina, J. M. (2010). A multi-agent architecture based on the BDI model for data fusion in visual sensor networks. *Journal of Intelligent and Robotic Systems, 62*(3), 299–328. doi:10.1007/s10846-010-9448-1.

Danelljan, M., Häger, G., Khan, F. S., & Felsberg, M. (2017). Discriminative scale space tracking. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 38*(9), 1561–1575. doi:10.1109/TPAMI.2016.2609928.

Dix, J., & Fisher, M. (2011). Where logic and agents meet. *Journal of Annals of Mathematics and Artificial Intelligence, 61*(1), 15–28.

Dlib C++ Library. (n.d.). Retrieved March 21, 2017, from Dlib C++ Library: http://dlib.net/files/data/dlib_face_detection_dataset-2016-09-30.tar.gz.

Dorigo, M., & Stützle, T. (2010). Ant colony optimization: Overview and recent advances. In M. Gendreau, & J.-Y. Potvin (Eds.), *Handbook of metaheuristics* (pp. 227–263). New York: Springer.

Felzenszwalb, P. F., Ross, G. B., McAllester, D., & Ramanan, D. (2010). Object detection with discriminatively trained part-based models. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 32*(9), 1627–1645. doi:10.1109/TPAMI.2009.167.

Ferber, J. (1999). *Multi-agent systems, an introduction to distributed artificial intelligence*. New York: Addison Wesley LongmanInc.

Fisher, M., Bordini, R. H., Hirsch, B., & Torroni, P. (2007). Computational logics and agents: A road map of current technologies and future trends. *Computational Intelligence, 10*(1), 41–66.

Garnier, S., Gautrais, J., & Theraulaz, G. (2007). The biological principles of swarm intelligence. *Swarm Intelligence, 1*(1), 3–31.

Gascuena, J. M., & Fernandez-Cabalero, A. (2011). On the use of agent technology in intelligent, multisensory and distributed surveillance. *The Knowledge Engineering Review, 26*(2), 191–208. doi:10.1017/S0269888911000026.

Graf, T., & Knoll, A. (2000). A multi-agent system architecture for distributed computer vision. *International Journal on Artificial Intelligence Tools, 9*(2), 305–319.

He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 770–778). Las Vegas: IEEE.

van der Hoek, W., & Wooldridge, M. (2012). Logics for multiagent systems. *AI Magazine, 33*(3), 92–105.

Ivasic-Kos, M., Pobar, M., & Ribaric, S. (2016). Two-tier image annotation model based on a multi-label classifier and fuzzy-knowledge representation scheme. *Pattern Recognition, 52*, 287–305. doi:10.1016/j.patcog.2015.10.017.

Jain, A. K., & Li, S. Z. (2011). *Handbook of face recognition*. New York: Springer.

Kasturi, R., Goldgof, D., Soundararajan, P., Manohar, V., Garofolo, J., & Bowers, R. (2009). Framework for performance evaluation of face, text, and vehicle detection and tracking in video: Data, metrics, and protocol. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 31*(2), 319–336. doi:10.1109/TPAMI.2008.57.

Kennedy, J., Eberhart, R. C., & Shi, Y. (2001). *Swarm intelligence*. San Francisco, CA: Morgan Kaufmann Publishers.

King, D. E. (2009). Dlib-ml: A machine learning toolkit. *Journal of Machine Learning Research, 10*, 1755–1758. doi:10.1145/1577069.1755843.

King, D. E. *Max-margin object detection* Retrieved April 5, 2017, from arXiv preprint arXiv:1502.00046 https://arxiv.org/pdf/1502.00046.pdf.

Kipčić, D., & Ribarić, S. (2005). A multi-agent-based approach to face detection and localization. In *International conference on information technology interfaces ITI* (pp. 377–382).

Liao, S., Jain, A. K., & Li, S. Z. (2016). A fast and accurate unconstrained face detector. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 38*(2), 211–223.

Lin, C.-C., & Hung, Y. (2018). A prior-less method for multi-face tracking in unconstrained videos. In *Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR)* (pp. 538–547). Salt Lake City: IEEE.

Liu, J. (2008). Autonomy-Oriented Computing (AOC): The nature and implications of a paradigm for self-organized computing. In *ICNC'08. Fourth international conference on natural computation* (pp. 3–11). JinanChina: IEEE.

Liu, J., & Tsui, K. C. (2006). Toward nature-inspired computing. *Communications of the ACM, 49*(10), 59–64.

Liu, J., Jin, X., & Tsui, K. (2004). *Autonomy oriented computing: From problem solving to complex systems modeling*. New York: Kluwer Academic Publishers - Springer.

Maleš, L., & Ribarić, S. (2016). A model of extended BDI agent with autonomous entities. In *8th International conference on intelligent systems* (pp. 205–214). Sofia: IEEE. doi:10.1109/IS.2016.7737422.

Marčetić, D., & Ribarić, S. (2018). An online multi-face tracker for unconstrained videos. In *The 14th international conference on signal image technology & internet based systems* (pp. 1–7). Las Palmas de Gran Canaria: IEEE.

Meng, Y. (2009). Agent-based reconfigurable architecture for real-time object tracking. *Journal of Real-Time Image Processing, 4*(4), 339–351. doi:10.1007/s11554-009-0116-2.

Munkres, J. (1957). Algorithms for the assignment and transportation problems. *Journal of the Society for Industrial and Applied Mathematics, 5*(1), 32–38. doi:10.1137/0105003.

Muñoz-Salinas, R., Aguirre, E., García-Silvente, M., Ayesh, A., & Góngora, M. (2009). Multi-agent system for people detection and tracking using stereo vision in mobile robots. *Robotica, 27*(5), 715–727. doi:10.1017/S0263574708005092.

Paulin, J., Calinescu, A., & Wooldridge, M. (2018). Agent-based modeling for complex financial systems. *IEEE Intelligent Systems, 33*(2), 74–82. doi:10.1109/MIS.2018.022441352.

Qin, Z., & Shelton, C. R. (2016). Social grouping for multi-target tracking and head pose estimation in video. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 38*(10), 2082–2095. doi:10.1109/TPAMI.2015.2505292.

Rao, A. S., & Georgeff, M. (1991). Modeling rational agents within a BDI-architecture. In *Proc. of knowledge representation and reasoning* (pp. 473–484). San Mateo: Morgan Kaufmann.

Rao, A. S., & Georgeff, M. P. (1995). BDI agents: From theory to practice. In *Proceedings of 1st international conference on multi-agent systems* (pp. 312–319).

Ribarić, S., & Pavešić, N. (2015). An overview of face de-identification in still images and videos. In *11th IEEE international conference and workshops on automatic face and gesture recognition (FG)* (pp. 1–6). Ljubljana: IEEE.

Ribarić, S., & Pavešić, N. (2017). De-identification for privacy protection in biometrics. In C. Vielhhauer (Ed.), *User-centric privacy and security in biometrics* (pp. 293–315). London: IET. doi:10.1049/PBSE004E.

Ribarić, S., Ariyaeeinia, A., & Pavešić, N. (2016). De-identification for privacy protection in multimedia content: A survey. *Signal Processing: Image Communication, 47*, 131–151. doi:10.1016/j.image.2016.05.020.

Shoham, Y., & Leyton-Brown, K. (2008). *Multiagent systems: Algorithmic, game-theoretic, and logical foundations*. New York: Cambridge University Press.

Simonyan, K., & Zisserman, A. (2015). Very deep convolutional networks for large-scale image recognition. In *Proc. of international conference on learning representations (ICLR),* (pp. 1–14).

Singh, D., Padgham, L., & Logan, B. (2016). Integrating BDI agents with agent-based simulation platforms. *Autonomous Agents and Multi-Agent Systems, 30*(6), 1050–1071. doi:10.1007/s10458-016-9332-x.

Singh, D., Sardina, S., Padgham, L., & Airiau, S. (2010). Learning context conditions for BDI plan selection. In *AAMAS, proceedings of the 9th international conference on autonomous agents and multiagent systems: 1* (pp. 325–332).

Soldić, M., Marčetić, D., Maračić, M., & Ribarić, S. (2017). Real-time face tracking under long-term full occlusions. In *10th International symposium on image and signal processing and analysis (ISPA 2017)* (pp. 147–152). doi:10.1109/ISPA.2017.8073586.

Stone, P., & Veloso, M. (2000). Multiagent systems: A survey from a machine learning perspective. *Autonomous Robots, 8*(3), 345–383.

Takemura, N., Nakamura, Y., Matsumoto, Y., & Ishiguro, H. (2012). A path-planning method for human-tracking agents based on long-term prediction. *IEEE Transaction on Systems, Man, and Cybernetics - Part C: Applications and Reviews, 42*(6), 1543–1554. doi:10.1109/TSMCC.2012.2203801.

Van Dyke Parunak, H., Nielsen, P., & Brueckner, S. (2006). Hybrid multi-agent systems: Integrating swarming and BDI agents. In *4th international workshop engineering self-organising systems, ESOA 2006* (pp. 1–14). Heidelberg: Springer, Berlin.

Wang, Y., Qi, Y., & Li, Y. (2013). Memory-based multiagent coevolution modeling for robust moving object tracking. *The Scientific World Journal, 2013*, 793013 1–13. doi:10.1155/2013/793013.

Weiss, G. (1999). *Multiagent systems: A modern approach to distributed artificial intelligence*. Cambridge: MIT press.

Wen, L., Lei, Z., Lyu, S., Li, S. Z., & Yang, M. H. (2016). Exploiting hierarchical dense structures on hypergraphs for multi-object tracking. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 38*(10), 1983–1996.

Yang, B., Liu, J., & Liu, D. (2010). An autonomy-oriented computing approach to community mining in distributed and dynamic networks. *Autonomous Agents and Multi-Agent Systems, 20*(2), 123–157. doi:10.1007/s10458-009-9080-2.

Zaghetto, C., Aguiar, L. H., Zaghetto, A., Ralha, C. G., & de Barros Vidal, F. (2017). Agent-based framework to individual tracking in unconstrained environments. *Expert System With Applications, 87*, 118–128. doi:10.1016/j.eswa.2017.05.065.

Zhang, S., Gong, Y., Huang, J.-B., Lim, J., Wang, J., & Ahuja, N. (2016a). Tracking persons-of-interest via adaptive discriminative features. In *European conference on computer vision* (pp. 415–433). Cham: Springer.

Zhang, X., Zou, J., He, K., & Sun, J. (2016b). Accelerating very deep convolutional networks for classification and detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 38*(10), 1943–1955.