Modelling the People Recognition Pipeline in Access Control Systems

[1] *F. Gossen <frederik.gossen@lero.ie>*
[1] *T. Margaria <tiziana.margaria@lero.ie>*
[2] *T. Göke <thomas.goeke@systeam-gmbh.com>*
[1] *Lero - The Irish Software Research Centre, University of Limerick, Tierney Building, University of Limerick, V94 NYD3, Ireland.*
[2] *SysTeam GmbH, Technologiepark, Martin-Schmeißer-Weg 14, 44227 Dortmund, Germany.*

Abstract. We present three generations of prototypes for a contactless admission control system that recognizes people from visual features while they walk towards the sensor. The system is meant to require as little interaction as possible to improve the aspect of comfort for its users. Especially for people with impairments, such a system can make a major difference. For data acquisition, we use the Microsoft Kinect 2, a low-cost depth sensor, and its SDK. We extract comprehensible geometric features and apply aggregation methods over a sequence of consecutive frames to obtain a compact and characteristic representation for each individual approaching the sensor. All three prototypes implement a data processing pipeline that transforms the acquired sensor data into a compact and characteristic representation through a sequence of small data transformations. Every single transformation takes one or more of the previously computed representations as input and computes a new representation from them. In the example models presented in this paper, we are focusing on the generation of frontal view images of peoples' faces, which is part of the processing pipeline of our newest prototype. These frontal view images can be obtained from colour, infrared and depth data by rendering the scene from a changed viewport. This pipeline can be modelled considering the data flow between data transformations only. We show how the prototypes can be modelled using modelling frameworks and tools such as Cinco or the Cinco-Product Dime. The tools allow for modelling the data flow of the data processing pipeline in an intuitive way.

Keywords: Visual Modelling, Face Recognition, People Recognition, Computer Vision.

DOI: 10.15514/ISPRAS-2016-28(2)-14

For citation: Gossen F., Margaria T., Göke T. Modelling the People Recognition Pipeline in Access Control Systems. *Trudy ISP RAN/Proc. ISP RAS*, vol. 28, issue 2, 2016, pp. 205-220. DOI: 10.15514/ISPRAS-2016-28(2)-14

1 Introduction

When using today's admission control systems some kind of interaction is required to check permission for every individual. Among the most widely used technologies are RFID chips on check cards. When attempting to pass the admission control system people have to swipe their check card through a reader to transfer a unique ID to the system. The system will then check whether or not access should be granted. Once a person is identified, it is easy to assign different levels of permission to different people. This might be useful to restrict access to certain areas in a building. Other methods for identification include PINs, passwords or keys. All these methods have one thing in common: They require the user to carry something, either physically or in mind. That means it is likely that someone who is allowed to pass the admission control system is not able to do so because he or she has forgotten his or her password or key. To overcome this issue iris recognition, fingerprint recognition and face recognition can be used [1]. All of these methods identify a person by something that cannot be forgotten such as the eye or the face.

In our work, we focus on identity recognition from colour and depth data using low-cost depth sensors such as the Microsoft Kinect 2 [2]. These sensors offer colour, infrared and depth images at high frame rates. Our goal is to recognize people with as little interaction as possible. The user should be able to walk towards the admission control system looking forwards as he or she walks. The system picks up his or her head and face and predicts the identity that is most likely to have caused the observation. Moreover, the system will have notion of certainty. In cases where the prediction is possibly wrong a fall back method for identification will be used. This can be a PIN, password or a check card but it is also possible to redirect the person to a staff member to be identified with human capabilities.

The proposed system primarily improves the aspect of comfort for everybody who uses the admission control system as they no longer have to carry check cards or keys or remember PINs or passwords. Such an admission control system can be used in many places. Starting from fitness studios, spa and swimming pools where members have to be recognized to give access, reaching to institutions where staff members have to be recognized. In these scenarios, the proposed system primarily improves the aspect of comfort. However, in some cases people are not able to use any of the alternative methods for identification. Especially in places like hospitals and retirement homes where many people suffer from impairments such a system can make a major difference. People with Parkinson's disease might be unable to swipe a check card with the tremor in way that allows the system to read the card. They will also have problems to enter a PIN or a password while a visual recognition system would not require them to interact in a particular way. Other examples include patients with Alzheimer's disease, people wearing a cast or doctors with sterilized hands.

We are currently working on the third version of a prototype for contactless admission control. Previous versions have suggested that geometric features can contribute to reliable recognition of individuals but are alone not sufficient for reliable access

control [3]. We are currently focusing on the generation of frontal view images from people's faces, which we will use to extract comprehensible and characteristic features for individuals. This will allow for recognizing them in the application of an admission control system. In what follows we will present the three versions of our prototype in Section 2. All three prototypes implement a data processing pipeline that transforms the acquired sensor data into a compact and characteristic feature representation. In order to show how this pipeline can be modelled, Section 3 introduces the reader to the meta modelling framework Cinco and to Cinco-Products that we use to model the pipeline. We present two alternative models of the data processing pipeline in Section 4 that are based on these Cinco-Products. Section 5 concludes this paper and points our directions of future work.

2 Prototypes

We are currently working on the third version of our prototype. The first two versions were based on the Microsoft Kinect [2] and its SDK while our new version will be based only on its low-level API that is similar to that of comparable sensors. The low-cost sensor provides capabilities to acquire colour, depth and infrared images at high frame rates. It comes with a powerful SDK that provides reliable algorithms to detect people's skeletons and faces.

In this section, we will describe the three versions of our prototype and we analyse their differences.

2.1 First Prototype

Starting with the first prototype, we decided to use the Microsoft Kinect 2 sensor. This sensor acquires colour, infrared and depth images at high frame rates. Moreover, the sensor comes with a Software Development Kit (SDK) that offers a high resolution face model and a skeleton model. The first system is based on the capabilities of the Kinect SDK. We use the high resolution face model to extract characteristic geometric features with clear interpretation. The features are extracted on frame by frame basis. The compact and comprehensible set of features is used to predict the person who is most likely to have caused the observation. In this first approach we use our own implementation of a Bayesian Classifier to perform this task on every frame separately.

As we aim to recognize human identities in a comprehensible way we need features that provide clear interpretation. As one of the first features that were used for facial recognition, geometric features fulfil this requirement [4] [5]. In contrast to early approaches we extract distances in space rather than in the image plane using the Kinect's face and its skeleton model.

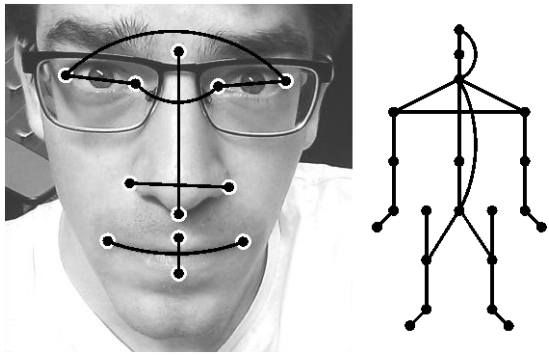

Fig. 1. Visualization of 8 distances that were extracted from the Kinect's face model (left) and 23 distances that were extracted from the Kinect's skeleton model (right). The distances are estimated in space rather than in the image plane to be invariant under the viewport.

The Kinect's face model provides a set of 1347 feature points many of which are interpolated. In order to obtain features with clear meaning we decided to focus on a subset of 12 feature points as shown in Figure 1. These feature points represent the eye corners, the moth corners, the lower and the upper lip and the left, right, lower and upper boundary of the nose. The face model provides these points' positions in both, the image plane and in space. As distances in the image plane are affected by the view point we use the Euclidean distances between points in space. We extract the following distances from the face model

- the inner and the outer eye distance
- the width and the height of the nose
- the width and the height of the mouth
- the width of the left and of the right eye.

Figure 1 visualizes all of the 8 facial features. Note, that some of these might vary a lot. For instance, the estimated height of the mouth will vary when people open or close their mouth. The features are therefore not invariant to facial expressions but can nevertheless characterize the face of an individual. Note, that we extract only a small subset of possible distances with clear interpretation that we expect to be characteristic for the human face. In this way we maintain comprehensibility of our representation. Moreover, too many features would lead to overfitting during the learning process as our data set was small at this stage of the development process.

The Bayesian Classifier that we used for classification assumes conditional independence of features. This assumption is obviously violated for the proposed features meaning that the Bayesian Classifier is no longer guaranteed to be optimal. However, Bayesian Classifier are often successful although their assumptions might be violated. In order to allow for a good representation of the conditional probability distribution, we introduce another assumption that the conditional probability for each features is normally distributed. Hence, the learned model for a person can be

represented by mean and variance per feature. Note, that a normal distribution is a reasonable assumption for geometric features from the Kinect face model as shown in [3]. However, this introduces yet another assumption that might be violated to some degree. The Bayesian Classifier is therefore no longer guaranteed to be optimal. However, it shows reasonable performance in many classification problems as well as in our preliminary evaluation in [3].

With recognition rates of up to 80% on a preliminary data set with only 5 people, this first prototype was far from being sufficiently accurate for reliable access control. However, it proofed, that geometric features from the Kinect's models can be used to recognize people. This first system was not exploiting the redundancy of the records among consecutive frames as its prediction was on a frame by frame basis. Moreover, the feature set was extremely small.

2.2 Second Prototype

To overcome the weaknesses of our first prototype, we introduce more features and feature aggregation in the second version of our prototype and tested different classifiers to analyse the quality of the feature set. One feature that is expected to be particularly predictive for a person's identity is his or her height. The height varies a lot between different individuals and can be estimated using the Kinect's skeleton model. The model provides the position for 25 joints in both, the image plane and in space. We extract Euclidean distances in the same way as we extract them from the face model. In order to capture meaningful features from the model we consider distances from a selection of adjacent skeleton joints. In addition, we extract features between joints that are not adjacent if we expected the feature to be characteristic for a person. In particular, we wanted to represent a person's height and his or her shoulder width. Figure 1 shows all of the proposed 23 features that are considered from the skeleton model.

We introduced feature aggregation to this version of our prototype. Frames are still processed separately to extract the set of features from them leading to 31 values per frame. When a person approaches the Kinect sensor, up to 15 frames were considered during our experiments. As the Kinect's models have high computational demands, the low frame rate did not allow for more records in most situations. All of the 15 records aim to measure the same set of distances. In order to benefit from this redundancy, we aggregated the sequence into a single value per feature using one of 6 aggregation methods. As the most prominent aggregation method for real values, we used the mean in our experiments. In order to be more robust against outliers, we also analysed the median and four variants of truncated mean which can be seen as intermediate aggregation methods between mean and median.

The larger the distance to an object, the more noise can be observed in depth images. This makes approximation of the facial shape more difficult leading to lower quality of the Kinect's models. We therefore expect the measures for the proposed geometric features to be particularly noisy for records that were taken far away from the sensor.

As these measures might have a negative impact on the quality of the aggregated features, we consider only a subset of the closest *N* records during our experiments.

In order to select a good classifier for our system, we use Rapid Miner [6] to evaluate the quality of our feature set. We tested two classifiers in our experiments, k-Nearest Neighbour (k-NN) [5] and Linear Discrimination Analysis (LDA) and report a recognition rate of up to 88% for k-NN and up to 89% for LDA on a data set with 37 individuals.

These recognition accuracies are a significant improvement over our previous system while the aggregated features are still as comprehensible as the previously used raw features. Most importantly, the aggregation of feature values was shown to improve the recognition accuracy significantly. In order to be used in an access control system, we aim to further increase our system's performance.

2.3 Third Prototype

As based on the Kinect's face and skeleton model, the first two versions of our prototype are not easily adoptable to the use of other sensor devices. Moreover, the Kinect's face and skeleton model have high demands with regard to hardware. This might be a problem once the system is in use on site where such machines are not available or increase the costs dramatically. Hence, we wanted to become independent of the Kinect SDK's advanced capabilities while we still use the sensor and its low-lever API. The subset of the provided functionality that we use in the third version of the system is available for many other low-cost sensors. We acquire colour, infrared and depth frames as well as a mapping between these data sources. This functionality is also offered in OpenNI [7] for a variety of different sensors.

Although the recognition accuracies using aggregated geometric features are a significant improvement over the first version of our prototype they are not yet sufficient for reliable access control. However, they have shown that geometric features can contribute to reliable recognition in a comprehensible manner. To explore additional features and to improve the system's accuracy, we currently focus on colour, infrared and depth data directly which were not used in the previous systems. We aim to extract comprehensible features from these images as an intermediate representation. These features can again be geometric features as the distances between certain feature points but they are not restricted in this way and more importantly, no longer based on the Kinect's models.

As a basis for feature extraction we decided to generate frontal view images of detected faces. When a person approaches the sensor, his or her head and face are detected. We also estimate the person's head pose meaning that the exact position and orientation of a person's face is known. As the depth frame provides spatial information, this allows to render the scene from a normalized position in front of a person's face. In this way we obtain depth images of detected faces that that are aligned in a predefined position.

Given the mapping from depth frame to the colour frame respectively the infrared frame, it is further possible to use these as a texture. Hence, we are able to render frontal views of a detected face using either the colour frame or the infrared frame as a texture. Three different kinds of frontal views of a person's face can be computed in this way. As the system is currently under development, we want to focus on this part of the data processing pipeline in what follows. These first steps as a part of the data processing pipeline are sufficient to point out the idea of how such a system can be modelled using existing modelling frameworks.

3 Modelling Frameworks and Tools

All three prototypes implement a data processing pipeline that transforms the acquired sensor data into a compact and characteristic feature representation through a sequence of small data transformations. Every single transformation takes one or more of the previously computed representations as its input and computes a new representation from them. As only the final outcome is of any interest while intermediate representations are solely used for the computation of the final outcome, all of our prototypes can me modelled intuitively by focusing on the data flow only. In fact, we will show that the control flow can be derived from the data flow in our example.

In the example presented in this paper, we are focusing on the newest version of our prototype. As the final recognition is not implemented to date, we will focus on the generation of frontal view images of peoples' faces which will be part of the final data processing pipeline. These frontal view images can be obtained from colour, infrared and depth data by rendering the acquired data from a changed viewport. In order to do so the face position and its orientation have to be estimated precisely. Our newest prototype approaches this task in four steps based on a single depth frame.

We show two example models of our prototype using two different modelling tools that were generated using the modelling framework Cinco. In what follows, we will first introduce the reader to the modelling framework Cinco and to Dime, the most complex Cinco-Product to date. We will further show a small custom Cinco-Product that models the data flow only and is tailored to the needs of our prototypes' models.

3.1 Cinco

Cinco is a meta modelling framework for graphical Domain Specific Languages that is developed at TU Dortmund University since 2014 [8] [9] [10]. It is based on the popular Eclipse Open-Source IDE and allows for the generation of Cinco-Products that are themselves based on the Eclipse IDE. Graphical Domain Specific Languages in Cinco are based on the concept of directed graphs meaning that a predefined set of custom nodes and edges is defined for a particular Cinco-Product. The meta modelling framework allows to define the appearance for each kind of node and edge and allows to constrain their connectivity. In this way it is possible to allow certain

edges to connect only very particular kinds of nodes, but many other ways of constraining the graphical language are possible.

To enable rich features in Cinco-Products, the framework implements the concepts of hooks which allows to programmatically adjust the graph in case of a particular event. Such an event can be that a node was moved on the canvas or that it was removed from it. In particular, this allows to implement custom spatial arrangement of multiple nodes relative to one another but many other applications are possible.

As a meta modelling framework Cinco is used to generate modelling tools that are referred to as Cinco-Products. Due to the only assumption that a graphical Domain Specific Language is a directed graph, Cinco is very flexible and allows to generate modelling tools for a wide range of applications. Cinco itself does not associate any semantics with the graphical language but allows for the generation of an API that can be used to generate code from the graph models or to interpret them otherwise. Particularly interesting for our example models, edges can be used to represent both, control flow and data flow.

3.2 Dime

As the most complex Cinco-Product to date, Dime is the prime example of the Cinco's flexibility. As a Cinco-Product, Dime defies a set of nodes and edges, their appearance and also constraints the way they can be connected. While nodes represent situations during model's execution, edges are used to model both, control flow and data flow.

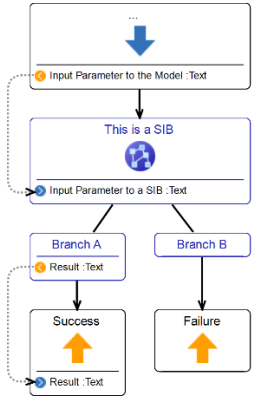

Fig. 2. Minimal example of a Dime model.

The most important nodes are the so called Service Independent Building Blocks (SIB) which represent executable code in the model. Every SIB has a list of input ports similar to function or method parameters in other programming languages. The functionality represented by the SIB relies only on the data provided by means of these input ports. The execution of a SIB can result in different cases which are

modelled using the concept of branches. Every SIB must have one or more branches as its successors, each representing one case. Depending on the outcome of the execution of the SIB, one branch is chosen that determines the SIB that is to be executed next. In this way branches are used to model the control flow of the system. In addition, the selection of a particular branch, any other outcome of a SIB will be represented as variable. In Dime the set of computed variables can be defined for every branch separately. This is often appropriate as there will be no computation result in some error cases or different results can be computed in different cases. Dime represents the outcome in terms of data by output ports that are associated with the branch nodes. Figure 2 shows a small example of a Dime model with one SIB that has two branches only one of which has an output port.

As Dime allows the user to model control flow and data flow, it has to provide at least two different kinds of edges. In fact, there are many more kinds of edges but for the sake of simplicity, we want to focus on data flow and control flow. The control flow starts at the start SIB which is represented by a blue arrow. To make the entry point unique, there can only exist one start SIB in every model. Together with the end SIBs they are the only SIBs that have no branches. During execution the start SIB will do nothing as it is solely used to represent the start of the control flow and potential input ports. The end SIBs are used to represent different cases as an outcome of the model's execution and their associated output ports. As such they serve a similar purpose as branches on the level of the entire model. In fact, this is how Dime allows users to model in a hierarchical manner, meaning that the whole model can be used as a SIB in other models. In order to define the control flow from the start SIB to one of the possible end SIBs, the user has to define the control flow. This is done by connecting branches as the outcome of SIBs to exactly one other SIB. Depending on the outcome of the execution of a SIB, this allows to define the successor separately for every case. The control flow must be defined for every possible branch to make the model valid.

When the control flow reaches a SIB, all of its input ports must be available. The required data can be provided by the initial input parameters on the level of the entire model or it can be provided as the outcome of a previously executed SIB. In any case, the variable to be used as an input must be defined using data flow edges. These edges are dashed and connect exactly one output port of a branch to one input port of a SIB. Moreover, the start SIB's ports can be used as output ports and the end SIBs' ports can be used as input ports. It is the user's responsibility to define the data flow and the control flow in such a way that required input data is available when a SIB is reached. Hence, the data flow imposes constraints on the control flow and vice versa, which can be exploited in the example that we present in this paper.

As Dime is a Cinco-Product and defines a set of nodes and edges, with clear interpretation, Dime is no longer as flexible as the use of Cinco for a tailored Cinco-Product. However, by modelling both, control flow and data flow its graphical models can express similar things as many programming languages in an intuitive fashion. Dime is still flexible in the sense that SIBs can have arbitrary functionality.

3.3 Custom Cinco-Product

Although Dime is suitable for many applications as it allows to model both, data flow and control flow, there are applications that can be modelled more intuitively in other ways. In our example, we are only interested in the final outcome of the computation, respectively the final data representation. As every data transformation depends on one or more data representations, an order of all data transformations is implicitly defined by the data flow. Hence, this example allows for modelling the data flow only while the control flow can be derived automatically.

For our second example we have therefore created our own Cinco-Product that allows for modelling the data flow only. There are three kinds of nodes, namely input representations (blue), output representations (green) and intermediate representations (white) and only one kind of edges to model data flow. All of these nodes represent a form of data that will be computed during the execution of the model if necessary. Note, that intermediate representations also represent a data transformation as they are computed from one or more other representations.

4 Example Models of our Prototype

All of our prototypes were developed in a way that allows to easily model their data processing pipeline using either Dime or a custom Cinco-Product. The recognition algorithm can be clearly separated into a sequence of data transformations as will be shown by means of the following two example models. We present example models for both of the modelling tools, Dime and our own Cinco-Product.

4.1 Dime Model Example

As Dime allows for modelling control flow and data flow, the data processing pipeline can be easily modelled in Dime. Each of the data transformations as part of the data processing pipeline can be represented as a SIB. The required input representations are inputs to the SIBs and will therefore be connected to the SIBs' input ports. In our example we want to focus on the data flow and we want to show that the control flow can be automatically derived from it. For the sake of simplicity this example is therefore limited to a single branch per SIB. When the model is used to generate code in future versions of our system, more than one branch will be necessary to handle exceptions in any of the data transformation steps. For instance, there might be no person visible in an acquired frame and hence no meaningful head pose can be computed.

Ignoring these exceptions in the current version of the model, every SIB has exactly one branch which is the success branch. The success branch is necessary to provide outputs of the computation, namely a new data representation. In the example new frames are acquired from each of the three sources as a very first step. The first three SIBs fulfil this purpose and provide colour, infrared and depth frames as output ports of their success branches. While colour and infrared frames are solely used as a texture for the generation of the frontal view images, the depth frame plays an

important role. As it provides information about the facial shape it can be used to approximate the head pose. In the newest version of our prototype, this problem is approached as a sequence of four refinements as shown in the second row of the model in Figure 3. First, the head position is roughly approximated from the raw depth image as given to the first SIB's input port. The success branch provides the head position approximation which serves as an additional input for the more precise head position computation. Both, the raw depth frame and the head position approximation are connected to the head position computation SIB's input ports. In a similar fashion, the head pose is computed in two consecutive steps. Finally, a precise estimate of the head pose is available which allows for rendering of frontal views.

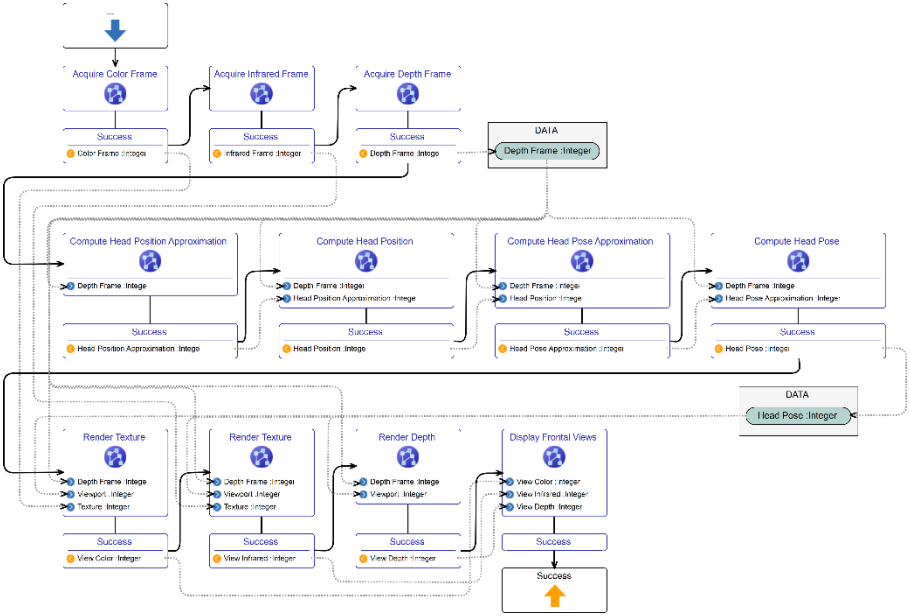

Fig.3. Dime model for the generation of frontal view images from colour, infrared and depth data.

Given the head pose, the first three SIBs in the third row of our model in Figure 3 render the frontal view images. As input parameters, all of them expect the raw depth image as acquired from the sensor which provides spatial information and also the precise head pose estimate which defines the viewport from which the scene is to be rendered. In addition, colour and infrared images are used as a texture leading to three different frontal view images of the detected face. For demonstration purposes the last SIB takes all of these images as inputs and displays them.

Note, that some data representations such as the depth image and the head pose are used more than once. In Dime this requires the use of a data context that holds

variables and allows for them to be used multiple times. In general, this is necessary as SIBs can change the value of their input parameters. However, in our example all of the input variables are only read which allows for using them multiple times in an arbitrary order.

Our goal is to define reusable components from the data processing pipeline of our final version of the admission control system and to provide them as SIBs in Dime. Not only would this allow to modify the system in a very intuitive way and would allow non-programmers to adjust the system at any time, but also would this allow for building similar systems from the existing components in a very easy way. Especially people with little to no programming skills would be enabled to create advanced systems that detect people, their head poses and many other things depending on the capabilities of the palate of provided SIBs.

4.2 Custom Cinco-Product Model Example

As only the displayed image the displayed image in this example as the final outcome of our model's execution is of any interest, the order of execution is irrelevant as long as required input representations are available for every data transformation in the data processing pipeline. In order to exploit this property, we use our custom Cinco-Product to model the data flow of the pipeline only. Every node represents data while intermediate data representations (white) implicitly represent data transformations that define how the new data representation can be obtained from others. In the example, raw sensor data is given as input representations (blue). In particular, these are colour, infrared and depth frames that were acquired using the Kinect sensor. Per execution of the pipeline one frame from each source is available and all of them can be used to obtain the final data representation. As the newest version of our prototype does not implement the final recognition of an individual yet, we limit the example to the generation of a collage of frontal view images from all three sources. The frontal views will later be used to extract facial features for the recognition of individuals. In order to render frontal view images of faces, the head pose must be known which defines the viewport from which the scene has to be rendered.

As the prototype is the same one that was used for the Dime model example, the computation of the head pose is again approached in four steps. First, the head position is approximated in the raw depth frame. The data representation *Head Position Approximation* implicitly represents the data transformation from a raw depth frame to an approximation of a person's head position. The new data represents only the approximation of the head position and no longer the depth frame. It is therefore a significantly smaller data representation than the *Depth Frame*, which was provided as an input representation. In a second step, the *Head Position* is computed more precisely from the raw *Depth Frame* and from the *Head Position Approximation*. The new data representation therefore depends on two others which have to be available before the *Head Position* can be computed. Hence, the data flow as defined in the model in Figure 4 imposes constraints on the order of data transformations.

Note, that the model in Figure 4 does not define the control flow but only the data flow. As the data flow imposes constraints on the order in which data representations must be available, a possible control flow can be deduced automatically from the topological order of the graph. More precisely every input representation must be available before a new data transformation can be applied. In our example, this means that *Head Position Approximation*, *Head Position*, *Head Pose Approximation* and *Head Pose* must be computed in exactly this order before any of the other data representations can be derived. For the generation of the separate frontal view images, the order can be arbitrary as they do not depend on one another but only on input representations and the *Head Pose*. Finally, the *Merged Image* must be computed at the very end. This is also defined as the final output representation (green) of our model. Any case in which the order of computation is irrelevant allows for parallelism. In our example, the generation of the separate frontal view images can in fact be performed in parallel once their input representations are available.

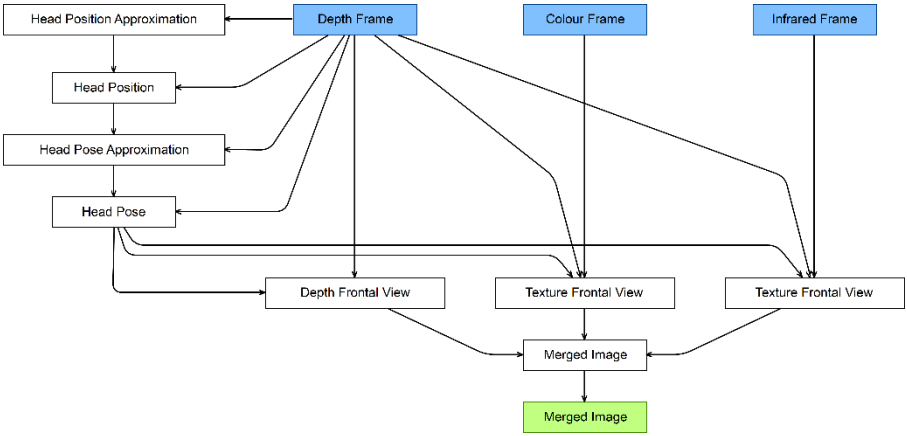

Fig. 4. Cinco-Product model for the generation of frontal view images from colour, infrared and depth data.

As an alternative to SIBs in Dime our custom Cinco-Product has strong focus on data flow. While this simplifies modelling of a data processing pipeline, there is no easy way of modelling side effects, defining the order of execution etc. This Cinco-Product is nevertheless useful for data oriented applications such as processing pipelines in computer vision systems similar to the one in our example.

5 Conclusion

In this paper, we presented three generations of prototypes for a contactless admission control system with high potential to be modelled with available modelling frameworks and tools.

We presented the three versions of our prototype and their commonalities and differences. In particular, we focused on the data processing pipeline which all prototypes implement in a similar fashion. This part of the system can be modelled intuitively with modelling tools such as Dime or our custom Cinco-Product.

In order to show our prototypes' potential to be modelled, we introduced the reader to Cinco, Dime and our custom Cinco-Product. Focusing on the first part the newest processing pipeline, we show examples of models in both tools. We discussed the possibility to derive the control flow automatically from a specification of the data flow in the data processing pipeline.

We continue to develop the most recent version of our prototype in a way that maintains its high potential to modelled visually. This will enable us to define reusable components from the data processing pipeline of our final admission control system and to provide them as SIBs in Dime. Moreover, we aim to model the system using a similar Cinco-Product to the one that we presented in this paper. Not only would this allow to modify the system in a very intuitive way and would allow non-programmers to adjust the system at a later stage, but also would this allow for building similar systems from the existing components in a very easy way. Especially people with little to no programming skills would be enabled to create advanced systems that detect people, their head poses and many other things. Once a set of powerful SIBs, respectively data transformations is developed for computer vision related applications, it can be extended continuously leading to a rich palate of SIBs. Depending on the extend of this palate, this would allow for modelling a wide range of computer vision related applications. In the long term, such a palate could also be extended to an even broader range of systems that implement any kind of a data processing pipeline.

Acknowledgment

This work was supported, in part, by Science Foundation Ireland grant 13/RC/2094 and co-funded under the European Regional Development Fund through the Southern & Eastern Regional Operational Programme to Lero - the Irish Software Research Centre (www.lero.ie).

References

[1]. G. S. Gagandeep Kaur and V. Kumar, "A review on biometric recognition," International Journal of Bio-Science and Bio-Technology, vol. 6, no. 4, pp. 69–76, 2014.

[2]. Z. Zhang, "Microsoft kinect sensor and its effect," IEEE MultiMedia, vol. 19, no. 2, pp. 4–10, 2012.

[3]. F. Gossen, "Bayesian recognition of human identities from continuous visual features for safe and secure access in healthcare environments," in Design Technology of Integrated Systems in Nanoscale Era (DTIS), 2015 10th International Conference on, 2015.

[4]. I. Marqués and M. Graña, Computational Intelligence in Security for Information Systems 2010: Proceedings of the 3rd International Conference on Computational Intelligence in

Security for Information Systems (CISIS'10), ch. Face Processing for Security: A Short Review, pp. 89–96. 2010.
[5]. T. Cover and P. Hart, "Nearest neighbor pattern classification," Information Theory, IEEE Transactions on, vol. 13, no. 1, pp. 21–27, 1967.
[6]. "Rapid miner." https://rapidminer.com/. Accessed: 2016-02-02.
[7]. "Openni 2 sdk." http://structure.io/openni. Accessed: 2016-04-01.
[8]. S. Naujokat, L.-M. Traonouez, M. Isberner, B. Steffen, and A. Legay, Leveraging Applications of Formal Methods, Verification and Validation. Technologies for Mastering Change: 6th International Symposium, ISoLA 2014, Imperial, Corfu, Greece, October 8-11, 2014, Proceedings, Part I, ch. Domain-Specific Code Generator Modeling: A Case Study for Multi-faceted Concurrent Systems, pp. 481–498. Springer Berlin Heidelberg, 2014.
[9]. S. Naujokat, M. Lybecait, B. Steffen, D. Kopetzki, and T. Margaria, "Full generation of domain-specific graphical modeling tools: a meta modeling approach." under submission, 2015.
[10]. "Cinco scce meta tooling framework." http://cinco.scce.info/. Accessed: 2016-04-01.

Моделирование конвейера распознавания людей в системах контроля доступа

[1] *Ф. Гёссен <frederik.gossen@lero.ie>*
[1] *Т. Маргариа <tiziana.margaria@lero.ie>*
[2] *Т. Гёке <thomas.goeke@systeam-gmbh.com>*
[1] *Lero - The Irish Software Research Centre, Лимерикский университет, Tierney Building, Лимерикский университет, V94 NYD3, Ирландия.*
[2] *SysTeam GmbH, Технопарк, Martin-Schmeißer-Weg 14, 44227 Дортмунд, Германия.*

Аннотация. В работе представлено три поколения прототипов бесконтактной системы пропуска, распознающей людей по их визуальным особенностям при подходе к сенсору. Назначением системы является увеличение удобства пользователей за счет минимизации взаимодействия. Такая система может быть особенно полезна людям с нарушениями тех или иных функций. Для получения и обработки данных в системе используется Microsoft Kinect 2, недорогой сенсор глубины, и связанные с ним инструменты разработки. Распознавание приближающегося к сенсору индивида основано на построении компактного характеристического представления; для этого вычисляется множество геометрических особенностей индивида и применяются методы агрегации для последовательности кадров. Каждый из трех прототипов реализуют некоторый конвейер обработки данных; конвейер преобразует данные, полученные от сенсора, в компактное характеристическое представление путем последовательного применения простых трансформаций. Каждая отдельная трансформация получает на вход одно или несколько представлений, полученных на предшествующих стадиях, и строит по ним новое представление. Примеры моделей, представленные в этой статье, фокусируются на генерации фронтальных изображений лиц людей — это часть конвейера обработки данных последнего прототипа. Фронтальные изображения могут быть получены по данным о цвете, инфракрасном излучении и глубине путем

рендеринга сцены относительно меняющейся области просмотра. Такой конвейер может быть представлен исключительно потоками данных между трансформациями. В статье показывается, как моделировать прототипы с помощью таких сред и инструментов, как Cinco и Cinco-Product Dime. Эти средства позволяют интуитивным образом моделировать потоки данных в конвейерах.

Ключевые слова: визуальное моделирование, распознавание лиц, распознавание людей, машинное зрение.

DOI: 10.15514/ISPRAS-2016-28(2)-14

Для цитирования: Гёссен Ф., Маргариа Т., Гёке Т. Моделирование конвейера распознавания людей в системах контроля доступа. Труды ИСП РАН, том 28, вып. 2, 2016 г., стр. 205-220 (на английском). DOI: 10.15514/ISPRAS-2016-28(2)-14

Список литераторы

[1]. G. S. Gagandeep Kaur and V. Kumar, “A review on biometric recognition,” International Journal of Bio-Science and Bio-Technology, vol. 6, no. 4, pp. 69–76, 2014.
[2]. Z. Zhang, “Microsoft kinect sensor and its effect,” IEEE MultiMedia, vol. 19, no. 2, pp. 4–10, 2012.
[3]. F. Gossen, “Bayesian recognition of human identities from continuous visual features for safe and secure access in healthcare environments,” in Design Technology of Integrated Systems in Nanoscale Era (DTIS), 2015 10th International Conference on, 2015.
[4]. I. Marqués and M. Graña, Computational Intelligence in Security for Information Systems 2010: Proceedings of the 3rd International Conference on Computational Intelligence in Security for Information Systems (CISIS’10), ch. Face Processing for Security: A Short Review, pp. 89–96. 2010.
[5]. T. Cover and P. Hart, “Nearest neighbor pattern classification,” Information Theory, IEEE Transactions on, vol. 13, no. 1, pp. 21–27, 1967.
[6]. “Rapid miner.” https://rapidminer.com/. Accessed: 2016-02-02.
[7]. “Openni 2 sdk.” http://structure.io/openni. Accessed: 2016-04-01.
[8]. S. Naujokat, L.-M. Traonouez, M. Isberner, B. Steffen, and A. Legay, Leveraging Applications of Formal Methods, Verification and Validation. Technologies for Mastering Change: 6th International Symposium, ISoLA 2014, Imperial, Corfu, Greece, October 8-11, 2014, Proceedings, Part I, ch. Domain-Specific Code Generator Modeling: A Case Study for Multi-faceted Concurrent Systems, pp. 481–498. Springer Berlin Heidelberg, 2014.
[9]. S. Naujokat, M. Lybecait, B. Steffen, D. Kopetzki, and T. Margaria, “Full generation of domain-specific graphical modeling tools: a meta modeling approach.” under submission, 2015.
[10]. “Cinco scce meta tooling framework.” http://cinco.scce.info/. Accessed: 2016-04-01.