Proceedings of the 1997 IEEE International Conference on Systems, Man, and Cybernetics (SMC'97) • Orlando, FL • Oct. 12-15, 1997

DATA REPRESENTATION AND ORGANIZATION FOR AN INDUSTRIAL MULTISENSOR INTEGRATION ARCHITECTURE

Michael D. Naish and Elizabeth A. Croft
Industrial Automation Laboratory
Department of Mechanical Engineering
University of British Columbia
Vancouver, B.C. V6T 1Z4, Canada
Email: ecroft@mech.ubc.ca

ABSTRACT

An open architecture for intelligent multisensor integration in an industrial environment is being developed. A logical sensor model is used to represent both real and abstract sensors within the architecture, allowing for the ready addition or replacement of sensors. Processing algorithms are also encapsulated by logical sensors. Objects are modeled using a connected graph structure wherein each node represents a salient feature of the object. Interactive training is used to determine the logical sensors required to extract desired features from objects. Extracted features are identified by the user and become part of the model. Once trained, the system can use object models for identification and classification purposes.

1. INTRODUCTION

Modern automated machinery utilizes a variety of independent sensors to collect and process information. Recent application examples include poultry grading [1], material surface inspection [2], printed circuit board inspection [3], catfish processing [4], produce classification [5], and herring roe grading [6]. In most of these applications, ad-hoc methods are used to develop a sensor integration system to monitor the process. Such systems tend to be difficult to understand, maintain, and upgrade. This is a problem for industrial users whose requirements in terms of speed, feature recognition, accuracy, and other process monitoring parameters invariably change over time.

To address these needs, a new open architecture for intelligent multisensor integration in an industrial environment is being developed. This framework allows for the computational evaluation and understanding of sensor uncertainty and data validity through the comparison of sensor data in a common format. A logical sensor model [7,8] is used to represent both real and abstract sensors within the architecture, allowing for the simple addition or replacement of sensors. In this model, structured control flow facilitates dynamic sensor adjustment based on the overall system requirements and individual sensor performance.

In this paper, the determination of a suitable robust data abstraction scheme is addressed. In this scheme, the representation of qualitative information such as non-uniform shape, texture, and other features required for inspection and grading tasks is facilitated.

For non-uniform product grading, the representation of objects must allow for the quantification of deviations from an ideal model. The proposed data representation, based upon work by Tomita and Tsuji [9], addresses this problem by representing objects in terms of user specified features relevant to the grading task, in addition to available 'crisp' data. This high level of data abstraction permits the comparison of sensor information from diverse sources. The features are then used for discordance based identification and classification. Discordance methods are described by Murphy in [10].

The data representation presented herein is validated through example real world applications in the food processing industry; however, this architecture is suitable for a broad range of industrial applications, especially those involving non-uniform product grading.

2. BACKGROUND

The use of intelligent systems and sensor technologies, including machine vision, can be applied to many industrial processes. However, many intelligent sensor systems, while successful from the research perspective, are still too slow to be of practical use. This is problematic, since industry users expect computer systems to offer not only improved quality, repeatability, and reliability but also to maintain or increase production speeds.

In many industrial applications, grading, inspection, and process control tasks occur within the controlled environment of a plant. In most cases, the generic recognition problems considered by machine vision researchers are greatly simplified by a priori information. This information encompasses both the background against which objects must be segmented, and knowledge about the objects themselves. Typical production arrangements involve the use of conveyor belts along which the product moves and provides physical separation between objects. This separation eliminates the need for algorithms which perform well when objects are occluded; such algorithms are typically more computationally expensive. In addition, structured and predictable lighting is possible, further simplifying the object recognition task by ensuring that objects appear under the same intensity of light and shadow field.

3. SYSTEM OVERVIEW

An architecture has been developed for the purpose of multisensor integration within an industrial setting. The architecture is intended to be open, flexible, and easily configured to a variety of applications. As many industrial processes involve the evaluation and inspection of products, vision systems offer the primary source of information.

The architecture uses *logical sensors* (see Section 3.1, below) to encapsulate physical sensors and processing algorithms. The logical sensor hierarchy orders sensor data in a bottom-up manner. The raw data collected by the physical sensors is processed through different levels of logical sensors to produce high-level representations of sensed objects and features. While this approach may be slower than an ad-hoc implementation, where the required data is extracted directly from the raw sensor data, it offers considerable flexibility. High-level tasks such as non-uniform product grading may be implemented without regard to the specific sensing devices. The low-level physical sensors and low-level data processing routines are transparent. This allows for the addition, removal, and replacement of sensors within the architecture with the minimum amount of disturbance to the overall system. The architecture is illustrated in Figure 1.

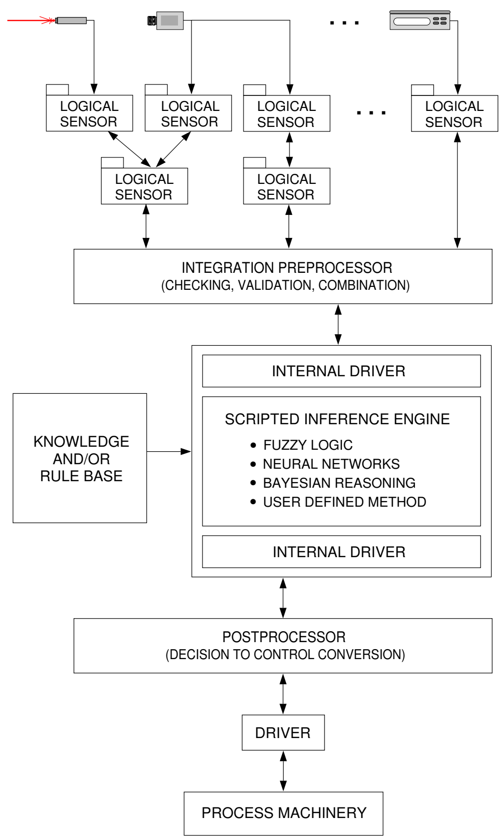


Figure 1: Overall system architecture.

Configuration of the architecture is an interactive process. A graphical user interface enables the user to select and configure logical sensors from a library. Links are created to indicate data flow. The system is trained by the user; object models, as described in Section 6, are constructed using data available from the logical sensors.

3.1 Logical Sensors

A logical sensor is an abstract definition for a sensor. Logical sensors were first defined by Henderson and Shilcrat [7] and later broadened to include a control mechanism by Henderson et al. [8]. This definition provides a uniform framework for multisensor integration by separating physical sensors from their functional use within a system. Figure 2 illustrates a logical sensor.

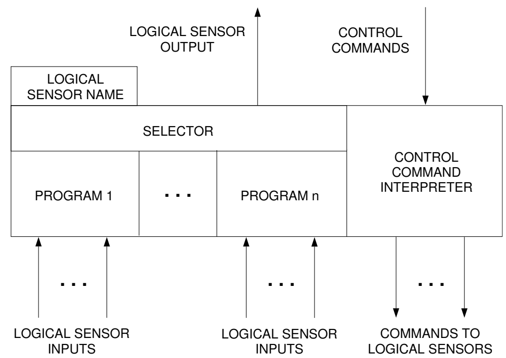


Figure 2: Basic components of a logical sensor. (Adapted from [8]).

Using this definition, physical sensors such as load cells, thermocouples, cameras, and lasers may be represented. The data from these sensors may also be combined and processed using a variety of available algorithms. In this way, "sensors," such as a line detector, which do not physically exist may be made available to the user. Output from a variety of logical sensors may be combined to extract complex features. Physical sensors may be replaced or added without disturbing the entire system – only the associated logical sensor need change.

4. EXTRACTION OF OBJECTS

Objects are extracted through the use of the appropriate logical sensors. The logical sensors which are available to the user depend on upon the types of physical sensors attached to the system. For example, a load cell would have a single logical sensor which provides measurement of an object's mass. On the other hand, a CCD camera may have a much larger number of associated logical sensors. The camera itself provides a raw image from which an edge detecting logical sensor can extract edges. Similarly, a line detecting logical sensor can extract straight line segments, or a colour logical sensor can extract relevant colour information, from the same raw image. Then, these sensors may be combined to detect compound objects.

Common logical sensors are available to the user from a library. The associated algorithms are chosen primarily for computational efficiency. Additional logical sensors may be added to the library to address special user needs, add new sensors, or introduce new algorithms. The user chooses logical sensors and/or combinations of logical sensors to achieve the desired level of performance. In general, this will involve a trade-off between accuracy and speed. After selecting the logical sensors to extract the desired information, the user must adjust the associated parameters to achieve the desired results. This is performed during the object modeling process outlined in Section 6.

Object types are determined by the extraction capabilities of the logical sensors. Various edge detectors, segmentation algorithms, texture identifiers, colour models, and other image processing routines each produce particular object types: lines, regions, textures, colours, etc. These types are used to identify the extracted objects and features.

5. PROPERTIES OF OBJECTS

Within the data representation, objects have two different types of properties, namely: physical object properties and relational properties (i.e. relationships to other objects). Physical object properties include position information, shape, colour, intensity, and texture. For each of these physical properties an associated confidence level is provided by the logical sensors. Relational properties describe the object relative to others. Symmetry, adjacency, relative position, and relative orientation are examples of relational properties.

6. OBJECT MODELING

6.1 Model Structure

Each object recognized by the system is represented by a connected graph. All but the simplest objects are represented by a hierarchy of nodes, where each node of the graph represents a salient feature of the object. The components are shown in Figure 3 and described in the following sections.

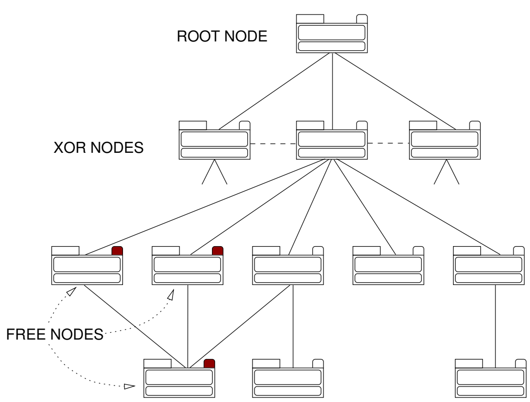


Figure 3: Graph structure for object representation.

The object model, in contrast to the logical sensor hierarchy, is a top-down representation of an object. The object itself, represented as a root node, is the highest-level of abstraction. Traversing down the graph, features of the object are represented as individual nodes. Each subsequent level becomes more and more detailed. This enables compact and efficient object models. Only the level of detail required for identification or classification need be specified.

6.1.1 Node: Each node of the graph represents a recognizable object or feature. These may be complex features extracted from information provided by one or more logical sensors. Each node may be a parent node, that is, it is associated with one or more child nodes which further details features of the parent node. For example, a parent node may be a digital image of an apple while child nodes may include the colour of the apple image, the size of the apple image, etc. Alternatively, a node may contain simple crisp measurements provided by a single sensor, for example, mass and temperature. The node structure contains the name of the object, the type of object, and the object properties. In addition, the logical sensor(s) used to extract the data are detailed here, along with relevant operating parameters. Features which are not always present in a parent node are marked by a *free node tag*. Finally, links to parent and child nodes are maintained within the structure. This is illustrated in Figure 4. The root node, encapsulating the representation of an entire object, is a special case.

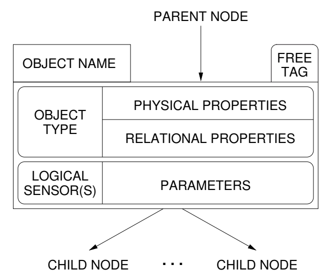


Figure 4: Basic components of object node.

6.1.2 Unconditional Link: An unconditional link is used to represent a *parent-child* relationship between nodes; it is graphically illustrated by a solid line. Logical sensors are used to extract child objects from parent objects. Extraction may involve segmentation, merging, or grouping of parent objects. Each child node specifies the logical sensor(s) used for processing and the associated parameters.

6.1.3 Conditional Link: A conditional link is used to represent an *exclusive OR* relationship. This type of link is depicted as a dashed line. The conditional link is used to group child objects from a single parent node. This grouping indicates that only one of the connected child objects will be recognized from a given parent node.

The conditional link is particularly powerful for the representation of most non-uniform products which cannot be uniquely defined. Variability in features, common to most non-uniform products may be accommodated through the definition of acceptable variations.

6.2 Initial Model Development

An initial model is built for each object to be classified by the system. The creation of this model is an interactive process. The user starts by providing a name for the overall object; this is used to identify the root node. Logical sensors are chosen to extract features from the overall object and parameters are adjusted via an interactive guided search until the desired results are achieved. Upon the successful extraction of each feature (which is also an object), a child node associated with the extracted object is created. This node stores the object name and logical sensor parameters. The properties of the object are also stored in the node. Property values are updated for each object presented to the system and are determined by the object type.

This process may be repeated to extract features from each new node until all relevant objects have been defined in the model. As objects are further subdivided new descendant nodes are created.

6.3 Model Refinement

Once the initial model has been created, refinement is often necessary to enable the recognition of objects not present in the first model or to define acceptable deviations of objects from the original model. Similar to the initial model building, refinement is an iterative process. Training proceeds by presenting additional objects to the system. The greater the number objects used for training, the more robust the object model will be.

The user may test the performance of system by allowing the system to perform automatic object recognition. If the system fails to recognize an expected object, the user is prompted to refine the model. This is repeated until all of the training objects are properly analyzed by the model. As models are built for each class, the distinguishing features (i.e. disparities between objects) are stored and used for classification and grading purposes.

The object model may be refined in the following ways:

- Refinement of parameters: If objects are not extracted properly, it is necessary for the user to adjust the logical sensor parameters. The proper values are determined by a guided search as in the initial model creation. After adjusting parameters, the relevant objects are selected and identified by the user to update the model.

- Refinement of properties: Properties may require refinement if an expected object is extracted but not properly identified. In this case, the user selects the object and provides a name. The model will be revised accordingly.

- Refinement of relations: The value of the relational property between objects may be insufficient, causing the system to be unable to determine a desired relation. The user selects the pair of objects and inputs an index for the relationship. The missing relation will be added to the model.

- Declaration of conditional relationship: A conditional relationship is defined in cases where the extracted object is a unique variation of an object presently in the model. The user identifies the object and declares the object(s) which are conditional grouped. Since each object in the conditional grouping is unique, only one object in the group may be identified at any one time.

- Declaration of free object: Objects which are not always present are tagged as free objects. This contrasts with other objects which, if not tagged, are always expected to be extracted and identified from the parent node.

The above refinements are used when objects are not correctly extracted. Refinements may also be necessary in the case of an identification error. Identification errors arise when the there is a common attribute shared by two different objects. These are overcome by identifying distinguishing features present in either or both of the objects. The feature(s) may be added to the model so that the disparity may be used to distinguish between them.

In cases where this approach is unable to distinguish between the two objects, it is necessary to determine if the object models are correct; if not, the incorrect model must be redefined.

7. AUTOMATIC RECOGNITION AND GRADING

Automatic recognition is the ultimate goal of any image processing system. Once the model has been properly constructed, the system may be used to automatically analyze objects presented to the system sensors. Recognition proceeds in a top down manner from the root nodes of the model graph. The selection of a particular parent is contingent on the successful identification of all descendant objects. Should the system fail to find an expected object at a particular level, the system returns to the previous level and attempts to follow another branch. If a proper match cannot be found, the system issues an error message requiring the user to improve the object model.

There are two approaches which may be taken towards object classification and grading. The first is an extension of the object recognition problem. The entire object is identified based upon the features contained within the object model and extracted by the logical sensors. The identified object will fall into a predetermined classification. Extracted object properties may then be used to further evaluate the object based on attributes such as size, colour, and mass. This information may then be used in the control of automated equipment for the separation of the different classifications.

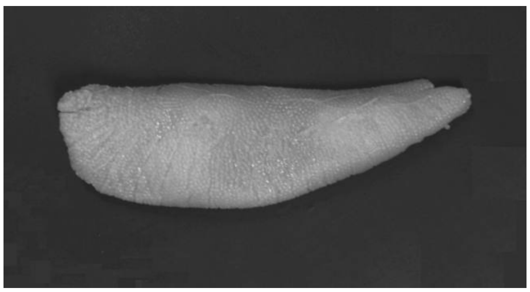
For complex objects it may be more efficient to define the object models somewhat differently. Instead of attempting to identify the entire object, only distinguishing features are extracted. The features may then used as input to an inference engine. The presence or absence of particular features and the associated object properties may then be used to classify the object into a particular grade.

8. EXAMPLE APPLICATION

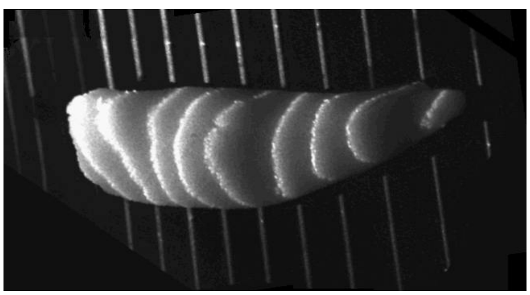
Herring roe grading is an example of a real world problem to which this system may be applied. The herring roe industry in Canada is valued at $100 million a year, catering primarily to the Japanese market. Appearance and texture of the salted herring roe are the primary factors influencing price. Proper classification allows processors to offer improved value to their customers.

Herring roe is assigned a grade according to aesthetic properties including colour, texture, size, and shape. Of these, all but texture are assessed visually; texture is assessed by tactile examination. The highest quality roe are light yellow and stain-free in colour, firm, over 75mm in length, and fully formed without twists, cracks, or breaks.

An example of *grade 2-H* or *henke* roe, under both ambient and structured light, may be seen in Figure 5. Figure 6 illustrates a number of features which may be extracted from the raw image data.



(a) Ambient light image.



(b) Structured light image.

Figure 5: Grade 2-H herring roe.

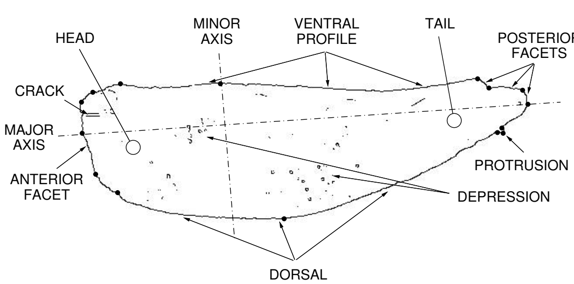


Figure 6: Herring roe features.

Shape is one of the most difficult features to assess, and has been attempted using computer vision [6, 11]. Other features not indicated in Figure 6 include mass and length. 3-D profile information extracted from the structured light image provides a measure of 'twist' and can be used to determine the existence of depressions – a characteristic of henke roe.

8.1 Sensors

The system makes use of two different vision systems. The first is an interlaced CCD camera under "ambient" light; the other is a passive scan CCD camera under structured laser light. Each of these physical sensors has a corresponding logical sensor. Image processing algorithms for edge detection, curvature estimation, dominant point detection, etc. are encapsulated in higher level logical sensors. The sensor hierarchy is shown in Figure 7.

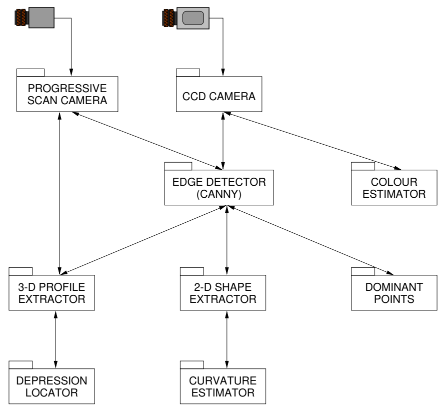


Figure 7: Sensor hierarchy for grade 2-H herring roe grading.

8.2 Object Model

The roe is modeled as shown in Figure 8. The model was constructed as outlined in Section 6. From the roe image (Figure 5), features such as the head, tail, ventral profile, and cracks were identified. Features are represented as descendant nodes extracted from parent nodes. The logical sensor and associated parameters used to extract the feature are associated with the object in the node. For example, the logical sensor used for depression location can extract either air bladders or depressions depending on the parameter values.

9. SUMMARY

To address the needs of industrial users, who require easily configurable, reliable, and flexible sensor systems, an open architecture for multisensor integration is being developed. Using a logical sensor model to encapsulate physical and abstract sensors, a bottom-up sensor hierarchy is constructed. This provides high-level decision making and inference modules with abstract information, independent from the low-level sensors. Sensor reconfiguration may proceed with minimal impact to the overall system.

The user interactively builds object models by selecting the logical sensors which will extract the desired features. This process is assisted by the system using a

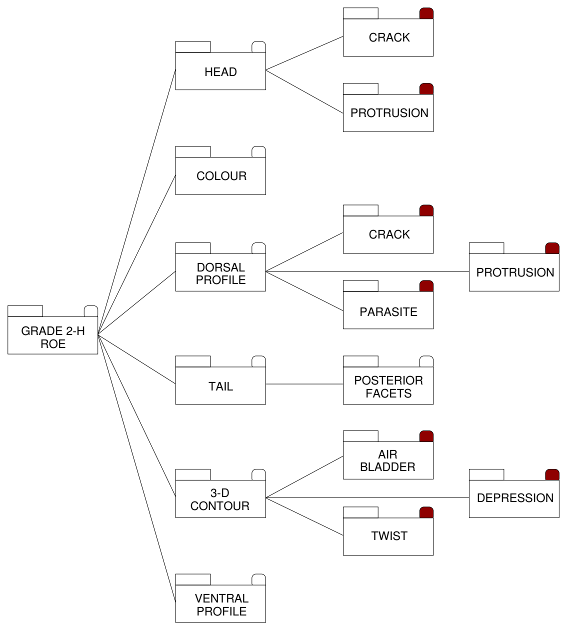


Figure 8: Object model for grade 2-H (henke) roe.

guided search strategy. Object models consist of a connected graph structure which indicates feature dependencies in a top-down manner. The root node encapsulates the entire object; descendants represent increasingly detailed features of the object.

The specification of the object models has a significant impact on the overall system speed. Should a number of unnecessary features be included, processing time may be wasted and optimal performance will compromised. For this reason, future directions may incorporate an expert system to aid in the creation of the model database and classification rules. This system could further ease the burden of model creation and ensure that representations are efficient in both space requirements (number of nodes) and processing time (appropriate selection and combination of logical sensors).

There are many applications for this architecture and approach to data representation. Examples include the sorting and grading of fruit and vegetables, meat, fish, and other non-uniform food products. In fact, the grading of any product, uniform or not, should be facilitated by the architecture. Other industrial applications which could benefit from sensor technologies, such as machining and manufacturing operations, will be the focus of future applications.

10. ACKNOWLEDGEMENTS

This work was supported by grants from the *Garfield Weston Foundation* and the *National Sciences and Engineering Research Council of Canada*.

REFERENCES

[1] W. Daley, R. Carey, and C. Thompson, "Poultry grading/inspection using color imaging," in *Proceedings of the SPIE - Machine Vision Applications in Industrial Inspection*, vol. 1907, pp. 124–132, 1993.

[2] J. H. Dan, D. M. Yoon, and M. K. Kang, "Features for automatic surface inspection," in *Proceedings of the SPIE - Machine Vision Applications in Industrial Inspection*, vol. 1907, pp. 114–123, 1993.

[3] G. Brown, P. Forte, R. Malyan, and P. Barnwell, "Object oriented recognition for automatic inspection," in *Proceedings of the SPIE - Machine Vision Applications in Industrial Inspection II*, vol. 2183, (San Jose, CA), pp. 68–80, 1994.

[4] P. Jia, M. D. Evans, and S. R. Ghate, "Catfish feature identification via computer vision," *Transactions of the ASAE*, vol. 39, pp. 1923–1931, Sep./Oct. 1996.

[5] R. M. Bolle, J. H. Connell, N. Haas, R. Mohan, and G. Taubin, "Veggievision: A produce recognition system," in *Proceedings of the 1996 3rd IEEE Workshop on Applications of Computer Vision*, (Sarasota, FL), pp. 244–251, 1996.

[6] E. A. Croft, C. W. de Silva, and S. Kurnianto, "Sensor technology integration in an intelligent machine for herring roe grading," *IEEE/ASME Transactions on Mechatronics*, vol. 1, pp. 204–215, Sept. 1996.

[7] T. C. Henderson and E. Shilcrat, "Logical sensor systems," *Journal of Robotic Systems*, vol. 1, no. 2, pp. 169–193, 1984.

[8] T. C. Henderson, C. Hansen, and B. Bhanu, "The specification of distributed sensing and control," *Journal of Robotic Systems*, vol. 2, no. 4, pp. 387–396, 1985.

[9] F. Tomita and S. Tsuji, *Computer Analysis of Visual Textures*. Norwell, Massachusetts: Kluwer Academic Publishers, 1990.

[10] R. R. Murphy, "Biological and cognitive foundations of intelligent sensor fusion," *IEEE Transactions on Systems, Man, and Cybernetics–Part A: Systems and Humans*, vol. 26, pp. 42–51, Jan. 1996.

[11] L. X. Cao, C. W. de Silva, and R. G. Gosine, "A knowledge-based fuzzy classification system for herring roe grading," in *Proceedings of the ASME Winter Annual Meeting on Intelligent Control Systems*, vol. DSC-48, (New York), pp. 47–56, 1993.