(12) STANDARD PATENT APPLICATION (11) Application No. **AU 2013248207 A1**
(19) AUSTRALIAN PATENT OFFICE

(54) Title
Method for superpixel life cycle management

(51) International Patent Classification(s)
G06F 19/26 (2011.01)

(21) Application No: **2013248207** (22) Date of Filing: **2013.10.24**

(30) Priority Data

(31) Number	(32) Date	(33) Country
13305157.3	**2013.02.08**	**EP**
12306419.8	**2012.11.15**	**EP**
12306418.0	**2012.11.15**	**EP**

(43) Publication Date: **2014.05.29**
(43) Publication Journal Date: **2014.05.29**

(71) Applicant(s)
THOMSON Licensing

(72) Inventor(s)
RESO, Matthias;JACHALSKY, Joern;ROSENHAHN, Bodo

(74) Agent / Attorney
FB Rice, Level 23 44 Market Street, Sydney, NSW, 2000

Abstract

Method for superpixel life cycle management

A method and an apparatus for life cycle management for superpixels associated to frames of a sequence of frames are described. An area occupied by a superpixel in a frame of the sequence of frames is monitored. In case the area of the superpixel becomes smaller than a specified first value, the superpixel is terminated (2) in the later frame. In case the area of the superpixel becomes larger than a specified second value, the superpixel is split (3) in the later frame.

Fig. 8

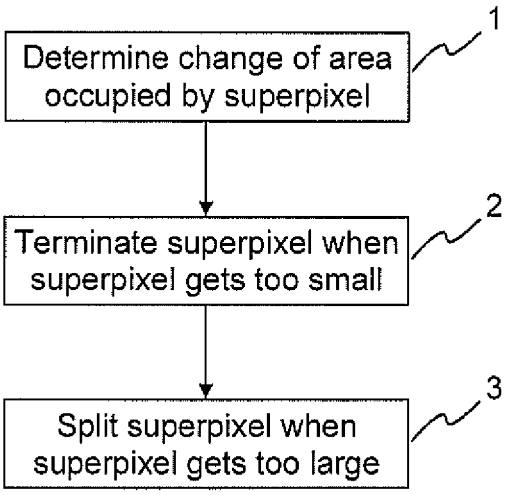
Determine change of area occupied by superpixel
1
Terminate superpixel when superpixel gets too small
2
Split superpixel when superpixel gets too large
3

Fig. 8

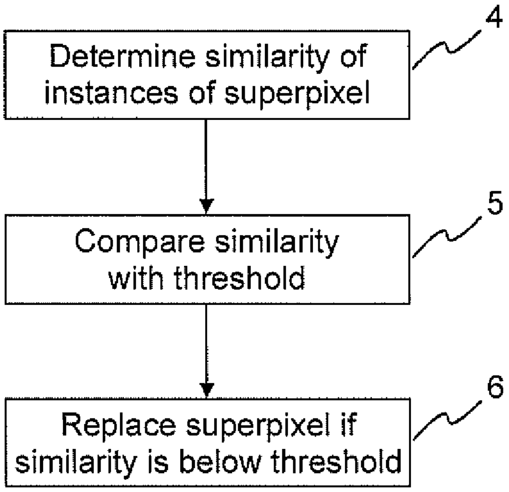
Determine similarity of instances of superpixel
4
Compare similarity with threshold
5
Replace superpixel if similarity is below threshold
6

Fig. 9

METHOD FOR SUPERPIXEL LIFE CYCLE MANAGEMENT

FIELD OF THE INVENTION

The present invention relates to a method and an apparatus for generating temporally consistent superpixels for a video sequence. More specifically, the invention addresses life cycle management for superpixels associated to frames of a sequence of frames.

BACKGROUND OF THE INVENTION

Superpixel algorithms represent a very useful and increasingly popular preprocessing step for a wide range of computer vision applications, such as segmentation, image parsing, classification etc. Grouping similar pixels into so called superpixels leads to a major reduction of the image primitives. This results in an increased computational efficiency for subsequent processing steps, allows for more complex algorithms computationally infeasible on pixel level, and creates a spatial support for region-based features.

Superpixel algorithms group pixels into superpixels. As indicated in X. Ren et al.: *"Learning a classification model for segmentation"*, IEEE International Conference on Computer Vision (ICCV) 2003, pp. 10-17, superpixels are local, coherent, and preserve most of the structure necessary for segmentation at scale of interest. As further stipulated in the above document, superpixels should be roughly homogeneous in size and shape. Though many superpixel approaches mostly target still images and thus provide only a limited or no temporal consistency at all when applied on video sequences, some approaches target video sequences. See, for example, O. Veksler et al.: *"Superpixels and Supervoxels in an Energy Optimization Framework"*, in Computer Vision - ECCV 2010, vol. 6315,

K. Daniilidis et al., Eds. Springer Berlin / Heidelberg, 2010, pp. 211-224, or A. Levinshtein et al.: *"Spatiotemporal Closure"*, in Computer Vision - ACCV 2010, vol. 6492, R. Kimmel et al., Eds. Springer Berlin / Heidelberg, 2011, pp. 369-382. These approaches start to deal with the issue of temporal consistency.

SUMMARY OF THE INVENTION

It is an object of the present invention to further improve the handling of temporally consistent superpixels associated to frames of a sequence of frames.

According to the invention, a method for life cycle management of a superpixel associated to frames of a sequence of frames comprises the steps of:

- terminating the superpixel in a frame in case an area of the superpixel becomes smaller than a specified first value; and
- splitting the superpixel in the frame in case the area of the superpixel becomes larger than a specified second value.

Accordingly, an apparatus for life cycle management of a superpixel associated to frames of a sequence of frames comprises:

- a terminating unit configured to terminate the superpixel in a frame in case an area of the superpixel becomes smaller than a specified first value; and
- a splitting unit configured to split the superpixel in the frame in case the area of the superpixel becomes larger than a specified second value.

Similarly, a computer readable storage medium has stored therein instructions enabling life cycle management of a superpixel associated to frames of a sequence of frames, which

when executed by a computer, cause the computer to:
- terminate the superpixel in a frame in case an area of the superpixel becomes smaller than a specified first value; and
- split the superpixel in the frame in case the area of the superpixel becomes larger than a specified second value.

According to the invention, a life-span, i.e. a duration, is determined for temporally consistent superpixels generated for video sequences. The duration is maximized without violating size and shape constraints. In general, the invention ensures a homogeneous size of the temporally consistent superpixels as superpixels that grow too large are split and superpixels that become too small are terminated. The maximized duration improves tracking of the superpixels, which in turn leads to a better scene and sequence understanding. The number of splits and terminations is preferably kept balanced, so that the number of superpixels per video frame is kept essentially constant. The decisions to split or terminate superpixels are based on information obtained from an analysis of the superpixels inside a sliding window that is shifted along the video volume.

Advantageously, in order to detect and correct a superpixel with an erroneous temporal consistency, the method further comprises the steps of:
- determining a similarity between an instance of the superpixel in a first frame and an instance of the superpixel in a later frame of the sequence of frames;
- comparing the determined similarity with a threshold; and
- replacing the superpixel in subsequent frames with a new temporally consistent superpixel in case the similarity is below the threshold.

Accordingly, the apparatus favorably further comprises:
- a determining unit configured to determine a similarity between an instance of the superpixel in a first frame and an instance of the superpixel in a later frame of the sequence of frames;
- a comparing unit configured to compare the determined similarity with a threshold; and
- a replacing unit configured to replace the superpixel in subsequent frames with a new temporally consistent superpixel in case the similarity is below the threshold.

A similarity check for the instances of a temporally consistent superpixel in a sliding window is introduced. The similarity between two or even more instances of a temporally consistent superpixel within the sliding window is determined. If it is below a certain threshold, the instances of the superpixel in all future frames of the sliding window are replaced by instances of a new temporally consistent superpixel starting at the first future frame. This approach provides a detection and start over strategy for superpixels with erroneous temporal consistency, which can be caused if the optical flow fails to provide a sufficiently reliable projection into the subsequent frame. Moreover, it provides a way to start over if there are sudden and disruptive changes in the video volume. Thus, it minimizes the distortion caused on subsequent processing steps by superpixels erroneously considered as temporally consistent. Thereby, it strengthens the flow of superpixels with a reliable temporal consistency.

For a better understanding the invention shall now be explained in more detail in the following description with reference to the figures. It is understood that the invention is not limited to this exemplary embodiment and that specified features can also expediently be combined and/or modified without departing from the scope of the present invention as defined in the appended claims.

BRIEF DESCRIPTION OF THE DRAWINGS

Fig. 1 illustrates a sliding window approach;

Fig. 2 shows image 5 of the so-called desk sequence;

Fig. 3 shows image 40 of the so-called desk sequence;

Fig. 4 depicts a contour image and a segment map for the image of Fig. 2 using life cycle management,

Fig. 5 depicts a contour image and a segment map for the image of Fig. 2 without life cycle management;

Fig. 6 shows a contour image and a segment map for the image of Fig. 3 using life cycle management;

Fig. 7 shows a contour image and a segment map for the image of Fig. 3 without life cycle management;

Fig. 8 schematically illustrates an implementation of a method for managing the life cycle of superpixels;

Fig. 9 schematically illustrates a method for detecting and correcting a superpixel with an erroneous temporal consistency;

Fig. 10 depicts an apparatus configured to implement the methods of Fig. 8 and Fig. 9.

DETAILED DESCRIPTION OF PREFERED EMBODIMENTS

In the following a brief look at the creation of superpixels based on energy-minimizing clustering from a unified perspective is given. It is based on the methods described in R. Achanta et al.: *"SLIC superpixels compared to state-of-the-art superpixel methods"*, IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 34 (2012), pp. 2274-2282, G. Zeng et al.: *"Structure-sensitive superpixels via geodesic distance"*, 2011 IEEE International Conference on Computer Vision (ICCV), pp. 447-454, and C. L. Zitnick et al.: *"Stereo for image-based rendering using image over-segmentation"*, International Journal of Computer Vision, Vol. 75 (2007), pp. 49-65. The three methods conceive the generation of superpixels as a clustering problem. Image pixels are seen as data points in a multidimensional space, in which each dimension corresponds to a color channel or image coordinate of the pixels. Superpixels are represented by clusters in this multi-dimensional space, where each data point can only be assigned to one cluster. This data point assignment determines the oversegmentation.

In order to generalize these approaches, an energy function E_{total} is defined, which sums up the energy $E(n,k)$ needed for all $n \in N$ data points to be assigned to a cluster $k \in K$:

$$E_{total} = \sum_{n=1}^{N} E(n, s(n)), \qquad (1)$$

where N is the total number of pixels and the function $s(\cdot)$ returns the label k of the cluster to whom a data point is currently assigned. For a given number of clusters K, an optimal over-segmentation in terms of energy can be created by finding a constellation of clusters that minimizes E_{total}. The three methods primarily differ in the definition of the energy term $E(n,k)$.

In order to minimize the total energy function E_{total}, the approaches use the iterative Lloyd's algorithm, which converges to a locally optimal solution. After a grid-like initialization, each iteration is performed in two steps, the *assignment*-step and the *update*-step. In the *assignment*-step, each data point is assigned to the cluster for which the energy term has its minimum. Based on these assignments, the cluster center parameters are re-estimated in the *update*-step. The iteration of these two steps continues until no changes in the *assignment*-step are detected or a maximum number of iterations has been performed.

The method described by Zitnick et al. models the distribution of data points in a cluster as a multivariate Gaussian with mean vector μ_k and covariance matrix $\sum_k$. The energy term is defined to be inversely proportional to the probability of a data point n belonging to a cluster k:

$$E(n,k) \propto \frac{1}{\mathcal{N}(n|\mu_k,\sum_k)}. \qquad (2)$$

The distributions of color and position of the image pixels are assumed to be statistically independent. Therefore, the joint probability distribution can be factorized into the multiplication of two Gaussians (one for color and one for the spatial extent). In the iterative optimization scheme, the covariance matrices in the color space are held fixed, while the covariance matrices for the spatial extent are updated in each iteration.

A simplified version of the algorithm above is presented by R. Achanta et al. and called *SLIC superpixels*. In contrast to the approach by C. L. Zitnick et al., a fixed covariance matrix for color and spatial extent is assumed, whose elements outside of the main diagonal are zero. Moreover, the variances for the color dimensions are equal as well as the variances for the spatial extent. As the fixed covariances can be omitted, only the means of the Gaussians matter. This leads to an energy term $E(n,k)$ that is directly proportional to the distance $D(n,k)$ between a data point n and the center of cluster k:

$$E(n,k) \propto D(n,k) \ . \qquad (3)$$

In R. Achanta et al. the following distance $D(n,k)$ between a data point n and the center of cluster k is introduced:

$$D(n,k) = \sqrt{d_{lab}^2(n,k) + \frac{m}{S} d_{xy}^2(n,k)}\, , \qquad (4)$$

where $d_{lab}(n,k)$ and $d_{xy}(n,k)$ are the Euclidean distances in CIELAB color space and image plane, respectively. The scaling factor m/S with $S = \sqrt{N/K}$ is due to the normalization of the two distances with respect to their maximum amplitude. The user-defined parameter m controls the trade-off between structure sensitivity and compactness of the generated superpixels.

In G. Zeng et al. the energy function E_{total} is split up into two energy terms weighted by a trade-off factor λ, whose functionality is similar to m in equation (4). The first energy term depends on the geodesic distance between the data points and the center of their assigned cluster k, while the second term penalizes clusters and thus superpixels, whose areas deviate from the average area:

$$E_{total} = \sum_{n=1}^{N} E_{geo}\big(n, s(n)\big) + \lambda \sum_{k=1}^{K} E_{size}(k). \qquad (5)$$

Using the geodesic distance allows to penalize color deviations along the path compared to a direct pixel comparison as used by R. Achanta et al.

As the spatial extent of the superpixels is limited a priori, it is sufficient in the *assignment*-step to search for pixels only in a limited search window around each cluster center. This leads to a significant reduction of the computational complexity. In order to enforce the spatial connectivity of the resulting segments, a post-processing step assigns split-off fractions, which are not connected to the main mass of the corresponding superpixel, to its nearest directly connected neighbor.

An improved approach for temporally consistent superpixels has recently been developed by the present inventors. It was inspired by the SLIC superpixels of R. Achanta et al. However, instead of introducing a temporal distance for the clustering in the video volume as done for the supervoxel approach of R. Achanta et al., which favors short-time consistency, the original 5D feature space for the superpixels is separated into a global color subspace comprising multiple frames and a local spatial subspace on frame level. The idea is that the color clustering is done globally and the spatial clustering locally. As a consequence, each temporally consistent superpixel has a single unified color center for all frames in scope and a separate spatial center for each frame. The latter one preserves the spatial locality on frame level and the former one ensures the temporal consistency. This approach is motivated by the observation that the color in a scene does not change rapidly in most cases. Therefore, the color of matching image regions and thus the mean colors of the associated superpixels is -in a first approximation- almost constant over multiple frames. In contrast, the positions can vary significantly depending on the motion in the scene.

For the generation of temporally consistent superpixels for video sequences, a sliding window comprising W consecutive frames is shifted along the video volume frame by frame. A visualization of this sliding window with $W=5$ is illustrated in Fig. 1. The frame t is the current frame and for the depicted example it is in the center of the sliding window. For this frame, the superpixels are to be generated. The previous frames $t-1$ and $t-2$ are the so called past frames. Their superpixel segmentation is fixed and will not be altered anymore. However, as they are inside the sliding window, their superpixel segmentation influences the superpixel generation in the current frame as well as the future frames through the global color center, which is updated in each iteration of the optimization scheme. The future frames in the sliding window, i.e. $t+1$ and $t+2$ in the depicted example, also have an impact on the superpixel generation in the current frame. However, in contrast to the past frames their superpixel segmentation is not fixed and thus can change in each iteration. The future frames help to adapt to changes in the scene, whereas the past frames are conservative and try to preserve the global color center found. The number of future frames F and the number of past frames P, with $W=F+P+1$, is preferably specified by the user and can also be asymmetrical. The more the number of past frames that are used exceeds the number of future frames, the more conservative the update of the color center is. The more the number of future frames that are used exceeds the number of past frames, the more adaptive the update is. When the sliding window is shifted forward, the results of the frame that leaves the sliding window are saved and a new frame that enters the sliding window is read.

As an initialization, the spatial centers of the latest frame are projected onto the new frame entering the sliding window. A weighted average of the dense optical flow computed over all pixels assigned to this center is used. The projection of the centers also initializes the search windows, which are important to reduce the complexity. With this sliding window approach, the proposed superpixel algorithm is adaptive to changes in the video volume.

In general, the generated superpixels should not only be temporally consistent, but also have a maximum life-span within the video volume. The latter allows for a good identification of a temporally consistent superpixel over time and is a first step towards a long-term tracking.

The adaptation of the superpixel generation to the video content can lead to steadily shrinking or growing superpixels that tend to violate the constraint of a rather homogeneous size. Therefore, it has to be decided when superpixels are to be terminated or when they need to be split up. This is called the life-cycle-management of temporally consistent superpixels. In the present approach this is accomplished by tracing the pixel area that a superpixel occupies in the frames of the sliding window. Therefore, a termination and a splitting condition are defined, which are described in the following.

It is assumed that the decrease of the area is linear in the sliding window for shrinking superpixels. Based on this assumption the following termination condition is created:

$$A(k,t+F)+\rho_T\cdot\Delta A(k)\leq 0, \qquad (6)$$

where $A(k,t+F)$ is the area of the superpixel k in the last future frame $t+F$ of the sliding window, $\Delta A(k)$ is the slope of the area for superpixel k, and ρ_T is a user-selected weighting parameter that can enforce or defer the termination. The slope $\Delta A(k)$ for $F>1$ is calculated as follows:

$$\Delta A(k)=\frac{A(k,t+F)-A(k,t+1)}{F-1}. \qquad (7)$$

In principle, $\Delta A(k)$ could be calculated using area samples from multiple future frames. The termination of the superpixel is finally done in the following way. The spatial center of a terminated superpixel is removed from the last frame of the sliding window, i.e. the frame $t+F$. As a consequence no pixels will be assigned to this superpixel when this frame becomes the current frame. Instead, the pixels previously assigned to this superpixel will be assigned to the adjacent superpixels. For the example that will be described below, two future frames, i.e. $F=2$, and $\rho_T=2$ are used.

If $F=1$, $\Delta A(k)$ can be calculated using the only future frame and the current frame.

In order to prevent an unlimited growth, the following splitting condition is defined:

$$A(k,t+F)+\rho_S\cdot\Delta A(k)\geq v\cdot\overline{A}\,, \qquad (8)$$

where $\overline{A}$ is the targeted average superpixel size, v is a user-selected parameter that controls the allowed maximum deviation from the average size, and ρ_S is a user-selected weighting parameter that can enforce or defer the splitting. For the example discussed below, $v = 1.5$ and $\rho_S = 0$.

If the splitting condition matches for a superpixel, two new superpixels are generated that replace this one. The replacement is done in the latest frame. While the global color center of the superpixel is copied, the two new spatial centers are shifted in opposite directions along the biggest eigenvector of the superpixel.

As the number of superpixels per frame should always match the user-defined value K, the last processing step of the life cycle management called superpixel balancing is applied after the terminations and splits. In order to keep the number of superpixels balanced, the difference δ between the number of terminations τ and the number of splits σ should always be zero:

$$\delta = \tau - \sigma. \qquad (9)$$

If the difference δ in equation (9) is positive, the largest δ superpixels are split in the same way described above. Analogously, if δ is negative, the smallest δ superpixels are terminated. If δ happens to be zero, the system is already in balance.

In some cases the optical flow fails for an image region so that the image content covered by the superpixel in the previous frame is not in the projected search window anymore. Moreover, there could also be sudden and disruptive changes in the video volume. In order to make the approach robust against those rare cases, it is best to initiate a start over. Thus, the life-cycle management is advantageously complemented with an additional start-over condition. After the superpixels were generated for the current frame, a similarity check is applied.

To this end, two instances of each temporally consistent superpixel are checked for similarity, one instance at frame $t+F$, i.e. the last frame of the sliding window, and the other instance at frame $t-P$, i.e. the first frame of the sliding window. For this similarity check different options are possible: histogram comparisons (e.g. chi-square test), feature and/or descriptor matching, color differences etc.

One implementation of such a similarity check is based on the following observation. Commonly, in those cases there is a significant difference in the average color of the superpixel k in the first frame $t-P$ and the last frame $t+F$ of the sliding window. If the Euclidean distance between the average colors differs more than a threshold Γ, superpixel k is replaced by a new one starting from frame $t+1$, i.e. the first future frame. Thereby, the local spatial centers are kept and a new global color center is introduced. The new color center is initialized with the average color of the replaced superpixel k for frame $t+F$. It was found that for the CIELAB color space $\Gamma = 30$ provides a good balance between error detection and preservation.

Within the sliding window approach the iterative optimization procedure explained above can basically be maintained for clustering. For each current frame of the sliding window a number of I iterations is performed, applying the *assignment*- and *update*-steps. In the *assignment*-step, for each pixel in each frame of the sliding window the distances to the superpixel centers are calculated using equation (4). The color distance d_{lab} is the Euclidean distance to the global color center, while d_{xy} is the distance to the local spatial center on frame level. In the *update*-step, the new global color centers are calculated using the color values of those pixels in all frames of the sliding window, which are assigned to this superpixel. The spatial centers are updated per frame using only the image coordinates of the pixels that are assigned to this superpixel in the corresponding frame.

At the beginning the sliding window is empty. The first frame of a video sequence to enter the sliding window is initialized by seeding the cluster centers in a grid-like structure including seed perturbing. This frame is positioned at index $t+F$. The remaining frames of the video sequence are initialized as described above. It should be mentioned that with the proposed sliding window approach the present algorithm for temporally consistent superpixels is also capable of streaming and thus in principle enables real-time processing.

In the following an example showing the benefits of the life cycle management (LCM) is given. Figs. 2 and 3 depict two images from a video sequence, namely the images 5 and 40 of the so-called desk sequence, respectively. Figs. 4 and 5 depict a contour map (left side) and a segment map for image 5 of the desk sequence generated with (Fig. 4) and without (Fig. 4) life cycle management. Figs. 6 and 7 show the contour map and the segment map for image 40 of the desk sequence generated with (Fig. 6) and without (Fig. 7) life cycle management. While the results at the beginning of the sequence, i.e. for image 5, are almost identical, the differences between the two versions for image 40 are evident, especially at the left and the right side of the contour images and the segment maps.

In addition to the qualitative analysis of the life cycle management presented above, below some quantitate results are shown. For this, the Variance of Area (VoA) as well as the mean iso-perimetric quotient Q are used. The Variance of Area is an indicator for the homogeneity of size and is calculated as the variance of the superpixel's area normalized by the mean area. The iso-perimetric quotient Q_k of a superpixel k is a measure for its similarity to a circle and thus a measure for the compactness. It is calculated as follows, where $L(k)$ is the perimeter and $A(k)$ the area of the superpixel k:

$$Q_k = \frac{4\pi A(k)}{L^2(k)}. \tag{10}$$

The following Table 1 shows the results generated for the Desk sequence using K =512 superpixels. It is evident that the Variance of Area is significantly higher without the life cycle management, whereas Q is identical. The results below are an average over all frames of the sequence.

	VoA	*Q*
WITHOUT LCM	0.17	0.79
WITH LCM	0.08	0.79

Fig. 8 schematically illustrates an implementation of a method for managing the life cycle of superpixels. In an optional first step a change of an area occupied by the superpixel between a current frame and a later frame of the sequence of frames is determined 1. In case the area of the superpixel becomes smaller than a specified first value, the superpixel is terminated 2 in the later frame. In case the area of the superpixel becomes larger than a specified second value, the superpixel is split 3 in the later frame.

Fig. 9 schematically illustrates a further aspect of life-cycle management, namely a method for detecting and correcting a superpixel with an erroneous temporal consistency. A similarity between an instance of the superpixel in a first frame and an instance of the superpixel in a later frame of the sequence of frames is determined 4. The determined similarity is then compared 5 with a threshold. In case the similarity is below the threshold, the superpixel is replaced 6 with a new temporally consistent superpixel in subsequent frames.

An apparatus 10 configured to perform the methods of Fig. 8 and Fig. 9 is schematically depicted in Fig. 10. The apparatus 10 has an input for receiving a sequence of frames and a processing unit 12 for generating superpixels for the received frames. A comparing unit 13 is configured to determine 1 a change of an area occupied by the superpixel between a current frame and a later frame of the sequence of frames. A terminating unit 14 is configured to terminate 2 the superpixel in the later frame in case the area of the superpixel becomes smaller than a specified first value. Also, a splitting unit 15 is configured to split 3 the superpixel in the later frame in case the area of the superpixel becomes larger than a specified second value. Preferably the apparatus 10 further comprises a determining unit 16, which is configured to determine 4 a similarity between an instance of the superpixel in a first frame and an instance of the superpixel in a later frame of the sequence of frames. A further comparing unit 17 is configured to compare 5 the determined similarity with a threshold. Also provided is a replacing unit 18, which is configured to replace 6 the superpixel in subsequent frames with a new temporally consistent superpixel in case the similarity is below the threshold. In any case the resulting superpixels are output via an output 19. Of course, two or more of the various units may likewise be combined into single unit.

Claims

1. A method for life cycle management of a superpixel associated to frames of a sequence of frames, the method **comprising** the steps of:
- terminating the superpixel in a frame in case an area of the superpixel becomes smaller than a specified first value; and
- splitting the superpixel in the frame in case the area of the superpixel becomes larger than a specified second value.

2. The method according to claim 1, **further** comprising the step of balancing a number of terminations of superpixels and splits of superpixels in order to keep a number of superpixels per frame essentially constant.

3. The method according to claim 1 or 2, **further** comprising the steps of:
- determining a similarity between an instance of the superpixel in a first frame and an instance of the superpixel in a later frame of the sequence of frames;
- comparing the determined similarity with a threshold; and
- replacing the superpixel in subsequent frames with a new temporally consistent superpixel in case the similarity is below the threshold.

4. An apparatus for life cycle management of a superpixel associated to frames of a sequence of frames, **characterized** in that the apparatus comprises:
- a terminating unit configured to terminate the superpixel in a frame in case an area of the superpixel becomes smaller than a specified first value; and
- a splitting unit configured to split the superpixel in the

frame in case the area of the superpixel becomes larger than a specified second value.

5. The apparatus according to claim 4, **further** comprising:
- a determining unit configured to determine a similarity between an instance of the superpixel in a first frame and an instance of the superpixel in a later frame of the sequence of frames;
- a comparing unit configured to compare the determined similarity with a threshold; and
- a replacing unit configured to replace the superpixel in subsequent frames with a new temporally consistent superpixel in case the similarity is below the threshold.

6. A computer readable storage medium having stored therein instructions enabling life cycle management of a superpixel associated to frames of a sequence of frames, which when executed by a computer, cause the computer to:
- terminate the superpixel in a frame in case an area of the superpixel becomes smaller than a specified first value; and
- split the superpixel in the frame in case the area of the superpixel becomes larger than a specified second value.

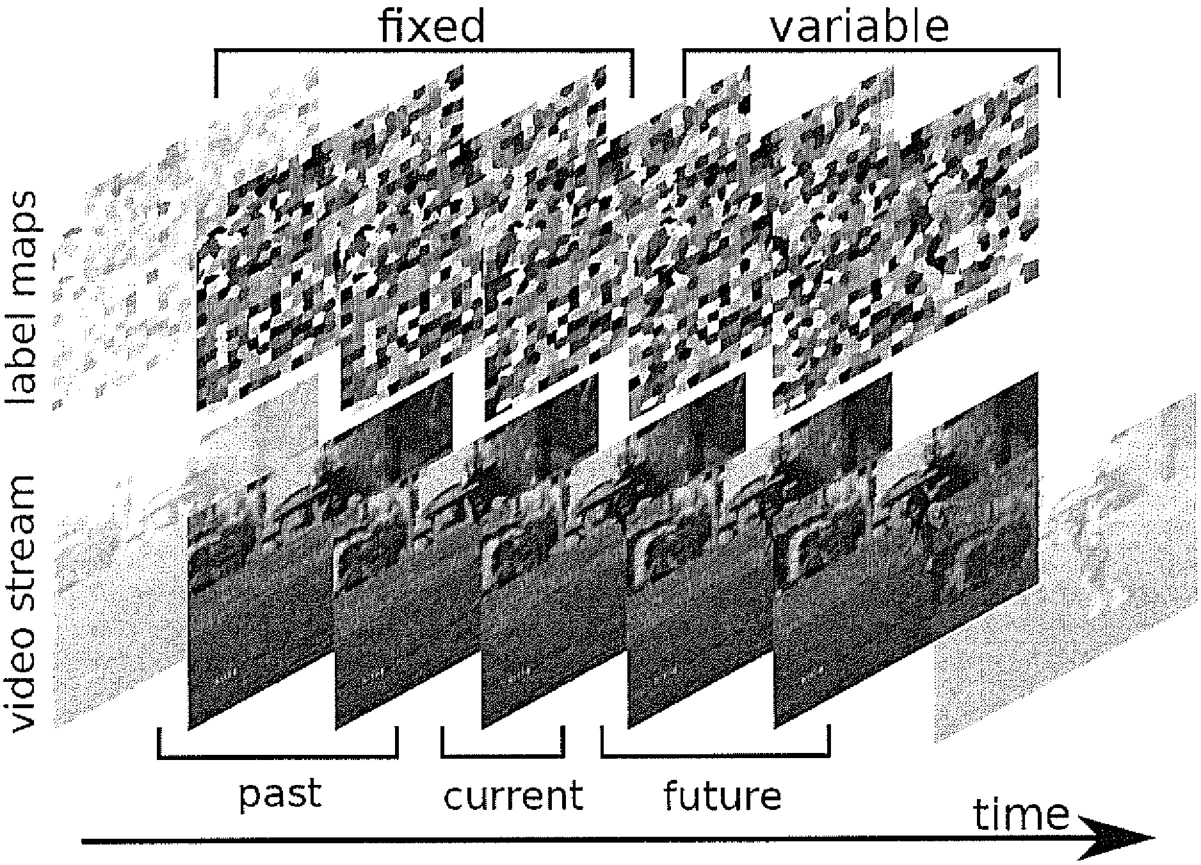
fixed
variable
label maps
video stream
past
current
future
time

Fig. 1

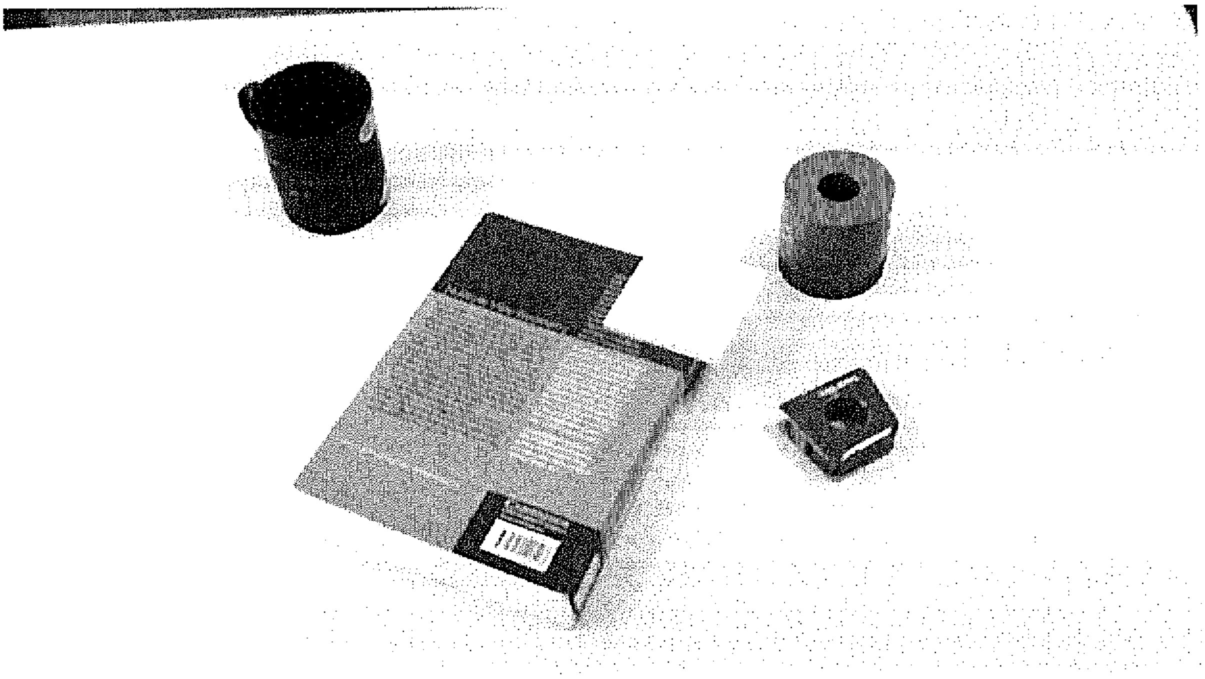

Fig. 2

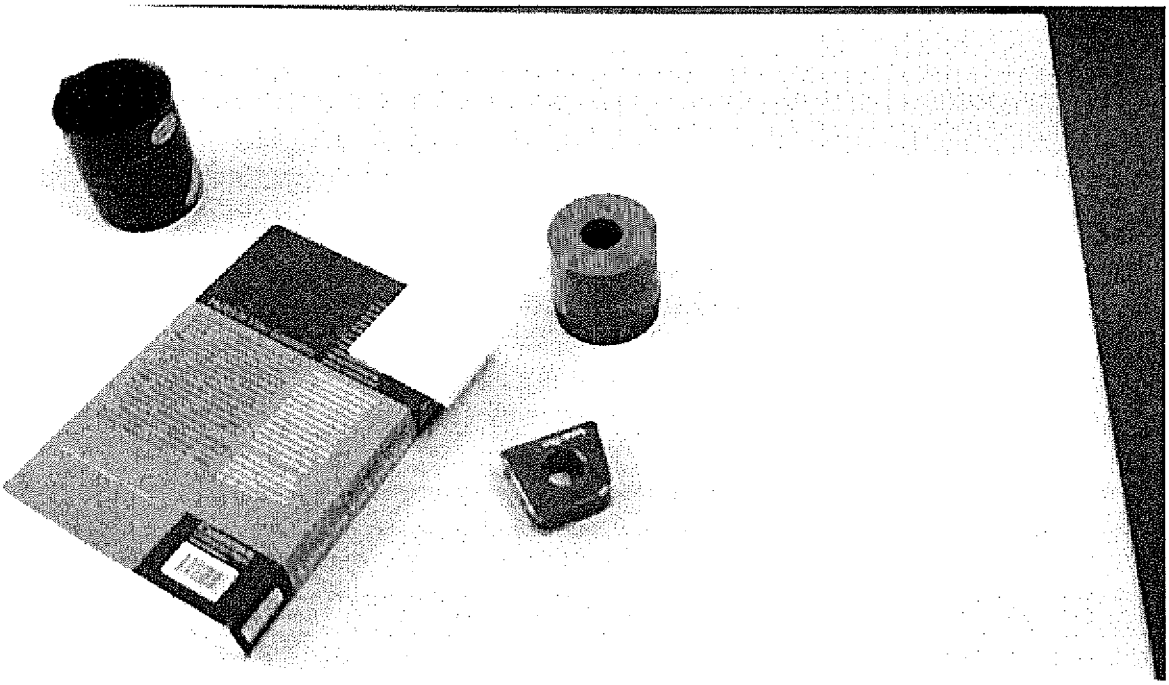

Fig. 3

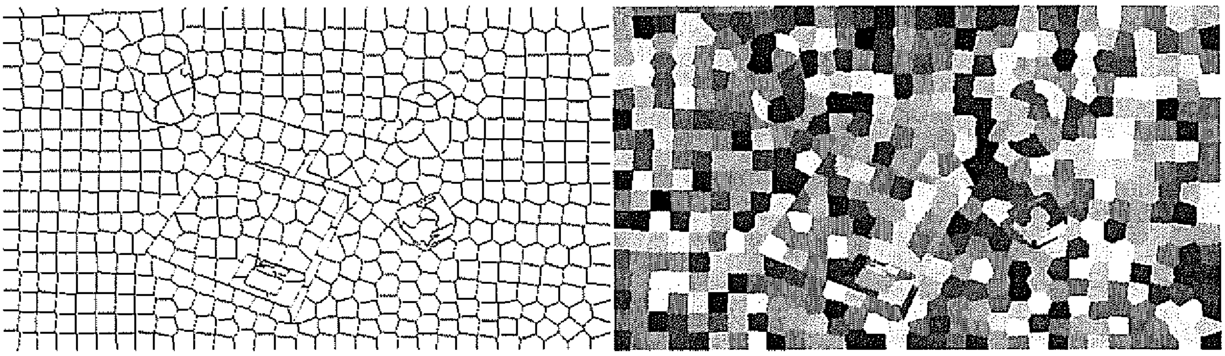

Fig. 4

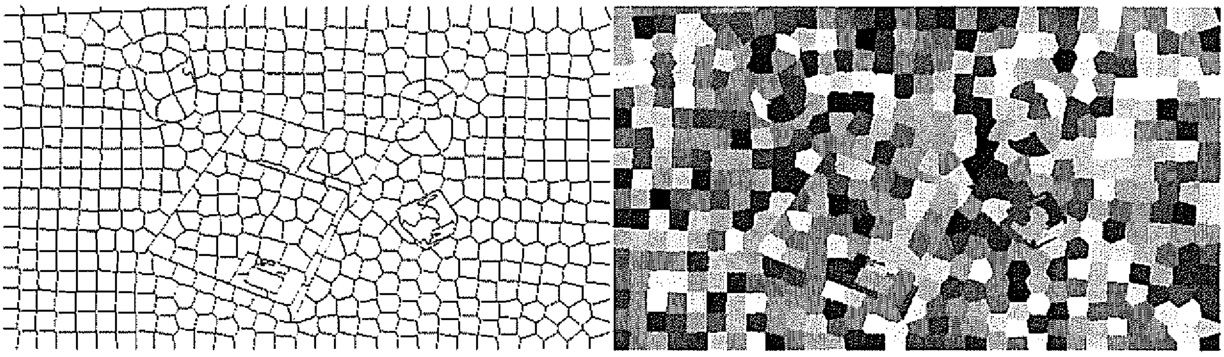

Fig. 5

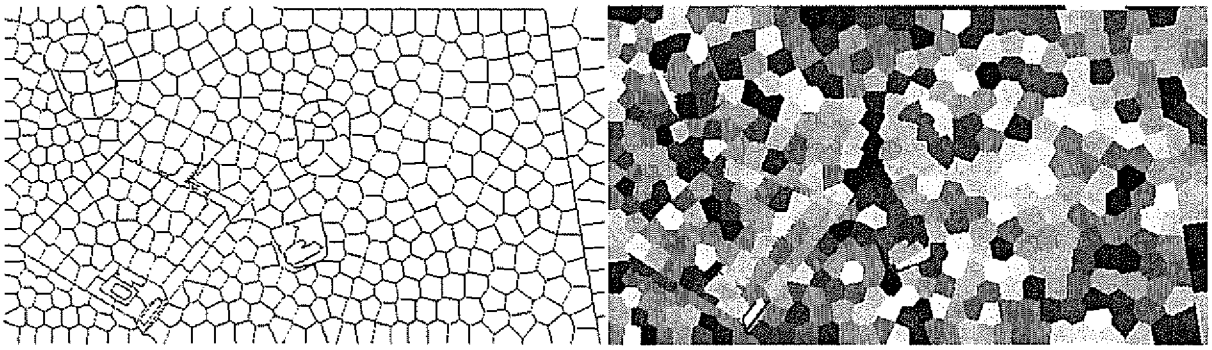

Fig. 6

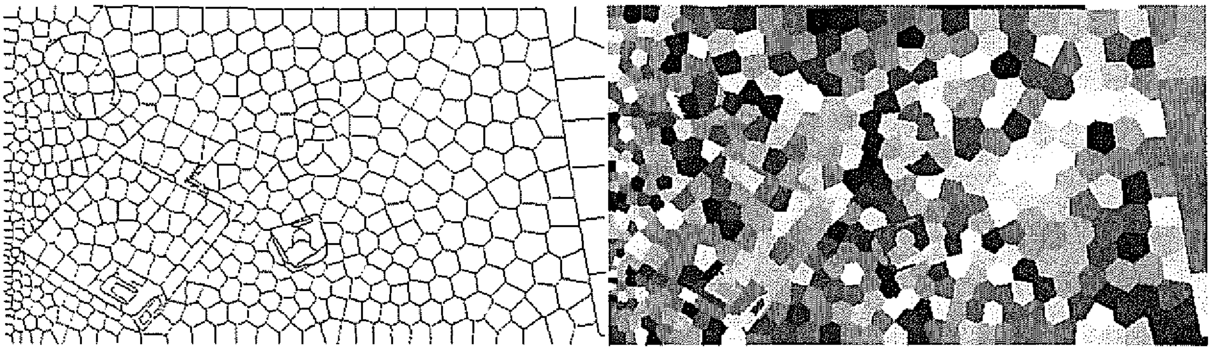

Fig. 7

2013248207 24 Oct 2013

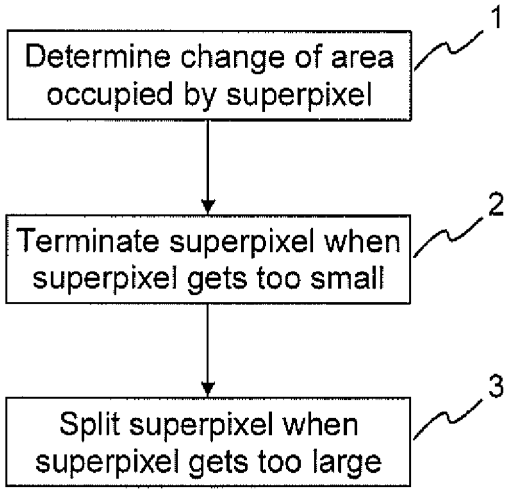
Determine change of area occupied by superpixel
1
Terminate superpixel when superpixel gets too small
2
Split superpixel when superpixel gets too large
3

Fig. 8

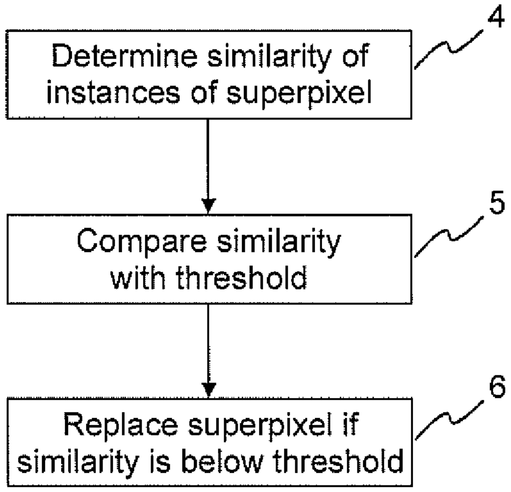
Determine similarity of instances of superpixel
4
Compare similarity with threshold
5
Replace superpixel if similarity is below threshold
6

Fig. 9

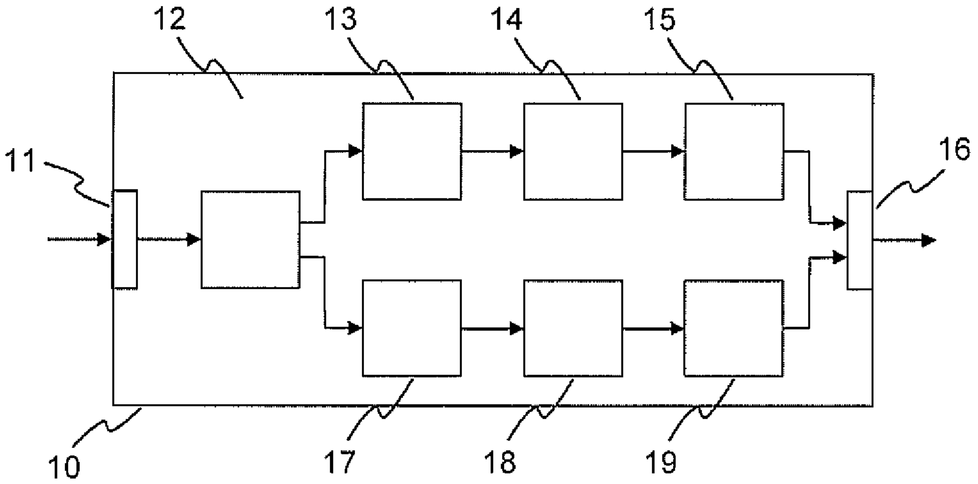

Fig. 10