Lilian Liu
Editor-in-Chief
Sensors

March 8, 2016

Dear Dr. Liu,

Re: Proof-reading for Manuscript reference No. sensors-113427

Please find attached the proofed version of our manuscript “**Fast Contour-Tracing Algorithm based on Pixel-Following Method for Image Sensors**”.

First of all, we would like to thank the editorial board for the acceptance of our manuscript. We attached a description of changes in the following pages. Please confirm the changes.

Thank you very much for your kind responses.

Yours sincerely,

**Jonghoon Seo**
Software Platform R&D Lab., LG Electronics Advanced Research Institute,
19 Yangjae-daero 11 gil, Seocho-gu, Seoul 137-893, Korea
E-Mail: jonghoon.seo@lge.com

## Description of changes

### 1. Modified the title

```
@@ -130,8 +130,8 @@

 %=================================================================

 % Full title of the paper (Capitalized)

-\Title{Fast Contour-Tracing Algorithm Based on a%please note the correction to the title

- Pixel-Following Method for Image Sensors}

+\Title{Fast Contour-Tracing Algorithm Based on \\

+a Pixel-Following Method for Image Sensors}
```

### 2. Provided (the renewed) postcode.

```
@@ -139,16 +139,16 @@

 % Affiliations / Addresses (Add [1] after \address if there is only one affiliation.)

 \address{%

-        $^{1}$ \quad Software Platform R\&D Lab., LG Electronics Advanced Research Institute, 19 Yangjae-daero 11 gil, Seocho-gu, Seoul 137-893, Korea;

+        $^{1}$ \quad Software Platform R\&D Lab., LG Electronics Advanced Research Institute, 19 Yangjae-daero 11-gil, Seocho-gu, Seoul, 06772, Korea;

  jonghoon.seo@lge.com \\

-        $^{2}$ \quad Department of Computer Science, Yonsei University, 134 Sinchon-dong Seodaemun-gu, \hl{Seoul postcode}, Korea;

+        $^{2}$ \quad Department of Computer Science, Yonsei University, 50 Yonsei-ro Seodaemun-gu, Seoul, 03722, Korea;
```

### 3. Modify corresponding author's e-mail address.

```
@@ -151,1 +151,1 @@

-\corres{Correspondence: hantack55@msl.yonsei.ac.kr; Tel.: +82-2-2123-2715; Fax: +82-2-365-2579}

+\corres{Correspondence: hantack55@gmail.com; Tel.: +82-2-2123-2715; Fax: +82-2-365-2579}
```

### 3. Preservation parenthesis and Arabian number format for the first paragraph at chapter 2.1.3.

The parenthesis are used to indicate the groups – they represent five type groups. So, it is suitable to use. Also, the Arabian numbers indicate the contour number at Figure 1c. So, it is better to use Arabian format.

## 4. Removed the part that was referenced in wrong order.

```
@@ -258, 1 +258,1 @@

-Table \ref{table:relatedworks} lists the characteristics of the contour-following methods. The pixel-following method and vertex-following method trace contours without scanning all of the pixels of the image, and their transition data, such as contour points and the tracing sequence, are generated automatically by the contour-following process. Therefore, only a few pixels need to be scanned in order to obtain the first contour pixel representing the starting point of the object. Despite these merits, they are not suitable for large images with many objects because they require more operations and a larger memory when compared to the run-data-based following method. In other words, they scan all of the pixels with an image-buffer size memory in order to obtain all of the objects, and they have several additional operations that enable them to detect whether to follow a contour pixel for all of the contour pixels and their adjacent background pixels (see \hl{Figure} \ref{fig:image8}).

+Table \ref{table:relatedworks} lists the characteristics of the contour-following methods. The pixel-following method and vertex-following method trace contours without scanning all of the pixels of the image, and their transition data, such as contour points and the tracing sequence, are generated automatically by the contour-following process. Therefore, only a few pixels need to be scanned in order to obtain the first contour pixel representing the starting point of the object. Despite these merits, they are not suitable for large images with many objects because they require more operations and a larger memory when compared to the run-data-based following method. In other words, they scan all of the pixels with an image-buffer size memory in order to obtain all of the objects, and they have several additional operations that enable them to detect whether to follow a contour pixel for all of the contour pixels and their adjacent background pixels.
```

## 5. Make italic format for several math notations.

```
@@ -293,2 +293,2 @@

-Let I

+Let $I$

  be a binary digital image with $M \times N$ pixels, where the coordinate of the top-leftmost pixel is $(0, 0)$ and that of the bottom-rightmost pixel is $(M - 1, N - 1)$. In $I$, a pixel can be represented as $P = (x, y),\, x = 0,1,2,\cdots,M-1,\, y = 0,1,2,\cdots, N-1$. Most contour-tracing algorithms use a tracer $T (P, d)$ with absolute directional information $d\in\{N,NE,NW,W,SW,S,SE,E,NE\}$, and they have the following basic sequence:
```

```
@@ -1155,2 +1155,2 @@

-The proposed algorithm does not save all of the contour pixels, but it saves only the representative points and the inner-outer corner pixels. Table \ref{table:table10} shows the data size acquired from the above experiments performed using CCITT standard fax images. It shows the data sizes of traced contour pixels and their compressed data. The number of traced contour pixels (A), which are the same results from Table \ref{table:table7}, and $C$ and $D$ in the table indicate the number of representative points and inner-outer corner points of the traced contour pixels. $A$ and $C$ are the number of $(x, y)$ coordinates, and $D$ represents the number of inner-outer corners that comprise $(x, y)$ coordinates and the type of inner-outer corner. The benefit of storing only the representative points based on the vertex of the contour pixel is that it can significantly reduce the data size. The experimental results obtained show that the proposed algorithm reduced the data size to 19\%--60\% of the memory used when all of the contour pixels were stored, as shown in Table \ref{table:table10}.

+The proposed algorithm does not save all of the contour pixels, but it saves only the representative points and the inner-outer corner pixels. Table \ref{table:table10} shows the data size acquired from the above experiments performed using CCITT standard fax images. It shows the data sizes of traced contour pixels and their compressed data. The number of traced contour pixels ($A$), which are the same results from Table \ref{table:table7}, and $C$ and $D$ in the table indicate the number of representative points and inner-outer corner points of the traced contour pixels. $A$ and $C$ are the number of $(x, y)$ coordinates, and $D$
```

represents the number of inner-outer corners that comprise $(x, y)$ coordinates and the type of inner-outer corner. The benefit of storing only the representative points based on the vertex of the contour pixel is that it can significantly reduce the data size. The experimental results obtained show that the proposed algorithm reduced the data size to 19\%--60\% of the memory used when all of the contour pixels were stored, as shown in Table \ref{table:table10}.

## 6. Insert reference numbers within the main text.

@@ -333,1 +333,1 @@

-The simple boundary follower (SBF) is also known as Papert's turtle algorithm \cite{Papert1973Uses} and as a square-tracing algorithm \cite{Ghuneim2015Contour}, and it is the simplest contour-tracing algorithm. Initially, the location of tracer $S$ is saved, and the tracer moves in a left or right direction. If the pixel tracer is located on a contour pixel, the tracer moves left; otherwise, it moves right. The procedure is as given \hl{below}.

+The simple boundary follower (SBF) is also known as Papert's turtle algorithm \cite{Papert1973Uses} and as a square-tracing algorithm \cite{Ghuneim2015Contour}, and it is the simplest contour-tracing algorithm. Initially, the location of tracer $S$ is saved, and the tracer moves in a left or right direction. If the pixel tracer is located on a contour pixel, the tracer moves left; otherwise, it moves right. The procedure is shown in Algorithm \ref{alg:sbf}.

@@ -373,1 +373,1 @@

-The SBF and MSBF require movement operations for both contour and background pixels; therefore, time is wasted during movement on the background pixel, and they cannot trace the inner-corner pixel in front of the tracer \cite{Cheong2006Improved,Ghuneim2015Contour}. Hence, we have proposed an improved SBF (ISBF)~\cite{Cheong2006Improved} that is based on our previous research aimed at overcoming these limitations. The ISBF has six cases for following contour pixels based on the local patterns of the contour pixels. The modified version of \cite{Cheong2012Advanced} is as \hl{follows}:

+The SBF and MSBF require movement operations for both contour and background pixels; therefore, time is wasted during movement on the background pixel, and they cannot trace the inner-corner pixel in front of the tracer \cite{Cheong2006Improved,Ghuneim2015Contour}. Hence, we have proposed an improved SBF (ISBF)~\cite{Cheong2006Improved} that is based on our previous research aimed at overcoming these limitations. The ISBF has six cases for following contour pixels based on the local patterns of the contour pixels. The modified version\cite{Cheong2006Advanced} of ISBF is shown in Algorithm \ref{alg:isbf}.

## 7. Inserted the image reproduction citation for Figure 4.

@@ -430,1 +430,1 @@

- \caption{Contour cases of the improved SBF (ISBF) \cite{Cheong2012Advanced}: (\textbf{a}) left neighbor; (\textbf{b}) inner-outer corner at the left-rear; (\textbf{c}) inner-outer corner at the front-left; (\textbf{d}) inner corner at the front; (\textbf{e}) front neighbor; (\textbf{f}) outer corner.}

+ \caption{Contour cases of the improved SBF (ISBF) \cite{Cheong2006Advanced}: (\textbf{a}) left neighbor; (\textbf{b}) inner-outer corner at the left-rear; (\textbf{c}) inner-outer corner at the front-left; (\textbf{d}) inner corner at the front; (\textbf{e}) front neighbor; (\textbf{f}) outer corner. Reproduced with permission from Cheong, C., Seo, J., and Han, T.D., in proceedings of the 33rd KISS conference; published by KIISE, 2006.}

## 8. Removed \hl tags for figure captions.

@@ -457,1 +457,1 @@

- \caption{Contour-following sequence of Moore-neighbor tracing (MNT) and the radial sweep algorithm (RSA): (\textbf{a}) \hl{MNT} \cite{Toussaint????Grids}; (\textbf{b}) RSA~\cite{Mirante1982Radial}.}

+ \caption{Contour-following sequence of Moore-neighbor tracing (MNT) and the radial sweep algorithm (RSA): (\textbf{a}) MNT \cite{Toussaint????Grids}; (\textbf{b}) RSA~\cite{Reddy2012Evaluation}.}

## 9. Define CCITT acronym

@@ -935,1 +935,1 @@

-We experimented on nine \hl{CCITT}

- standard fax images with 200 dots per inch (dpi) \cite{Miyatake1997Contour}. All of these images have $1728 \times 2339$ pixels and a file size of 11,842~KB. Table \ref{table:ccitt} shows the document type of these images and the total number of contour pixels. We used these large-sized images because they have various types of contours, which is useful when comparing the efficiencies with regard to parameters, such as processing time and the accuracy of the trace results of the contour-tracing~algorithms.

+We experimented on nine CCITT (Consultative Committee for International Telephony and Telegraphy) standard fax images with 200 dots per inch (dpi) \cite{Miyatake1997Contour}. All of these images have $1728 \times 2339$ pixels and a file size of 11,842~KB. Table \ref{table:ccitt} shows the document type of these images and the total number of contour pixels. We used these large-sized images because they have various types of contours, which is useful when comparing the efficiencies with regard to parameters, such as processing time and the accuracy of the trace results of the contour-tracing~algorithms.

@@ -954,1 +954,1 @@

- \caption{\protect \hl{CCITT} fax standard images.}

+ \caption{\protect CCITT (Consultative Committee for International Telephony and Telegraphy) fax standard images.}

## 10. Remove the italics for OpenCV and its function.

@@ -1047,1 +1047,1 @@

-In order to measure the tracing time for each algorithm, we performed each algorithm 20 times per image and calculated the average time. We used the \hl{\textit{cv2.getTickCount()}} function supported by \hl{\textit{OpenCV~3.0.0}} to measure the processing time. Table \ref{table:table8} shows the average processing time of each algorithm used for tracing the images and a linear model for estimating the process time as the number of traced pixels increases using the least-square estimation (LSE) method. In the table, we obtain the average processing time per traced contour pixel by dividing the total processing time by the total number of traced contour pixels.

+In order to measure the tracing time for each algorithm, we performed each algorithm 20 times per image and calculated the average time. We used the cv2.getTickCount() function supported by OpenCV 3.0.0 to measure the processing time. Table \ref{table:table8} shows the average processing time of each algorithm used for tracing the images and a linear model for estimating the process time as the number of traced pixels increases using the least-square estimation (LSE) method. In the table, we obtain the average processing time per traced contour pixel by dividing the total processing time by the total number of traced contour pixels.

## 11. Remove * at the bottom of Table 8.

@@ -1137,10 +1130,10 @@

-\multicolumn{1}{c}{\footnotesize \hl{*}~Results of best or faster speed/smaller standard deviation than the proposed algorithm are marked with a shadow.}

+\multicolumn{1}{c}{\footnotesize Results of best or faster speed/smaller standard deviation than the proposed algorithm are marked with a shadow.}

\end{tabular}

-\end{table}%Please add * in table.

+\end{table}

## 12. Remove (A), because it is not described at figure.

@@ -1309,1 +1302,1 @@

-To overcome the problem, we applied an eight-connection mask to the images to obtain the starting pixel, but the mask required many operations. In other words, we attempted to measure the performance of multi-direction scanning in order to eliminate the missing contour-pixel problem by using vertical and horizontal scans instead of an eight-connection mask operation. Table \ref{table:table12} shows the increase in the number of pixels traced using bidirectional scanning, and Table \ref{table:table13} describes the processing time for this method. Moreover, Figure \ref{fig:image22} shows the tracing result that was obtained using the proposed algorithm based on bidirectional scanning, and it shows that seven of the missing pixels are traced, but one diagonal connective-contour pixel \hl{(A)} remained untraced.

+To overcome the problem, we applied an eight-connection mask to the images to obtain the starting pixel, but the mask required many operations. In other words, we attempted to measure the performance of multi-direction scanning in order to eliminate the missing contour-pixel problem by using vertical and horizontal scans instead of an eight-connection mask operation. Table \ref{table:table12} shows the increase in the number of pixels traced using bidirectional scanning, and Table \ref{table:table13} describes the processing time for this method. Moreover, Figure \ref{fig:image22} shows the tracing result that was obtained using the proposed algorithm based on bidirectional scanning, and it shows that seven of the missing pixels are traced, but one diagonal connective-contour pixel remained untraced.

## 13. Provide cities and countries for references.

@@ -1494,1 +1494,1 @@

-\newblock {\em Digital Image Processing}; \hl{Wiley: City, Country}, 1978.

-%Please add the publisher's location (city, country).

+\newblock {\em Digital Image Processing}; Wiley \& Sons: Hoboken, NJ, USA., 1978.

@@ -1508,8 +1500,7 @@ Gose, E.; Johnsonbaugh, R.; Jost, S.

\bibitem[Pitas(2000)]{Pitas2000Digital}

Pitas, I.

-\newblock {\em Digital Image Processing Algorithms and Applications}; \hl{John

- Wiley \& Sons: City, Country}, 2000.

+\newblock {\em Digital Image Processing Algorithms and Applications}; Wiley \& Sons: Hoboken, NJ, USA., 2000.

@@ -1524,1 +1524,1 @@

-\newblock {\em Algorithms for Graphics and Image Processing}; \hl{Springer Science

- \& Business Media: City, Country}, 2012.

+\newblock {\em Algorithms for Graphics and Image Processing}; Springer-Verlag: Berlin-Heidelberg-New York, 2012.

## 14. Provide accessed date for references.

@@ -1514,1 +1514,1 @@

\bibitem[Papert(1973)]{Papert1973Uses}

Papert, S.

-\newblock Uses of Technology to Enhance Education. Available online: http://hdl.handle.net/1721.1/6213 (accessed on \hl{Day Month Year}).

+\newblock Uses of Technology to Enhance Education. Available online: http://hdl.handle.net/1721.1/6213 (accessed on 16 August 2015).

@@ -1600,9 +1585,10 @@ Shoji, K.; Miyamichi, J.; Hirano, K.

\newblock {\em Syst. Comput. Jpn.} {\bf 1999}, {\em 30},~1--11.

\bibitem[Toussaint()]{Toussaint????Grids}

-\hl{Toussaint, G.}

-\newblock Grids, Connectivity and Contour Tracing. \hl{ the book name it belongs to; Publisher: City Country, Year}.

-%Please check this reference type.

+Toussaint, G.

+\newblock Grids, Connectivity and Contour Tracing.

+\newblock

+ Available online: \href{http://www-cgrl.cs.mcgill.ca/~godfried/teaching/pr-notes/contour.ps}{\nolinkurl{http://www-cgrl.cs.mcgill.ca/~godfried/teaching/pr-notes/contour.ps}}. (accessed on 26 August 2015).

```
@@ -1617,8 +1603,7 @@ Ghuneim, A.

 \newblock

  Available online:
\href{http://www.imageprocessingplace.com/downloads\_V3/root\_downloads/tutorials/contour\_tracing\_Abeer\_George\_Ghuneim/index.html}


{\nolinkurl{http://www.imageprocessingplace.com/downloads\_V3/root\_downloads/tutorials/contour\_tracing\_Abeer\_George\_Ghuneim/index.html}}.

-(\hl{accessed on}).

-%Please add the online resource update day month year.

+(accessed on 28 August 2015).
```

## 15. Remove reference [8] and change referring to [23].

Change from \cite{Mirante1982Radial} to \cite{Reddy2012Evaluation} over whole document.

## 16. Fix [15]'s year from 2012 to 2006.

```
@@ -1573,11 +1558,11 @@ Zhang, S.; Zhao, X.; Lei, B.

 \newblock Robust Facial Expression Recognition via Compressive Sensing.

 \newblock {\em Sensors} {\bf 2012}, {\em 12},~3747--3761.



-\bibitem[Cheong \em{et~al.}(2012)Cheong, Seo, and Han]{Cheong2012Advanced}

+\bibitem[Cheong \em{et~al.}(2006)Cheong, Seo, and Han]{Cheong2006Advanced}

 Cheong, C.; Seo, J.; Han, T.D.

 \newblock Advanced contour tracing algorithms based on analysis of tracing

  conditions.

-\newblock In Proceedings of the 33rd KISS Fall Conference, \hl{ City, Country, start date--end date 2012}; Volume

+\newblock In Proceedings of the 33rd KISS Fall Conference, Seoul, Korea, 20--21 October 2006; Volume

  33, pp. 431--436.
```