

UNIVERSIDAD AUTÓNOMA DE MADRID
ESCUELA POLITÉCNICA SUPERIOR



SELECCIÓN AUTOMÁTICA DE VISTA EN SISTEMAS MULTICÁMARA

-PROYECTO FIN DE CARRERA-

Javier Díez Salcedo
Abril 2012

SELECCIÓN AUTOMÁTICA DE VISTA EN SISTEMAS MULTICÁMARA

Autor: Javier Díez Salcedo

Supervisor: José María Martínez Sánchez

email: Javier.diezs@estudiante.uam.es, JoseM.Martinez@uam.es



Video Processing and Understanding Lab
Departamento de Tecnología Electrónica y de las Comunicaciones
Escuela Politécnica Superior
Universidad Autónoma de Madrid
Abril 2012

Abstract

The objective of this project has been the implementation of a complete system which function is to evaluate and mount the content of several cameras. For this purpose a collection of algorithms has been developed: they extract the different features from the cameras to evaluate them in order to decide which camera is the most important in every moment. Finally with these decisions the system proceeds to mount the video output.

The developed algorithms evaluate the content of every frame, the quality of the image and if there are others that simulate the attention of a hypothetical user avoiding probable boredom. Everyone of these algorithms returns collected data, which is used by another algorithm to assign a score to the whole frame. When every frame has its own score an algorithm decides which camera is more important in every instant; with this info the editing of the video output is done. There are two different mountings of the video output, the first one has only the output, but the second one shows the output and the view of every camera.

In order to analyze the utility of the whole system and that everything works as it should be there are two tests, firstly every algorithm is analyzed individually using images, because in the system they will work with single frames. Secondly, the whole system is tested using for this task three datasets, the content will be different: football, basketball and tennis.

Owing to there are many possibilities to mount the multiple videos, a great number of tests has been done, changing all the possible parameters in order to get the best results.

Resumen

El objetivo de este proyecto ha sido la implementación de un sistema completo de evaluación y montaje del contenido de múltiples cámaras. Para ello se han desarrollado un conjunto de algoritmos, encargados de extraer las distintas características de la cámaras, para después poder evaluar cual de esas cámaras tiene el contenido más importante en cada momento y finalmente, hacer el montaje correspondiente a la información obtenida.

Cuenta con algoritmos que evalúan desde el contenido de cada fotograma, a la calidad de la imagen o incluso algoritmos que simulan la atención del usuario a cada una de las cámaras, evitando así el aburrimiento del espectador. Todos estos algoritmos devolverán los datos recogidos, los cuales serán usados para a través de otro algoritmo dar una puntuación a cada uno de los fotogramas. Estas puntuaciones serán analizadas y otro algoritmo será el encargado de elegir que cámara será mostrada en cada momento. Finalmente se procederá a hacer el montaje, el cuál se verá de dos maneras; la primera será únicamente el montaje, y la segunda será una interfaz en la que podremos ver tanto la salida como lo que está mostrando cada una de las cámaras.

Para evaluar la utilidad de toda la implementación, se ha trabajado en dos partes, primeramente los algoritmos han sido evaluados individualmente mediante el uso de imágenes, ya que luego actuarán fotograma a fotograma. Y finalmente se ha puesto a prueba todo el sistema, realizando el montaje con tres distintos conjuntos de vídeos: fútbol, baloncesto y tenis.

Debido a las grandes posibilidades de montaje que se pueden realizar, ya sean variaciones de cámara

más rápidas o lentas, evaluación en paralelo o con cierto retraso, etc. se han hecho un gran número de pruebas con cada uno de los dataset.

Keywords

Algorithms, Blur, Frames, Blobs, Multicamera, Content scoring, Feature analysis, Camera selection, Autonomous video production, Attention coefficient.

Palabras clave

Algoritmos, Emborronamiento, Fotogramas, Blobs, Multicámara, Puntuación de contenido, Análisis de características, Elección de cámara, Montaje de vídeo, Coeficiente de atención

Agradecimientos

Es difícil agradecer a todas las personas que durante este tiempo han contribuido a llegar a este día, y no porque me cueste dar las gracias, sino porque han sido muchas.

Me gustaría agradecer a los que desde el laboratorio me han ayudado con este pfc, a Juan Carlos porque me atendió siempre que tenía dudas, porque cuando mi proyecto se vio amenazado por la repentina muerte del ordenador, se preocupó por ayudarme a recuperar todos los datos. A Marcos, porque estuvo ahí para atender mis dudas. Y por supuesto a mi tutor, Chema. Gracias.

También quiero aprovechar esta oportunidad para agradecer a todos aquellos compañeros que se han convertido en amigos a lo largo de la carrera, a los que se sentaban al lado sufriendo horas de clases, a los que usaban y seguro que siguen usando pijama de seda, a los que buscaban treses en sudokus, a los que descubrí en el último año, a los que siempre ponían la foto picante en los mails, a los que prestaban apuntes, a todos. Gracias.

A mi amiga y fiel compañera de prácticas y mesa. Que con su afán de aprobar las mismas que yo siempre estaba ahí, hiciera lo que hiciera. Porque me has hecho la carrera mucho más llevadera y divertida. Porque hemos pasado muchas cosas y al final parece que no hay hoyito. Gracias.

A ti Luis por estar siempre ahí aun cuando ya no estas. Gracias.

A mi mejor amigo, mi leal compañero de biblioteca que obligándome a ir a la biblioteca me hizo suspender alguna en segundo pero que finalmente me ayudó a aprobar todas las demás. Me alegro de haber disfrutado contigo parte de la vida en la universidad y espero que esto solo sea una etapa más de las muchas que compartamos. Gracias.

A ti, que has sido entregada correctora de todo este trabajo, que me has ayudado a cada paso, que me has animado cuando no veía el fin. Aunque llegaste tarde a esta etapa te has convertido en uno de los pilares. Gracias.

A mis padres, gracias por jamás presionarme, por darme todas las facilidades para poder desarrollarme como persona, por estar siempre ahí, por todo. Gracias.

A mi hermana, por no hacerme salir de casa para tener que encontrar a una de mis mejores amigas. Gracias.

Todos habéis sido mi gran apoyo para poder acabar esta carrera, y por eso he querido dedicaros estas líneas a vosotros.

¡GRACIAS!

Índice general

1. Introducción	1
1.1. Motivación	1
1.2. Objetivos	1
1.3. Organización de la memoria	2
2. Estado del arte	3
2.1. Introducción	3
2.2. Análisis del contenido y extracción de las características del vídeo	3
2.2.1. Distintas características	3
2.2.2. Detección de eventos	5
2.3. Puntuación del contenido	6
2.4. Selección de vista	7
3. Diseño, desarrollo e integración	9
3.1. Introducción	9
3.2. Arquitectura del sistema	9
3.2.1. Resize	9
3.2.2. AreaNBlobs	11
3.2.3. QualityBlur	15
3.2.4. userAttention	16
3.2.5. Score	21
3.2.6. Decisor	22
3.2.7. Presentation	23
3.3. Detalles de implementación	26
4. Resultados	29
4.1. Introducción	29
4.2. Elección de parámetros y sus consecuencias	29
4.2.1. Baloncesto (SET 1)	30
4.2.2. Baloncesto (SET 2)	35
4.2.3. Baloncesto (SET 3)	39
4.2.4. Baloncesto (SET 4)	44

4.2.5. Baloncesto (SET 5)	48
4.2.6. Baloncesto (SET 6)	52
4.2.7. Conclusiones finales sobre los parámetros	56
4.3. Resultados para distintos dataset	56
4.3.1. Baloncesto	56
4.3.2. Fútbol	57
4.3.3. Tenis	62
4.4. Aportación de QualityBlur y userAttention	67
4.4.1. QualityBlur	67
4.4.2. userAttention	67
4.5. Respuesta en tiempo del sistema	70
5. Conclusiones y trabajo futuro	73
5.1. Conclusiones	73
5.2. Trabajo futuro	74
5.2.1. Parametrizable para cualquier número de cámaras	74
5.2.2. Estudio interpolación	74
5.2.3. Función de Score	74
5.2.4. Algoritmo de orden de cámaras	74
5.2.5. Reducción tiempo de ejecución	74
5.2.6. Detección de objetos	75
5.2.7. Detección de eventos	75
5.2.8. Sonido	75
5.2.9. Análisis subjetivo de funcionamiento del sistema.	75
Bibliografía	77
Appendix.	81
A. Parametrización	81
A.1. Aplicación a imágenes	81
A.1.1. Dataset de 6 cámaras (Fútbol)	85
A.1.2. Dataset de 7 cámaras (Baloncesto)	85
A.1.3. Dataset de 9 cámaras (Tenis)	85
B. Gráficas de resultados intermedios	91
B.1. Introducción	91
B.2. Distinta elección de los parámetros	92
B.2.1. Baloncesto (SET1)	92
B.2.2. Baloncesto (SET2)	98
B.2.3. Baloncesto (SET3)	102
B.2.4. Baloncesto (SET4)	106
B.2.5. Baloncesto (SET5)	109

B.2.6. Baloncesto (SET6)	112
B.3. Resultados para distintos dataset	115
B.3.1. Fútbol	115
B.3.2. Tenis	121
C. Presupuesto	127
D. Pliego de condiciones	129

Índice de figuras

2.1. Aprendizaje de background	4
2.2. Detección de movimiento (foreground)	4
2.3. Clasificación según la orientación	4
2.4. Red Bayesiana	5
2.5. Evento: Intento de canasta	6
2.6. Puntuación de objeto	6
2.7. Grafo DBN	7
3.1. Diagrama de bloques	10
3.2. Blobs según altura y anchura mínima	13
3.3. Problemas con los blobs	14
3.4. blob por persona	15
3.5. Coeficiente de atención del usuario.	17
3.6. Logarítmica	18
3.7. Logarítmica con parte plana	19
3.8. Arcotangente	19
3.9. Arcotangente limitada	20
3.10. Arcotangente con reinicialización	21
3.11. Imagen comparativa <i>AntiFlickering</i> vs. <i>Retardo</i>	24
3.12. Presentación gráfica	26
3.13. Petición de parámetros	27
3.14. Archivos de texto	28
4.1. Presentación de baloncesto	30
4.2. Histograma del uso de cámaras.	32
4.3. Problemas blobs y baloncesto	32
4.4. Puntuación acumulada (vídeo completo)	33
4.5. Puntuación acumulada (parcial)	34
4.6. Histograma del uso de las cámaras. SET 2 vs SET 1	36
4.7. Puntuación acumulada (vídeo completo). SET 2 vs SET 1	37
4.8. Puntuación acumulada (parcial). SET 2 vs SET 1	38
4.9. Histograma del uso de las cámaras. SET 3 vs SET 1	40

4.10. Puntuación acumulada (vídeo completo). SET 3 vs SET 1	42
4.11. Puntuación acumulada (parcial). SET 3 vs SET 1	43
4.12. Histograma del uso de las cámaras. SET 4 vs SET 1	45
4.13. Puntuación acumulada (vídeo completo). SET 4 vs SET 1	46
4.14. Puntuación acumulada (parcial). SET 4 vs SET 1	47
4.15. Histograma del uso de las cámaras. SET 5 vs SET 1	49
4.16. Puntuación acumulada (vídeo completo). SET 5 vs SET 1	50
4.17. Puntuación acumulada (parcial). SET 5 vs SET 1	51
4.18. Histograma del uso de las cámaras. SET 6 vs SET 1	53
4.19. Puntuación acumulada (vídeo completo). SET 6 vs SET 1	54
4.20. Puntuación de las cámaras (parcial). SET 6 vs SET 1	55
4.21. Presentación de fútbol	57
4.22. Histograma del uso de cámaras	58
4.23. Puntuación acumulada (vídeo completo)	60
4.24. Puntuación acumulada (parcial)	61
4.25. Presentación de tenis	62
4.26. Histograma del uso de cámaras	63
4.27. Puntuación acumulada (vídeo total)	65
4.28. Puntuación acumulada (parcial)	66
4.29. Distintas salidas. Dataset fútbol.	68
4.30. Distintas salidas. Dataset fútbol.	69
4.31. Tiempo empleado por cada bloque	71
4.32. Tiempo empleado por cada bloque (II)	72
A.1. Imagen referencia.	81
A.2. Prueba 1.	82
A.3. Prueba 2.	83
A.4. Dataset fútbol.	86
A.5. Dataset baloncesto.	87
A.6. Dataset Baloncesto (II)	88
A.7. Dataset tenis.	89
A.8. Dataset tenis (II)	90
B.1. Número de blobs	92
B.2. Coeficiente de área	93
B.3. Coeficiente de emborronamiento	94
B.4. Coeficiente de atención	95
B.5. Puntuación de las cámaras (vídeo completo)	96
B.6. Puntuación de las cámaras (parcial)	97
B.7. Número de blobs. SET 2 vs SET 1	98
B.8. Coeficiente de atención. SET 2 vs SET 1	99
B.9. Puntuación de las cámaras (vídeo completo). SET 2 vs SET 1	100

B.10.Puntuación de las cámaras (parcial). SET 2 vs SET 1	101
B.11.Número de blobs. SET 3 vs SET 1	102
B.12.Coeficiente de atención. SET 3 vs SET 1	103
B.13.Puntuación de las cámaras (vídeo completo). SET 3 vs SET 1	104
B.14.Puntuación de las cámaras (parcial). SET 3 vs SET 1	105
B.15.Coeficiente de atención. SET 4 vs SET 1	106
B.16.Puntuación de las cámaras (vídeo completo). SET 4 vs SET 1	107
B.17.Puntuación de las cámaras (parcial). SET 4 vs SET 1	108
B.18.Coeficiente de atención. SET 5 vs SET 1	109
B.19.Puntuación de las cámaras (vídeo completo). SET 5 vs SET 1	110
B.20.Puntuación de las cámaras (parcial). SET 5 vs SET 1	111
B.21.Coeficiente de atención. SET 6 vs SET 1	112
B.22.Puntuación de las cámaras (vídeo completo). SET 6 vs SET 1	113
B.23.Puntuación de las cámaras (parcial). SET 6 vs SET 1	114
B.24.Número de blobs	115
B.25.Coeficiente de área	116
B.26.Coeficiente de emborronamiento	117
B.27.Coeficiente de atención	118
B.28.Puntuación de las cámaras (vídeo completo)	119
B.29.Puntuación de las cámaras (parcial)	120
B.30.Número de blobs	121
B.31.Coeficiente de área	122
B.32.Coeficiente de emborronamiento	123
B.33.Coeficiente de atención	124
B.34.Puntuación de las cámaras (vídeo total)	125
B.35.Puntuación de las cámaras (parcial)	126

Índice de cuadros

4.1. Reinicio Coef. atención	39
4.2. Coeficientes emborronamiento.	67
4.3. Valores de tiempo medio por <i>frame</i> (ms)	70
4.4. Valores de tiempo empleado por los bloques en los 1499 <i>frames</i> del vídeo (ms)	70
A.1. Coeficiente de emborronamiento.	84
A.2. Coeficientes emborronamiento. Dataset fútbol.	88
A.3. Coeficientes emborronamiento. Dataset baloncesto.	88
A.4. Coeficientes emborronamiento. Dataset tenis.	90

Capítulo 1

Introducción

1.1. Motivación

La motivación principal de este proyecto es la realización de un sistema que sea capaz de analizar el contenido de distintos conjuntos de vídeos y poder discernir cuales son los puntos de vista más interesantes en cada momento para ir mostrándolos en ese orden.

1.2. Objetivos

El objetivo principal de este proyecto es la creación de un sistema completo, que consiga montar un único vídeo formado por las mejores vistas de un sistema multicámara.

Para ello deberá extraer información de cada una de las cámaras dadas como entrada y analizarla. Una vez hecho esto, deberá elegir cuál es la mejor vista en cada momento y ofrecerla como salida.

En un primer momento será necesario diseñar cada uno de los módulos que formarán parte del sistema:

- Adecuación de las posibles entradas al sistema.
- Extracción de información de cada una de las cámaras.
- Evaluación y comparación de los datos extraídos.
- Toma de decisión.
- Realización del montaje a partir de las decisiones tomadas.

Una vez que todos estos módulos han sido diseñados, se procederá a su desarrollo. Estos mismos bloques serán evaluados individualmente, si así lo requieren, para su calibración o la comprobación de su correcto funcionamiento.

Cuando se tengan los módulos en funcionamiento, se procederá a su ensamblaje como un todo. Después de lo cual el sistema estará completo y se pasará a su comprobación; para ello se preparará una batería de pruebas formada por diferentes dataset de vídeos.

De las pruebas se obtendrán conclusiones sobre su funcionamiento.

Si bien aunque no es un objetivo principal, también se obtendrán estadísticas sobre el tiempo de ejecución, con vistas a trabajo futuro.

1.3. Organización de la memoria

La memoria consta de los siguientes capítulos:

- **Capítulo 1:** Introducción.
- **Capítulo 2:** Estado del arte.
- **Capítulo 3:** Diseño, desarrollo e integración.
- **Capítulo 4:** Resultados.
- **Capítulo 5:** Conclusiones y trabajo futuro.

Y los siguientes anexos:

- **Anexo A:** Parametrización del algoritmo de calidad.
- **Anexo B:** Resultados intermedios.

Capítulo 2

Estado del arte

2.1. Introducción

A continuación se describen algunas de las soluciones existentes a problemas que se han de resolver para la realización del sistema objetivo de este proyecto.

Si bien quizás algunas de las explicaciones no conformen en sí mismas una solución completa para alguno de los problemas, sí que suponen un paso intermedio para su resolución.

Este capítulo se dividirá en tres secciones, en las que se seguirá el flujo del sistema a crear:

- Análisis del contenido y extracción de características del vídeo.
- Puntuación de las mismas.
- Selección de la vista.

2.2. Análisis del contenido y extracción de las características del vídeo

2.2.1. Distintas características

A la hora de seleccionar que características del contenido se observarán para tomar la decisión sobre que cámara será la elegida para ser mostrada, hay infinitas posibilidades. Dependerán en mayor medida del tipo de contenido del vídeo, y por supuesto, del uso que se vaya a hacer de él.

En [1], se extraen varias características, como la cantidad de movimiento, que será el número de píxeles en movimiento normalizados entre el tamaño de píxeles de la imagen; para ello se lleva a cabo una segmentación de background y foreground, que se realiza mediante un detector de cambio de color, [2].

Una vez detectado el movimiento, en [1], se lleva a cabo un análisis para ver si los objetos encontrados en el foreground son reales o ruido. Para ello se tendrán en cuenta parámetros como anchura, altura y orientación esperada. Según lo que se busque en el contenido, se dará más o menos importancia a cada



Figura 2.1: Aprendizaje de background

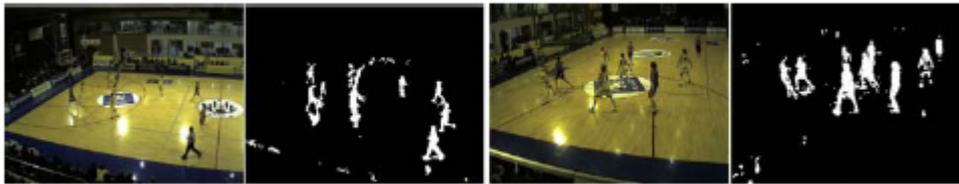


Figura 2.2: Detección de movimiento (foreground)

uno. Por ejemplo, en el caso de un vídeo cuyo contenido son personas, dado que las personas suelen estar en posición vertical, cualquier objeto en horizontal será menos tenido en cuenta, como se puede ver en la figura 2.3.

También se puede extraer el tamaño del objeto; sin embargo, hay que tener en cuenta que no vale con la medida tradicional, puesto que objetos más cercanos a la cámara darán valores de tamaño mayores, indicando que éstos tienen más importancia cuando esto no tiene por que ser así. Para solucionar esto, en [3, 4], el área obtenida se normaliza dividiéndola entre la distancia entre el objeto y la cámara, pero para esto es necesaria información 3D.

En [1], se propone una aproximación que no requiere información adicional, como en el caso anterior. Se plantea la normalización del área de los objetos entre la mitad de su perímetro. De manera que se reducen las diferencias entre objetos grandes y pequeños.

Otra de las características que se propone extraer en [1] es la localización del objeto; para ello se divide la zona captada en regiones que no se solapen, y a cada una se la otorga una importancia distinta, valores entre $[0, 1]$. Esto puede resultar muy útil para casos como el fútbol, donde se podría distinguir entre jugadores dentro o fuera del área, y darles mayor o menor importancia según su localización.

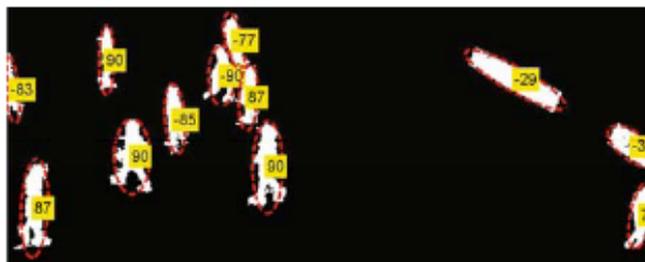


Figura 2.3: Clasificación según la orientación

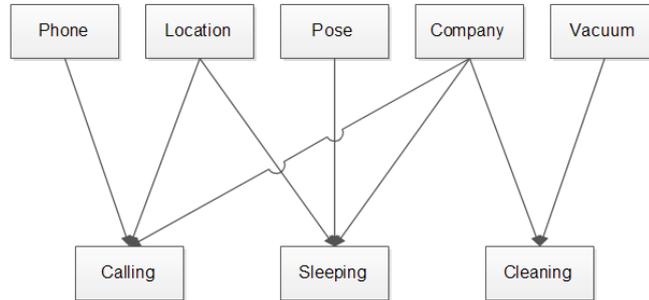


Figura 2.4: Red Bayesiana

2.2.2. Detección de eventos

Para la detección de eventos es necesario en primer lugar la extracción de características del vídeo, y una vez analizadas éstas, se busca si se ha producido un evento o no.

En [5] se propone definir un conjunto de eventos para el entorno a analizar, una vez elegidos se relacionan, en este caso, con objetos y posturas (características extraídas del vídeo).

En este caso se define:

- Evento: Llamando, limpiando, conversando, reunión, presentación, durmiendo y trabajando.
- Objeto: Ordenador, teléfono, cuaderno, aspiradora, etc.
- Postura: De pie, sentado, etc.

Una vez definidos los eventos y relacionados con las características analizadas, se diseña un modelo de red Bayesiana, como el ejemplo de la figura 2.4. Cuando se encuentren las características asociadas a un evento éste será marcado como reconocido.

Para la selección de vista, cada evento tendrá asociada una importancia, que junto con las demás características extraídas de cada cámara proporcionarán un valor distinto a cada una de ellas, ya que un mismo evento puede ser captado por distintas cámaras.

Así mismo, una misma cámara puede capturar varios eventos.

Otra propuesta de detección de eventos es la que se encuentra en [1], donde se utiliza una combinación de características de bajo y alto nivel. Las de bajo nivel incluyen, las que se han citado en el punto anterior, vectores de movimiento y detección de actividad. Mientras que las de alto nivel se basan en un detector de objeto, también citado en el apartado anterior.

Como en la propuesta anterior, aquí también se realiza una lista de eventos para cada dataset, y una puntuación para cada uno de los eventos según su importancia.

Por otro lado hay que destacar que este modelo hace uso de umbrales, para las características tales como vectores de movimiento, y características a nivel de *frame*. Para entender esto mejor, se usará el ejemplo de un dataset de baloncesto: un evento podría ser “intento de canasta”, en este caso se comprobaría el posicionamiento de los objetos en la vecindad de la canasta y si la magnitud de los vectores de movimiento superan cierto umbral; de ser así, se supondría “intento de canasta”. Pero todavía faltaría un



Figura 2.5: Evento: Intento de canasta



Figura 2.6: Puntuación de objeto

detalle: que estas características, posición y magnitud, se mantengan en esos valores o por encima de ese umbral durante un mínimo número de *frames*, a esto se refiere con características a nivel de *frame*.

2.3. Puntuación del contenido

En [1] el contenido se puntúa en dos pasos. En el primero se puntúan individualmente los objetos captados en cada cámara independientemente, y en segundo lugar se combina esa puntuación con las características generales a nivel de *frame* para obtener la puntuación final de cada cámara.

Los objetos se puntúan de acuerdo a la característica de tamaño multiplicada con la característica de posición. Como se ve en la figura 2.6, cuando el jugador pasa la línea de 3, su puntuación se ve aumentada aunque el tamaño no ha cambiado significativamente, esto se debe a la característica de posición.

Para el segundo paso se hace uso de las puntuaciones de los objetos junto con la cantidad de movimiento, número de objetos y eventos detectados.

En [6] la puntuación total de cada una de las cámaras se calcula como una suma ponderada de las características individuales de cada objeto. Debido a que esta aproximación no utiliza análisis de la escena global (como detección de eventos, calidad de la imagen, etc), es bastante probable que al solo examinar las características de los objetos, la cámara con mayor puntuación sea aquella con más objetos, ya que

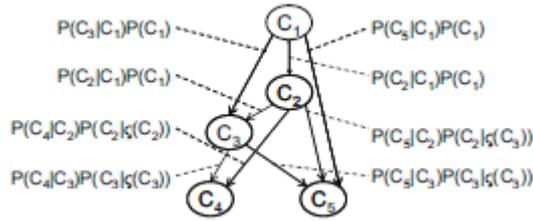


Figura 2.7: Grafo DBN

proporcionarán más valores a la suma. De tal manera que puede producirse una pérdida de información al ser ignoradas cámaras con menos objetos pero con contenido interesante.

2.4. Selección de vista

La selección de la vista según la puntuación *frame a frame* puede provocar cambios frecuentes entre cámaras, produciendo un efecto de parpadeo incómodo para el espectador. Para la solución de este problema, [7, 8]añaden una constante que constituirá el mínimo tiempo de visionado, es decir, un número mínimo de *frames* que han de reproducirse antes de cambiar a otra cámara. Sin embargo, si esto no se controla adecuadamente puede causar pérdida de información.

El método utilizado en [9] incluye un comportamiento simple gracias al uso de una máquina de estados finita, la cual proporciona continuidad en lo referente al cambio de cámara.

En [10] se propone utilizar Partially Observable Markov Decision Processes (POMDP) para así estimar el estado del objetivo en cualquier momento y elegir una configuración para la cámara de manera que el error de estimación en detectar el estado del objeto es minimizado. Este método no tiene en cuenta la interacción de los objetos con el entorno y no hay formulación para múltiples objetos.

En [11] se utilizan políticas de planificación “avariciosas”, basadas en técnicas First Come First Served (FCFS) and Earliest Deadline First (EDF). Sin embargo estas aproximaciones no tienen en cuenta las posibles consecuencias del cambio de cámara al cambiar de objetivo.

En [1] se propone evitar el efecto parpadeo considerando los estados anteriores y teniendo un conocimiento previo sobre la red de cámaras. El conocimiento previo se refiere a la actividad en cada cámara y su localización. Este modelo utiliza Redes Bayesianas Dinámicas (DBN).

Un DBN es un grafo donde cada nodo es conectado a otro mediante un arco. Siendo cada nodo una cámara, cada cambio será dependiente de la cámara anterior, siguiendo un orden establecido en el grafo.

En este caso el grafo tendrá $N + 1$ nodos, donde N será el número de cámaras, dejando el primer estado para la etapa de aprendizaje (detección de background).

Cada cámara, como vemos en la figura 2.7, puede estar conectada a varias, de tal manera que cada arco estará asociado a una probabilidad.

Capítulo 3

Diseño, desarrollo e integración

3.1. Introducción

El objetivo del proyecto es obtener a la salida un vídeo que será montado a partir de varios vídeos de distintas cámaras. Para ello se evaluará cada uno de los *frames* de cada una de las cámaras y se le asignará una puntuación según los siguientes criterios:

1. Número de blobs.
2. Cantidad de área del *frame* que varía con respecto al *frame* anterior.
3. Calidad de la imagen, en términos de emborronamiento (blur).
4. Atención del usuario.

3.2. Arquitectura del sistema

En la figura 3.1 se pretende ilustrar la línea general del proyecto. Como se puede apreciar, el sistema se alimentará de varias cámaras. Los vídeos obtenidos de estas señales de entrada serán tratados y analizados para obtener una puntuación de cada *frame*, obteniendo a la salida, como resultado, el montaje de un vídeo conformado por las secuencias más importantes de cada una de las cámaras de entrada.

En la figura vemos el flujo de cada una de las señales a lo largo del sistema.

A continuación, y visto ya el diagrama de bloques, se pasará a hacer una explicación más detallada de la función que realiza cada uno de los bloques que conforman el diseño.

3.2.1. Resize

3.2.1.1. Introducción

A lo largo de este proceso, la entrada y la salida serán vídeos, pero internamente se trabajará con imágenes. Lo normal es que los vídeos a la entrada sean de iguales características, pero esto no es siempre así, ya que en ocasiones el tamaño de las imágenes de una cámara puede variar con respecto a las otras.

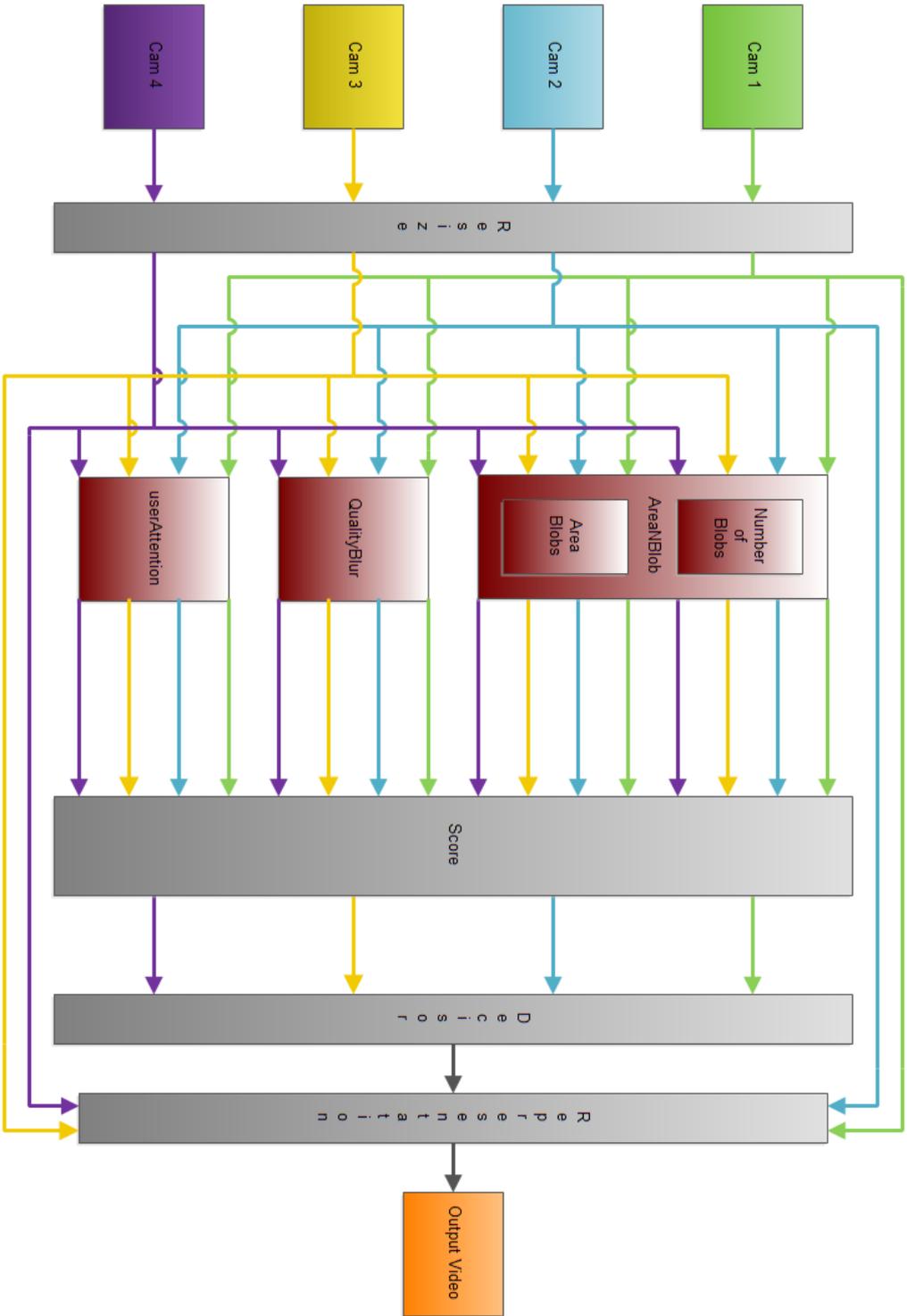


Figura 3.1: Diagrama de bloques

Este bloque se encarga de transformar las distintas imágenes de las cámaras a la misma resolución, en este caso a 640×480 .

3.2.1.2. Entradas

Capture_cam corresponde al `cvCapture` del vídeo de una de las cámaras.

Image_resize es una imagen en blanco, con el tamaño al que se van a transformar las capturadas de los vídeos. Posteriormente se reutilizará esta variable para proporcionar la salida.

3.2.1.3. Salidas

Image_resize en este caso la imagen de salida coincide con la de entrada. Esto se debe a que la imagen de salida se sobrepondrá en la de entrada, ahorrando así una variable. La imagen de salida será la de entrada interpolada o diezmada, según sea necesario para ajustarla al tamaño objetivo.

3.2.1.4. Desarrollo

Este bloque es una adaptación de la función de OpenCV, `cvQueryFrame`, cuya función es obtener un *frame* de un vídeo. La diferencia entre ambas es simplemente el redimensionado de la imagen obtenida del vídeo. Primeramente se obtiene la imagen, con la función `cvQueryFrame`, pero después se redimensiona mediante interpolación y submuestreo al tamaño elegido. En este caso se ha elegido como tamaño estándar 640×480 , por lo tanto todas las imágenes capturadas de las distintas cámaras pese a sus diferentes dimensiones acabarán siendo de tamaño homogéneo, así la salida también compartirá esta misma resolución. Esta parte compone el algoritmo `cvQueryFrameResize`.

Finalmente la imagen redimensionada, se devolverá usando la imagen de entrada *Image_resize*.

Cabe destacar, que en ocasiones puede ser necesaria la interpolación o el submuestreo. En los casos sometidos a investigación no ha sido necesario realizar ningún proceso de submuestreo, por lo tanto el filtro de submuestreo queda pendiente para trabajos futuros. En el caso de la interpolación, se ha elegido realizar sobre esas imágenes interpolación bilineal, por su rapidez y sencillez. No obstante esto puede ser sometido a un estudio más intensivo, como se propondrá en trabajos futuros.5.2.2

3.2.2. AreaNBlobs

3.2.2.1. Introducción

Previamente, para la obtención de los blobs hay que hacer una extracción de fondo, de tal forma que se obvие toda parte de la imagen que esté estática, pudiendo así encontrar todo lo que esté en movimiento. Una vez hecho esto se podrá extraer los blobs, que no son más que conjuntos de píxeles que están en movimiento, y por lo tanto el número de ellos que habrá por *frame* y el área que ocuparán de la imagen total.

Con estos dos parámetros: número de blobs y área de los mismos, se obtendrán dos coeficientes, normalizados entre $[0, 1]$. El primero será normalizado mediante una variable controlada por el usuario, número máximo de blobs esperados; por ejemplo, en el caso de un dataset de fútbol, el valor máximo

esperado será el número de jugadores de ambos equipos, los árbitros y la pelota, así se podría poner un máximo de 26. El segundo parámetro, será normalizado por el número total de píxeles de la imagen.

3.2.2.2. Entradas

Image_cam se trata de una variable *IplImage*, aquí irá el fotograma que se va a analizar, tras haber sido obtenido del vídeo.

pGamma_cam objeto usado para la extracción del fondo de cada una de las imágenes. Algoritmo proporcionado por VPULab.

pBlobExtractor_cam objeto usado para la extracción de los blobs de cada una de las imágenes. Algoritmo proporcionado por VPULab.

minWidth anchura mínima de blob, usada para la extracción de blobs del *frame (Image_cam)*, valor elegido por el usuario.

minHeight altura mínima de blob, usada para la extracción de blobs del *frame (Image_cam)*, valor elegido por el usuario.

maxWidth anchura máxima de blob, usada para la extracción de blobs del *frame (Image_cam)*, valor elegido por el usuario.

maxHeight altura máxima de blob, usada para la extracción de blobs del *frame (Image_cam)*, valor elegido por el usuario.

cuenta_frames es el contador de *frames*, utilizado para que la lectura de los *frames* del vídeo continúe.

blobs_max es el número de blobs máximo esperado por el usuario, el cual es solicitado al comienzo del programa.

3.2.2.3. Salidas

Num_blobs esta será la variable donde se almacenará el coeficiente de número de blobs, el cual oscilará entre $[0, 1]$.

Area_blobs aquí se devolverá el coeficiente del área que ocupan los blobs de la imagen, también entre $[0, 1]$.

3.2.2.4. Parámetros configurables

Los siguientes parámetros son elegidos por el usuario, según sus preferencias a la hora de montar el vídeo de salida.

minWidth

minHeight

maxWidth

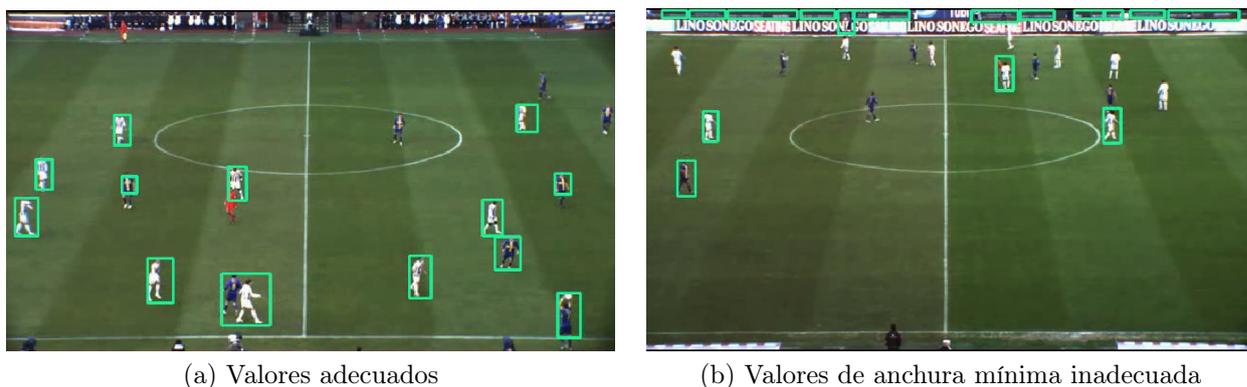


Figura 3.2: Blobs según altura y anchura mínima

maxHeight

blobs_max

3.2.2.5. Desarrollo

Para poder obtener los blobs, como se ha mencionado antes, primeramente hay que realizar una extracción de fondo y, una vez hecha ésta, se obtiene lo que está en el frente, que es lo que está en movimiento, ya que si algo permanece estático el tiempo suficiente pasará a formar parte del fondo. Cuando se tiene extraído el frente, se pueden obtener los blobs. Todo este proceso se lleva a cabo con el algoritmo cedido por el VPU Lab.

AreaNBlobs es el bloque que se encarga de extraer las dos primeras características del análisis, las dos referentes al contenido de la imagen. Por un lado se obtendrán el Núm. de blobs, y por otro el área que ocupan estos mismos blobs.

Para la extracción de los blobs, primeramente hay que establecer los tamaños mínimos y máximos tanto de altura como anchura, esto es muy importante, ya que según el contenido del vídeo variarán, haciendo necesario que se reajusten para cada dataset y cada cámara, es por esto que son solicitados al usuario.

Dado que cada cámara tiene distintos ángulos de visión, el tamaño de los objetos varía, y por lo tanto el de los blobs también, es por eso que cada cámara tiene sus máximos y mínimos independiente de las demás.

Cuando se elijan estos parámetros hay que tener en cuenta que es lo que se está buscando en el vídeo: personas, objetos, etc. y ser muy cuidadoso, ya que si se eligen bien se reducen las probabilidades de marcar como blobs elementos que no interesan para análisis, ya sean sombras, cambios de iluminación, etc. Ver Figura 3.3.

El usuario tendrá dos opciones: elegir él mismo el tamaño que desee para cada cámara, o elegir el optimizado para cada dataset y cámara, el cual se encontrará embebido en el código.

Una vez seleccionado el tamaño de los blobs, se puede, como se realizará en esta implementación, contar el número de blobs y así tener un valor para comparar. En este caso, se limita el tamaño de los

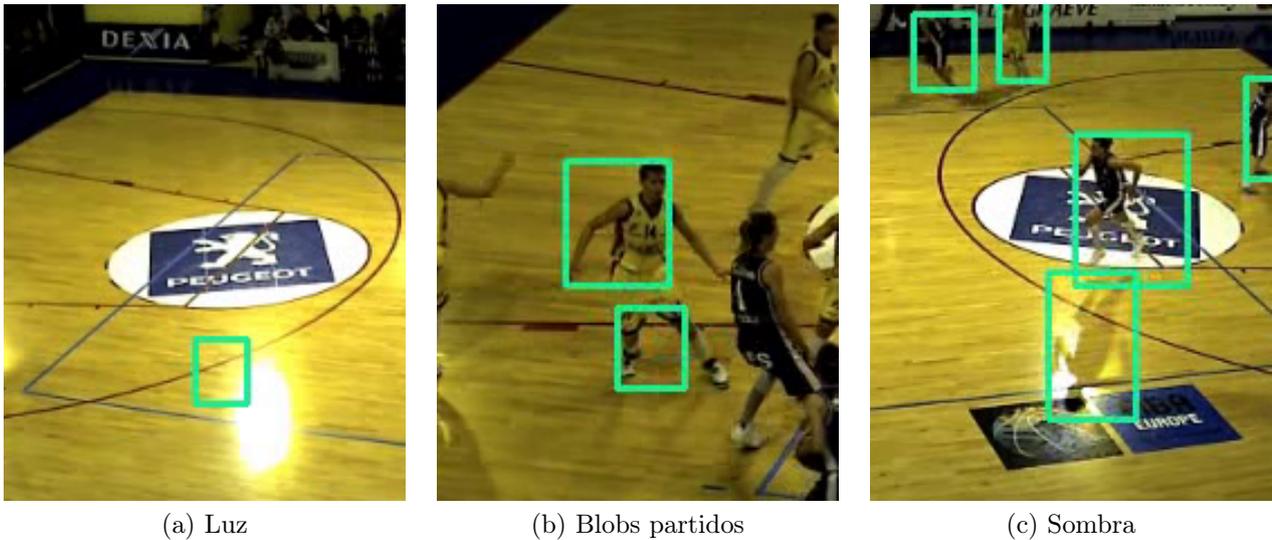


Figura 3.3: Problemas con los blobs

blobs para que capte en su mayoría personas (jugadores) de tal manera que este valor será sinónimo de personas en el campo. Ver figura 15.

Hay que tener en cuenta que no sería para nada incorrecto que un blob marcara algo que no es una persona, pues no hay que olvidar que el blob denota movimiento de fotograma a fotograma, y, aunque no sea una persona, se asume que algo en movimiento es más importante que algo quieto.

Para obtener el coeficiente entre $[0, 1]$ el número de blobs obtenidos por el algoritmo se normalizará, dividiendo entre el número máximo de blobs esperados, cuyo valor se solicita al usuario al comienzo de la ejecución, ver sección 3.3.

En el caso de un partido de fútbol, como en el ejemplo, si se ha elegido correctamente los tamaños de los blobs, un valor razonable por el que normalizar sería 26 (11 jugadores de cada equipo, el conjunto arbitral y el balón).

Existe el problema de los blobs partidos; esto es que, en ocasiones, un jugador puede mover un brazo y una pierna, pero el resto quedarse fijo, de tal manera que el algoritmo encontrará dos blobs donde solo hay un jugador, para este caso, si hay más de 26 blobs el coeficiente se quedará en 1 como máximo. Ver figura 3.3. No obstante cabe destacar que esto es una solución intermedia; quedaría como trabajo futuro buscar una segmentación mejor para evitar esto, puesto que el objetivo de este proyecto es la creación del sistema completo.

Por otro lado, se tiene el área que ocupan esos blobs, lo que proporciona información de todo lo que está en movimiento en la imagen, la mejor manera de normalizarlo es dividiendo entre el conjunto total de píxeles, en este caso y gracias a la normalización de tamaño de las imágenes, se sabe que conjunto total son $640 \cdot 480 = 307200$ píxeles, así este será el número por el que se divida.

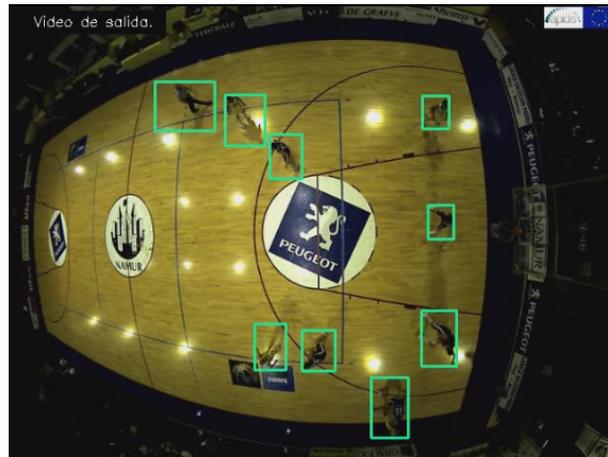


Figura 3.4: blob por persona

3.2.3. QualityBlur

3.2.3.1. Introducción

Este bloque será el encargado de asignar a cada *frame* un coeficiente de calidad, el cual será usado posteriormente por el bloque Score, para asignar una puntuación. El factor de calidad se basará en el emborronamiento de la imagen, el cual se calculará usando la DCT.

En definitiva, de este bloque se obtendrá un coeficiente que variará entre $[0, 1]$, para cada *frame* de cada cámara. Siendo el “1” totalmente nítido, y el “0” completamente emborronado.

3.2.3.2. Entradas

Image recibe una variable *IplImage*, donde se encuentra el fotograma a analizar.

3.2.3.3. Salidas

Blur devolverá un *double*, el coeficiente de calidad, el cual oscilará entre $[0, 1]$. Totalmente emborronado y nada, respectivamente.

3.2.3.4. Parámetros configurables

En el caso de este algoritmo no hay parámetros configurables por el usuario, aunque sí que hay que comentar la existencia de variables de configuración del programa, que se encuentran definidas en código, pero que dependiendo del entorno pueden resultar más o menos interesante alterar. La elección de sus valores se discutirá en A.

MaxHistValue Es el valor umbral por debajo del cual un coeficiente DCT es considerado borroso.

MinDCTValue Es el valor umbral mínimo que ha de superar para ser tenido en cuenta y no ser considerado ruido.

3.2.3.5. Desarrollo

El coeficiente de calidad será referente al emborronamiento de la imagen sometida a análisis.

Intuitivamente, emborronado (blur) es lo opuesto a nítido. El método propuesto en [1] busca la ausencia de nitidez, a través de los coeficientes de la DCT.

El uso de la DCT para determinar si una imagen es o no borrosa parte del hecho de que una imagen sin bordes carece de altas frecuencias. Esto quiere decir que sus principales coeficientes AC son en su mayoría cercanos a 0.

Se propone establecer una medida que tenga en cuenta la información de la DCT de la imagen entera, como un todo. Para ello se divide la imagen (luminancia) en bloques de 8x8, a los cuales se les aplicará la DCT.

Con los datos obtenidos de cada bloque se compone un histograma de 64 valores (8×8), que da una idea de cómo se distribuyen las frecuencias de la imagen. Para que un valor del histograma aumente, el coeficiente en la posición correspondiente tiene que tener un valor absoluto típico superior a 8 (*MinDCTValue*, como se verá en A para nuestro caso usaremos 10), que es el valor que tendría el coeficiente DC de un bloque uniforme de luminancia 1. Si es menor a este umbral se desecha, ya que puede ser debido a ruido.

Una vez construido el histograma, se estima el coeficiente de emborronamiento a partir de éste y de una matriz de pesos, considerando que un coeficiente es borroso si su aparición en el histograma es menor que un 10% de las veces que aparece el coeficiente DC (*MaxHistValue*). Si efectivamente es menor, se suma el valor de la matriz de pesos que corresponde a esa posición.

Finalmente, valor obtenido se divide entre 344 (suma de todos los números de la matriz de pesos) para así obtener un valor entre $[0, 1]$. Se realiza la operación $1 - \text{Coeef. de emborronamiento}$, de tal manera que el 0 significaría totalmente borrosa y el 1 totalmente nítida.

3.2.4. userAttention

3.2.4.1. Introducción

Este bloque simulará el estado de atención del usuario, que será representado como una función decreciente, es decir, a medida que pasan los *frames*, si proceden de la misma cámara, éstos van sufriendo una leve penalización, simulando así el “aburrimiento” del usuario.

3.2.4.2. Entradas

Counter esta variable será el argumento de la función matemática, el cual comenzará con valor Control, e irá disminuyendo a medida que la misma cámara se siga mostrando.

Control es el límite superior del argumento de la función matemática, este valor permitirá manipular la penalización que se hará sobre los *frames*.

EstadoAnterior indica que cámara fue mostrada en el *frame* anterior.

Cam es el valor de la cámara que está siendo evaluada.

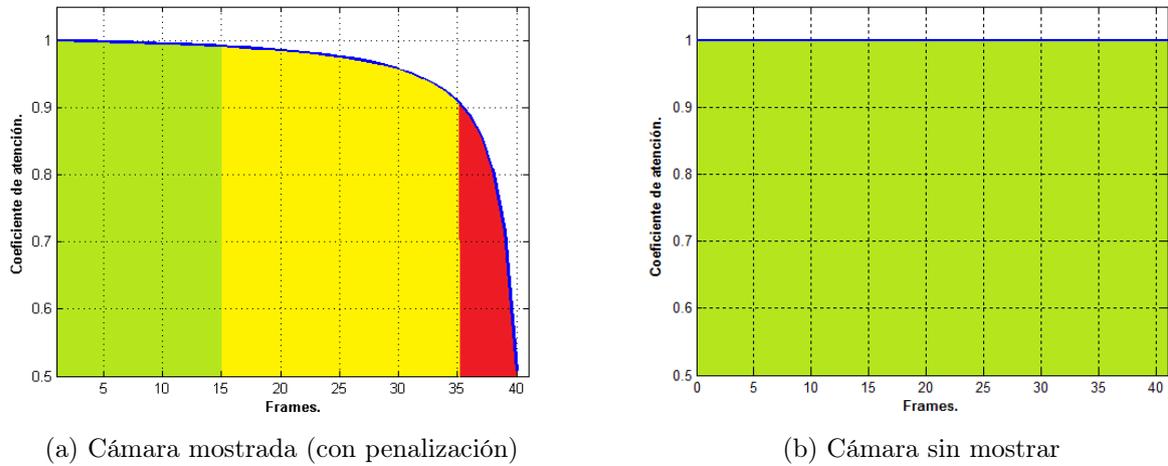


Figura 3.5: Coeficiente de atención del usuario.

Reset es un interruptor, se pondrá a 1 en el caso de que la puntuación de la cámara seleccionada haya aumentado con respecto a los instantes anteriores. Provocará la reinicialización de **Counter** al valor de **Control**, es decir, reinicializa el Coef. de Atención (**CoefAttention**).

3.2.4.3. Salidas

CoefAttention se trata de un double donde irá el valor obtenido de la función matemática elegida; oscilará entre $[0, 1]$.

3.2.4.4. Parámetros configurables

Control

3.2.4.5. Estudio

El resultado de este bloque será la devolución del Coef. de Atención, cuyo valor oscila entre $[0, 1]$.

Lo primero que se debe elegir es qué función matemática se va a usar para simular el decaimiento de la atención del usuario. Parece lógico pensar que sea una función con una parte más o menos plana al comienzo, incluso con un leve descenso, y que a medida que pasa el tiempo (*frames*) se incremente la pendiente negativa. Pudiendo llegar a estancarse en un valor elegido, cero o continuando hacia valores negativos (lo cual solamente indicaría más penalización).

Este bloque proporcionará un coeficiente de atención por cada *frame*, que posteriormente será usado para asignarle puntuación a dicho *frame*.

Como se ve en la figura 3.5, a medida que van pasando *frames* de la misma cámara, el coeficiente de atención va disminuyendo, cada vez con más pendiente, sin embargo en la cámara que no se muestra, el coeficiente se mantiene constante.

Para este bloque se van a evaluar dos posibles funciones, la logarítmica y arcotangente.

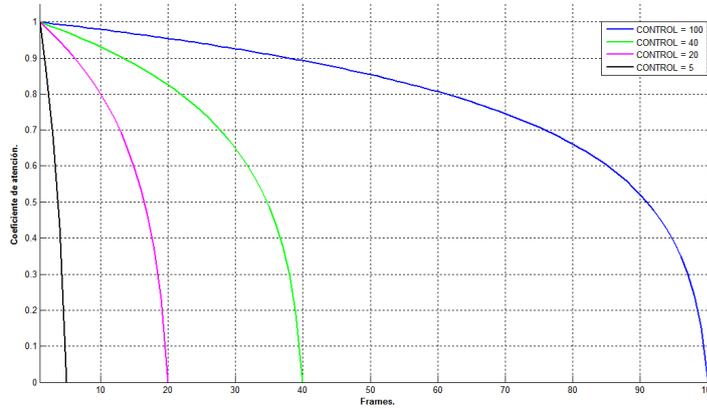


Figura 3.6: Logarítmica

3.2.4.6. Logarítmica

La función que se va a aplicar aquí es la siguiente.

$$\frac{\log_{10}(\text{Control}-x)}{\log_{10}(\text{Control})}$$

Hay que destacar que debido a la respuesta del logaritmo a una entrada igual a cero, en este algoritmo no se le permitirá llegar a tal valor.

En este caso el valor que determina si se realizarán transiciones cortas o largas será la variable **Control**, según se quiera empezar a penalizar el *frame* antes o después, se colocará un valor menor o mayor, respectivamente, lo que conllevaría montajes con transiciones más rápidas o más lentas.

Como se aprecia en la figura 3.6, el primer *frame* no obtiene penalización, pero a partir de éste los demás sí la tendrán. La pendiente de esta función es muy abrupta para el uso que se le quiere dar, ya que en principio no se busca comenzar a penalizar el *frame* tan pronto.

Por esta razón, se plantea una alternativa a la función anterior, añadir una parte plana previa a la logarítmica, la cual se podrá variar en longitud, según se busquen transiciones más o menos rápidas.

$$\begin{cases} 1 & x < \text{ParteCTE} \\ \frac{\log_{10}(\text{Control}-(x-\text{ParteCTE}+1))}{\log_{10}(\text{Control})} & x \geq \text{ParteCTE} \end{cases}$$

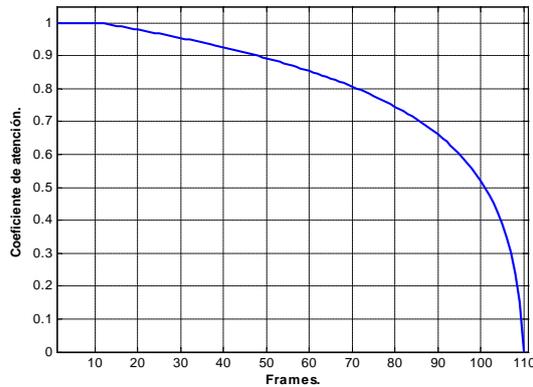
ParteCTE será el número de *frames* en los que no se introducirá penalización.

Con esta adición, ahora sí cumple la necesidad de no penalizar al comienzo los *frames*, para luego a medida que pase el tiempo ir reduciendo su importancia, en el caso expuesto, hasta llegar a “0”, se podría haber limitado la penalización a un valor determinado como se verá en 3.2.4.7.

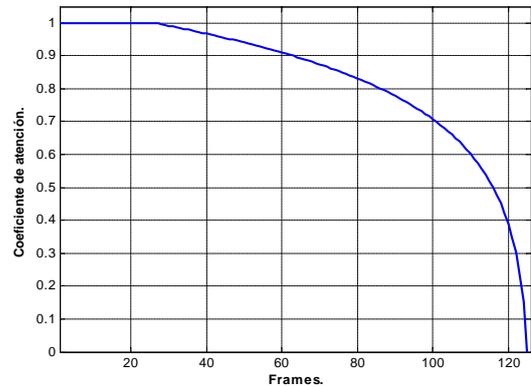
3.2.4.7. Arcotangente

La formulación de esta función será la siguiente.

$$\frac{\arctan(\text{Control}-x)}{\arctan(\text{Control})}$$



(a) $ParteCTE = 10 \text{ frames}$



(b) $ParteCTE = 25 \text{ frames}$

Figura 3.7: Logarítmica con parte plana

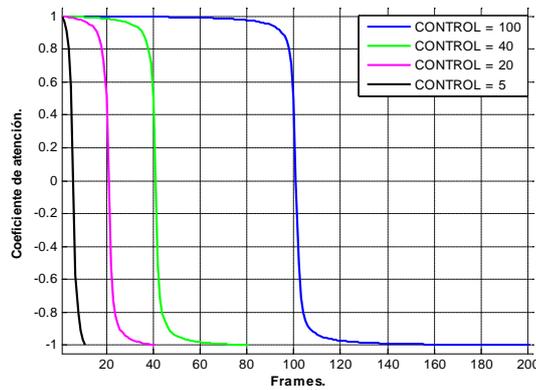


Figura 3.8: Arcotangente

De nuevo la variable **Control** determinará a partir de que *frame* comienza la penalización. Se puede ver gráficamente en la figura 3.8.

Se aprecia cómo al variar **Control** la penalización cambia, es más, se puede ver que el valor de **Control** coincide con el *frame* en el que la pendiente de penalización es máxima.

Analizando la figura, se observa un buen comportamiento para los primeros *frames*, puesto que no sufren penalización, y cuando empiezan a sufrirla es muy progresiva. En el tramo medio se produce el gran cambio, llega la pendiente máxima, que coincide con la pérdida de atención del usuario, llegando a valores negativos, este aspecto se discutirá más adelante.

3.2.4.8. Desarrollo.

Entre las dos posibles funciones evaluadas se ha elegido la arcotangente, por ser menos abrupta en el inicio, ya que sin necesidad de otra función incluye, la parte plana al comienzo, también por la gran pendiente a la mitad la cual simula adecuadamente el aburrimiento del usuario.

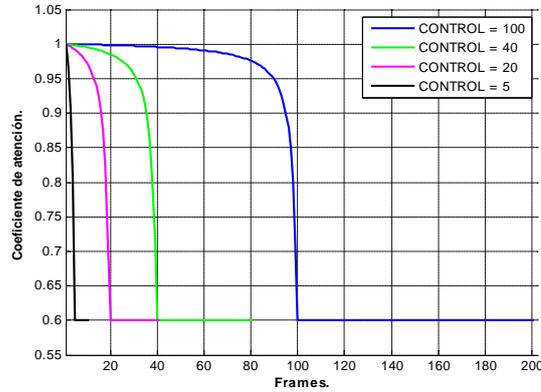


Figura 3.9: Arcotangente limitada

No obstante, se va a llevar a cabo una modificación en la parte final de la función, ya que hay que tener en cuenta que por mucho que una cámara se mantenga y empiece a aburrir al usuario, esa cámara siempre será más importante que una que no muestre nada en ese instante, es decir, hay que penalizar pero llegando a un compromiso entre el aburrimiento del usuario y el contenido de la cámara. Si se tiene en cuenta el entorno que se presentará más adelante, de un partido de baloncesto, aunque la cámara 1, que muestra todos los jugadores debajo de la canasta, lleve dos minutos reproduciéndose, no debería cambiarse por la cámara 6, en la que no hay más que pista vacía.

Por estos motivos se presenta una variación de la función anterior de la arcotangente, la cual limita la penalización a un 60%, como se aprecia en la figura 3.9.

Este bloque también incluye un algoritmo que permite la posibilidad de reinicializar la penalización. Hasta ahora si una cámara llevaba cierto tiempo mostrándose como salida, los *frames* siguientes empezaban a ser penalizados, esto es correcto, pero qué pasa si en ese transcurso un evento novedoso es grabado por la misma cámara, eso sin duda se ganaría la atención del usuario de nuevo, y por lo tanto el algoritmo ha de tenerlo en cuenta, reseteando el coeficiente de atención a 1.

Para poder satisfacer esta necesidad se ha incluido un detector, cuya función es comprobar si la puntuación de los *frames* siguientes al inicial es mayor a cierto umbral, de ser así, se procederá a reinicializar la simulación de atención del usuario. Ver figura 3.10

$$Score_{nuevo\ frame} \geq 0,2 \cdot Score_{frame\ inicial} + Score_{frame\ inicial}$$

Como se ve en el ejemplo la cámara 7 se empieza a mostrar cuando tiene un $score = 0,14447$, por lo tanto el límite de reinicialización se fija en $0,2 \cdot Score_{frame\ inicial} + Score_{frame\ inicial} = 0,173364$, la primera vez que se supere este límite superior el coeficiente de atención se reinicializará, como se ve cuando el score alcanza el valor $0,20950$.

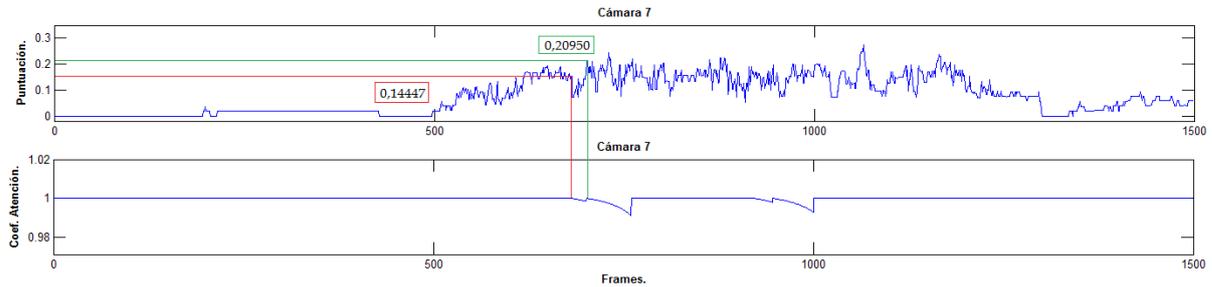


Figura 3.10: Arcotangente con reinicialización

3.2.5. Score

3.2.5.1. Introducción

Será el que finalmente otorgue una puntuación a cada *frame*, para ello recibirá los cuatro coeficientes citados anteriormente, por cada fotograma, y los fusionará mediante una función matemática, dando como resultado otro valor entre $[0, 1]$, donde el “1” será puntuación máxima y el “0” la mínima.

3.2.5.2. Entradas

Num_blobs obtenido en 3.2.2, que denota el coeficiente de número de blobs en el *frame* analizado, normalizado entre $[0, 1]$.

Area_blobs obtenido en 3.2.2, coeficiente del área encerrado por los blobs del *frame* analizado, normalizado entre $[0, 1]$.

Blur obtenido en 3.2.3, coeficiente de emborronamiento del *frame* analizado, normalizado entre $[0, 1]$.

CoefAttention obtenido en 3.2.4, coeficiente de atención del usuario para el *frame* analizado, normalizado entre $[0, 1]$.

3.2.5.3. Salidas

Score es un double resultado de la función matemática que relaciona los cuatro valores de entrada. Oscila entre $[0, 1]$.

3.2.5.4. Desarrollo

La manera de fusionar todas las características extraídas, está sujeta a muchas consideraciones, las cuales no son el objetivo de este proyecto, debido a la limitación en tiempo que conlleva un PFC. A continuación se plantea una de las muchas posibilidades.

$$Score = Coef.Atención \cdot Blur \cdot (0,5 \cdot Area + 0,5 \cdot NumBlobs)$$

Como se puede observar tanto la atención del usuario como la calidad del *frame*, son directamente proporcionales a la puntuación de éste. Se puede considerar que ponderan el valor del *frame*, si la calidad

es perfecta y la atención del usuario completa, la puntuación del *frame* será totalmente el valor de $(0,5 \cdot Area + 0,5 \cdot NumBlobs)$.

Por otro lado tanto el área como el número de blobs, forman la puntuación máxima del *frame*. En este caso se le dará a cada uno la misma importancia que al otro.

Como parte de la función del bloque **Score** se podrían añadir más características extraídas del vídeo, como por ejemplo la detección de eventos, sin embargo este proyecto pretende ser la base del sistema, por lo tanto esto se dejará para trabajos futuros.

3.2.6. Decisor

3.2.6.1. Introducción

No siempre la cámara con más puntuación será la cámara de salida, y ahí es donde interviene este bloque. Para elegir cual será la salida, se tendrá en cuenta, entre otras cosas, si hay otra cámara con la misma puntuación pero con menos *frames* reproducidos y el tiempo que lleva mostrándose la misma cámara, ya que tendrá que salir durante un tiempo mínimo para evitar el molesto parpadeo en el vídeo de salida, etc.

3.2.6.2. Entradas

Cam_ Output es un entero, que indica cual es la cámara elegida como salida.

EstadoAnterior es un entero, indica la cámara mostrada en el momento anterior. Esta variable es la que se utilizará cuando se agregue el algoritmo de orden de cámara, sobre lo que se hablará en trabajo futuro, 5.2.4.

Suma_ total Aquí se almacenan la suma de las puntuaciones de cada cámara, $\sum^{retardo} Score$.

Resistencia es la variable que cuenta el número de *frames* mostrados por la misma cámara.

AntiFlickering es el límite mínimo de *frames* que una misma cámara ha de mostrar una vez seleccionada. Cuando **Resistencia** supere este valor se podrá producir un cambio de cámara.

HistogramCams Lleva la cuenta de cuantos *frames* lleva cada cámara.

NCAMS Es el número total de cámaras que compone el dataset.

Contador Cuenta el número de *frames*. Al llegar al valor de Retardo se pone a cero y vuelve a empezar.

Retardo Antes de elegir la cámara de salida el sistema espera a analizar un número N de *frames*, tras ese intervalo la cámara con más puntuación durante dicho periodo será la elegida. Ese número N es la variable Retardo.

3.2.6.3. Salidas

Como en el caso del bloque **AreaNBlobs** la función en sí, no devuelve nada, los parámetros son pasados por referencia. En este caso **Cam_ Output**, **EstadoAnterior**, **Resistencia** y **Contador**.

3.2.6.4. Parámetros configurables

AntiFlickering.

Retardo.

Tienen una relación muy estrecha entre ellos, la elección de los valores determinará el tipo de montaje. Se realiza un análisis exhaustivo en el siguiente punto.

3.2.6.5. Desarrollo

La función del bloque **Decisor** es la que finalmente elige que cámara será mostrada, para ello actúa con un pequeño retraso, denotado por la variable *Retardo*, la cual como se ha comentado antes es elegida por el usuario.

Durante el número de *frames* seleccionados por la variable *Retardo*, se van sumando los valores de la puntuación de cada cámara en otra variable, *suma_total*, la cual es la que finalmente se usará para tomar la decisión. Cada N frames, donde N es el valor de *Retardo*, se comprueba el valor de *suma_total*, y si se ha superado el tiempo de *AntiFlickering*, se escoge la cámara de mayor puntuación.

Sin embargo hay una excepción, y es el caso del empate a puntuación. En ese caso, la cámara elegida será la que menos veces haya sido mostrada, ya que supone más novedad para el usuario.

Hay que elegir el tamaño de estas variables cuidadosamente, ya que la relación entre ellas puede provocar montajes muy diferentes. Ver figura 3.11.

Como se puede ver en la imagen, según el valor de *Retardo* que elijamos respecto al *AntiFlickering*, tendremos distintos tipos de montaje.

Si el *Retardo* es demasiado pequeño se pueden colar espúreos, entendidos éstos como periodos pequeños en los que una cámara tiene mucha puntuación, pero en los siguientes no, si esta cámara es seleccionada por causa de este *Retardo* tan pequeño, estará siendo mostrada un mínimo de tiempo igual al denotado por la variable *AntiFlickering*.

Sin embargo, si es casi tan grande como la variable *AntiFlickering*, se presentará un problema de falseo; cuando una cámara es seleccionada, ésta estará un mínimo de tiempo, sino se seguirá mostrando hasta que lo alcance, en el caso citado, al ser el *Retardo* un poco más pequeño que *AntiFlickering*, se puede afirmar con total seguridad que toda cámara seleccionada se mostrará como mínimo $2 \cdot Retardo \sim 2 \cdot Antiflickering$.

Se aprecia en la imagen cómo un valor intermedio entre los dos citados se adecua de mejor manera. Por un lado evita el seleccionar cámaras de corta estancia en el máximo, y por otro se aproxima mejor a la duración real de cada cámara en la posición de más puntuación.

Una primera aproximación, sobre la cual se puede trabajar más en proyectos futuros, indica que la elección del valor de *Retardo* como la mitad del valor de *AntiFlickering* da buenos resultados.

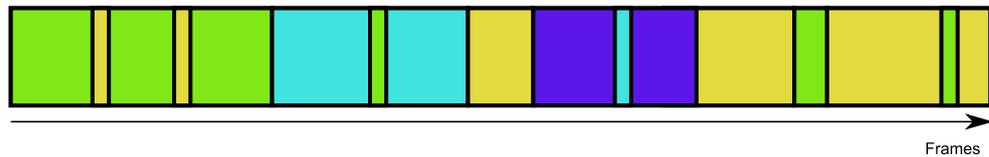
3.2.7. Presentation

3.2.7.1. Introducción

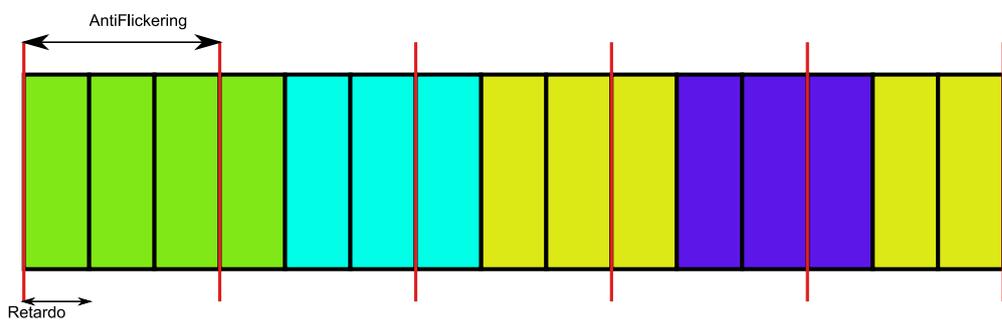
Después de que el **Decisor** especifique la cámara elegida para mostrar como salida, este bloque se encargará de realizar el archivo de vídeo final. En este caso serán dos vídeos, el primero “montaje.avi”

CAM 1
 CAM 2
 CAM 3
 CAM 4

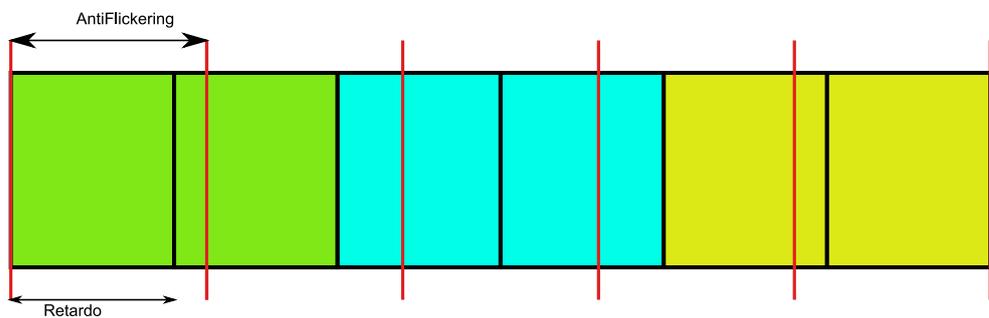
Cámara con más puntuación en cada instante.



Salida, según relación $\text{Retardo} = 1/3 \text{ AntiFlickering}$.



Salida, según relación $1/2 \text{ AntiFlickering} < \text{Retardo} < \text{AntiFlickering}$.



Salida, según relación $\text{Retardo} = 1/2 \text{ AntiFlickering}$.

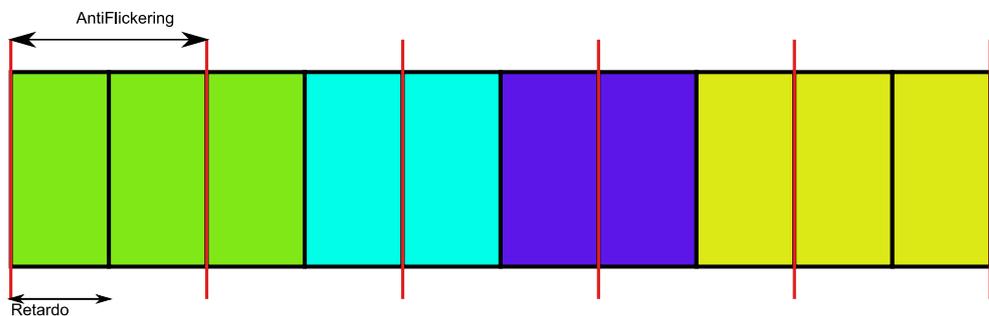


Figura 3.11: Imagen comparativa *AntiFlickering* vs. *Retardo*

mostrará únicamente la imagen de salida. El segundo, sin embargo, mostrará la imagen de salida y las de todas las cámaras.

3.2.7.2. Entradas

Fondo IplImage con la imagen que se utilizará como fondo.

Image_output la imagen de la cámara seleccionada como salida.

Image_texto es una IplImage creada con el tamaño que tendrá el texto de la presentación.

Image_salida es una IplImage creada con el tamaño que tendrá el vídeo de salida, es decir, 640×480 más el ancho necesario para colocar la rejilla de imágenes procedentes de las cámaras del dataset, en total 1150×480 .

txt_size es un cvSize, donde se encuentra las variables correspondientes al tamaño del texto.

font estructura font, en la cual se especifica el formato del texto.

Image_camX recibirá una imagen por cada cámara del dataset, para poder incrustarla en la imagen a devolver.

NCAMS número de cámaras que conforman el dataset.

3.2.7.3. Salidas

Image_prueba esta variable que se recibía como entrada, será rellenada con la imagen de presentación y devuelta.

3.2.7.4. Desarrollo

Una vez decidida cuál es la cámara que se va a mostrar, se pasan las imágenes a este bloque, el cual es el encargado de crear una presentación diferente, en ella se mostrarán la salida a tamaño completo, y en rejilla a la derecha las cámaras entre las cuales se ha elegido.

Para ello, se creará una imagen de un tamaño mayor a la salida, el vídeo de salida, por defecto, es de 640×480 , como la rejilla irá a la derecha hay que añadir ancho de imagen. En este caso será un tamaño de 1150×480 . Para crear una presentación más amigable, se coloca una imagen como fondo, la cual también será enviada por argumentos, así el usuario puede elegir la suya propia.

Una vez creada la imagen más ancha, y colocado el fondo, un algoritmo se encarga de insertar las imágenes de las cámaras, según los parámetros que le se envíen, 6, 7 o 9 cámaras (acorde con los dataset utilizados). La salida como se ha dicho anteriormente irá a tamaño completo (640×480) en la parte izquierda de la pantalla, y se le añadirá el texto “vídeo de salida”, mientras, en el lado derecho se colocarán las cámaras en miniatura, con un texto identificativo. Ver figura 3.12.

Este bloque se ha desarrollado para sistemas multicámara de 6 a 9 cámaras, aunque es extrapolable a otros números.



Figura 3.12: Presentación gráfica

3.3. Detalles de implementación

Al programa habrá que pasarle como argumentos los vídeos de las cámaras que se vayan a utilizar, así como la imagen que se vaya a utilizar como fondo de la presentación (ver 3.2.7). Una vez hecho esto se pedirá por pantalla una serie de parámetros, que permiten realizar los distintos tipos de montaje. Figura 3.13.

NCAMS

Blobs_max.

AntiFlickering

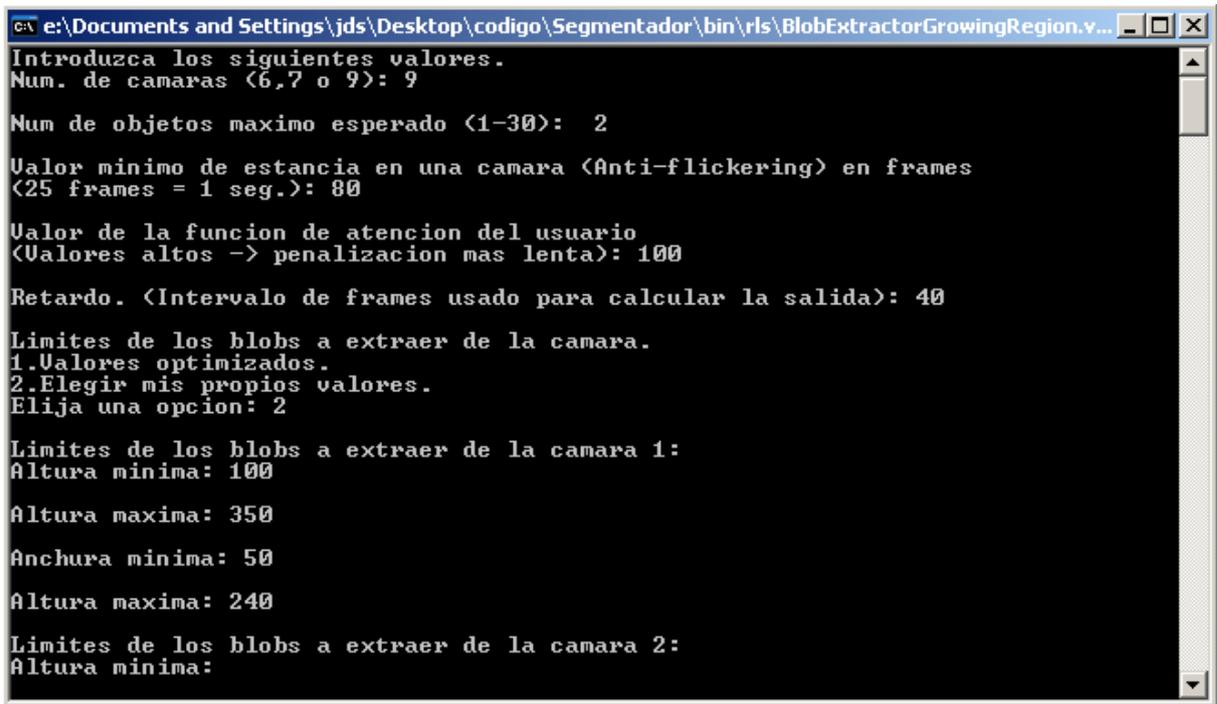
Control

Retardo

Límites en la selección de los blobs.

Una vez acabado esto, la participación activa del usuario terminará. Cuando finalice la ejecución se dispondrá de dos vídeos: uno mostrará únicamente la cámara de salida, y el otro una presentación especial, que incluye la salida y la miniatura de las diversas cámaras de entrada. También contará con una serie de documentos de texto, en los que se encuentran los datos obtenidos para cada *frame*:

- Núm. de blobs.
- Área de los blobs.
- Coeficiente de emborronamiento.
- Coeficiente de atención del usuario.



```
e:\Documents and Settings\jds\Desktop\codigo\Segmentador\bin\rls\BlobExtractorGrowingRegion.v...
Introduzca los siguientes valores.
Num. de camaras (6,7 o 9): 9
Num de objetos maximo esperado (1-30): 2
Valor minimo de estancia en una camara (Anti-flickering) en frames
(25 frames = 1 seg.): 80
Valor de la funcion de atencion del usuario
(Ualores altos -> penalizacion mas lenta): 100
Retardo. (Intervalo de frames usado para calcular la salida): 40
Limites de los blobs a extraer de la camara.
1.Valores optimizados.
2.Elegir mis propios valores.
Elija una opcion: 2
Limites de los blobs a extraer de la camara 1:
Altura minima: 100
Altura maxima: 350
Anchura minima: 50
Altura maxima: 240
Limites de los blobs a extraer de la camara 2:
Altura minima:
```

Figura 3.13: Petición de parámetros

- Puntuación.
- Cámara de salida.
- Histograma de cámaras utilizadas como salida.

Los cuales se pueden ver en la figura 3.14.

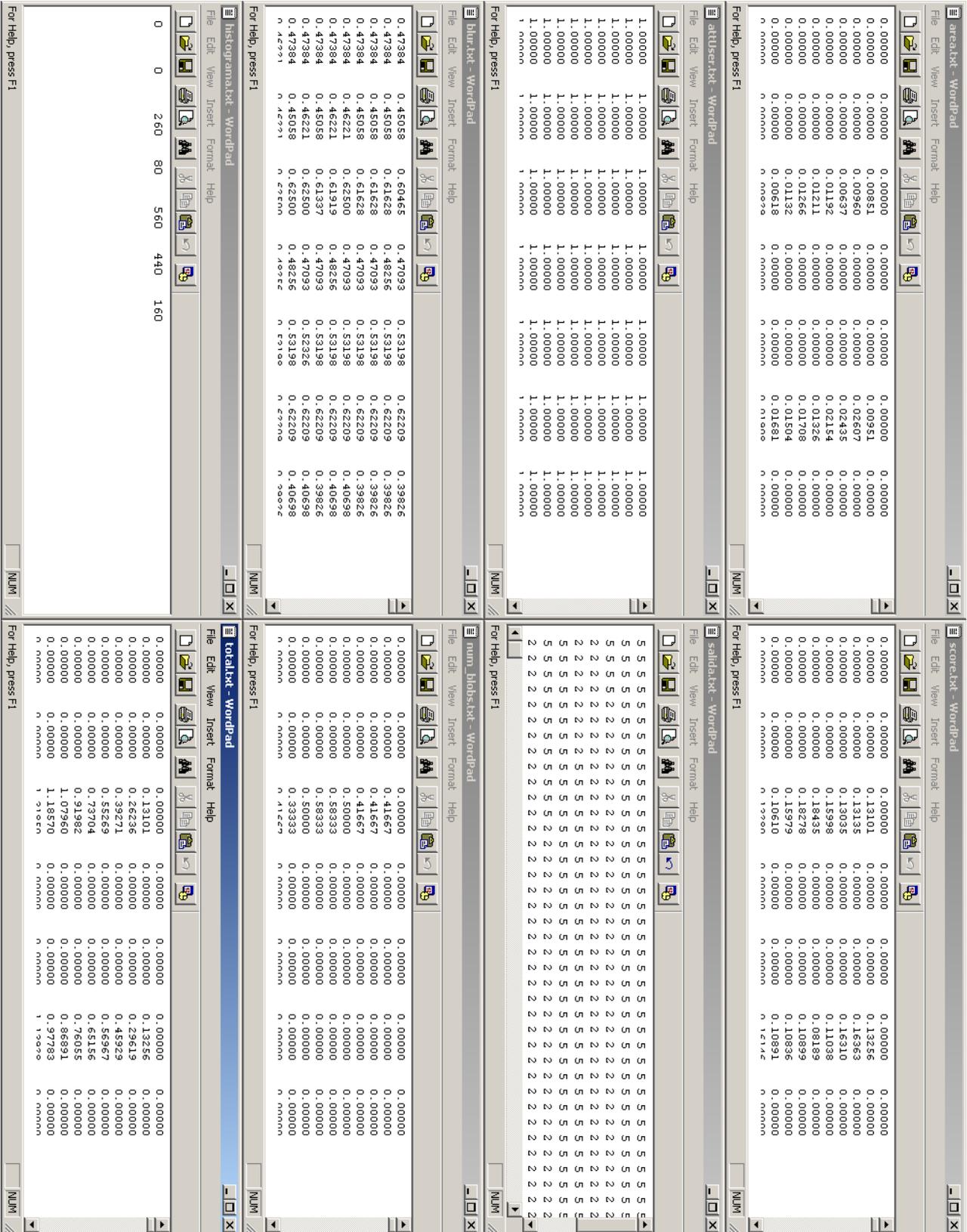


Figura 3.14: Archivos de texto

Capítulo 4

Resultados

4.1. Introducción

A lo largo de la sección anterior se ha explicado el funcionamiento teórico del sistema completo y de sus partes. En esta sección lo que se propone es ver la utilidad de éste programa y poder evaluar los resultados obtenidos del mismo.

Primeramente, se ha analizado el bloque de **QualityBlur** mediante un banco de pruebas basado en imágenes emborronadas artificialmente, el cual se puede ver en A.1, en la página 81. Hecho esto, se procederá al análisis de los demás bloques y sus distintos algoritmos variando sus parámetros, usando los dataset de multicámara disponibles.

Cada una de las pruebas realizadas ha dado como resultado un vídeo montaje, a los cuales se puede acceder en el siguiente link:

- <http://www-vpu.eps.uam.es/publications/AutomaticViewSelectionInMulticameraSystems/>

4.2. Elección de parámetros y sus consecuencias

Tras observar las tablas de parámetros de los distintos dataset, y comparando con la imagen de referencia, se ha decidido usar el valor obtenido de las pruebas con imágenes emborronadas. Para este tipo de aplicación se podrían haber usado otros parámetros siempre que el usuario considere que el orden subjetivo de calidad de imágenes es otro. Esto quiere decir que con cada combinación de parámetros en el bloque **QualityBlur**, se consigue un orden diferente de calidad entre las distintas cámaras, y por lo tanto, el usuario puede elegir el que le parezca más adecuado a su percepción.

En definitiva, se usará para estas pruebas.

- *MinDCTValue* = 10
- *MaxHistValue* = 0,1

Otro de los apartados importantes, es la elección del tamaño de los blobs que el algoritmo reconocerá como tal. Para elegir dicho tamaño, existen dos posibilidades, el usuario podrá insertar los valores que

él considere oportuno para cada una de las cámaras, o podrá cargar unos valores por defecto para cada dataset y cada cámara dentro de éste, que han sido previamente optimizados.

4.2.1. Baloncesto (SET 1)

Primeramente se va a analizar este dataset. Con él se van a realizar distintas simulaciones variando los parámetros y con ello se va a demostrar los diversos efectos que produce la elección de los mismos.

Como se verá a lo largo de la exposición, este dataset se analiza un total de seis veces. En todas ellas los parámetros variarán. A partir de ahora para hablar de esas distintas simulaciones se hablará de SETs.

En esta sección se mostrarán únicamente las figuras más importantes, para visionar las demás ir al apéndice B.



Figura 4.1: Presentación de baloncesto

4.2.1.1. Parámetros elegidos

Los parámetros elegidos en el SET 1, son por así llamarlos, los óptimos. Este set además será el usado como referencia para evaluar el resto de posibles elecciones de parámetros. Esto es una conclusión obtenida de la evaluación de los seis SETs, pero que se expone aquí para mayor claridad.

- Número máximo de blobs (*blobs_max*) = 12
- *Control* = 100
- *AntiFlickering* = 80
- *Retardo* = 40

4.2.1.2. Número de blobs y área

En la figura B.1 se mostrarán la variación del número de blobs a lo largo del vídeo, para cada cámara. Hay que recordar que el coeficiente de número de blobs está normalizado por el número máximo de blobs, así que cualquier variación de éste último provocará una variación en el primero.

También se verá el coeficiente de área, el cual está normalizado por el número de píxeles de la imagen, que es igual para todos los casos, gracias al algoritmo `cvQueryResize` que convierte todos los *frames* a 640×480 .

4.2.1.3. Coeficiente de emborronamiento

Éste coeficiente proviene del bloque **QualityBlur** y depende de los parámetros *MinDCTValue* y *MaxHistValue*. Si no hay variación en ninguno de los dos, el valor de éste coeficiente no variará.

4.2.1.4. Coeficiente de atención

Para interpretar bien la figura B.4 hay que tener claro el funcionamiento del algoritmo: primero se analizan 40 *frames*, y una vez analizados se decide cuál es la salida. Son 40 *frames* porque en este caso ese es el valor del parámetro **Retardo**.

4.2.1.5. Puntuación de las cámaras

Una vez obtenidos los cuatro coeficientes citados anteriormente se procede al cálculo de la puntuación, la cual se plasma en las figuras B.5 y B.6. Una de ellas mostrará la duración del vídeo entero, mientras que la otra solo mostrará una parte, para poder analizar mejor el detalle.

No hay que confundir la puntuación con la salida, puesto que no siempre la máxima puntuación corresponderá con la salida.

4.2.1.6. Puntuación acumulada

Como se ha dicho anteriormente primero se analizan 40 *frames*, porque ese es el valor de **Retardo**. En las siguientes figuras se muestra ese análisis, que no es más que la suma de todas las puntuaciones durante ese intervalo. Al final de dicho intervalo se mirará cuál es la cámara con mayor puntuación, y si la cámara que estaba siendo mostrada anteriormente ya ha cumplido el tiempo mínimo de salida (**AntiFlickering**), se procederá al cambio por la cámara con mayor puntuación acumulada.

4.2.1.7. Histograma de cámaras

Como se puede ver en la figura 4.2, el reparto de los *frames* entre las cámaras es muy desigual.

En ocasiones, como en el caso de la cámara 1, simplemente hay una cámara orientada en la misma dirección pero más cercana (cámara 7), al estar grabando lo mismo, la que esté situada más cerca, obtendrá mayor puntuación. Por lo tanto, se deduce que en este dataset, la elección de las cámaras (sus vistas) no ha sido la óptima.

El caso de la cámara 2 es similar: hay hasta 5 cámaras enfocando a la misma zona (1,2,4,5 y 7).

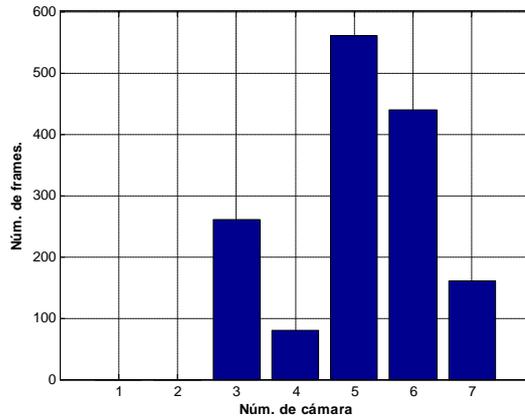


Figura 4.2: Histograma del uso de cámaras.

4.2.1.8. Conclusiones

Estos parámetros dan como resultado el montaje de un vídeo muy bueno, siguiendo en todo momento el transcurso del juego, cambios de cámara progresivos y un tiempo de estancia en la misma cámara adecuado.

En el caso del baloncesto la función del bloque **score** debería cambiar, quitándole importancia al número de blobs en favor del área, ya que se trata de un deporte de contacto, y en un pequeño espacio, por lo que el movimiento de varios jugadores suele superponerse, a veces llevando a engaño, lo que el algoritmo interpreta como 1 blob/jugador en realidad puede ser 3 jugadores o más.



Figura 4.3: Problemas blobs y baloncesto

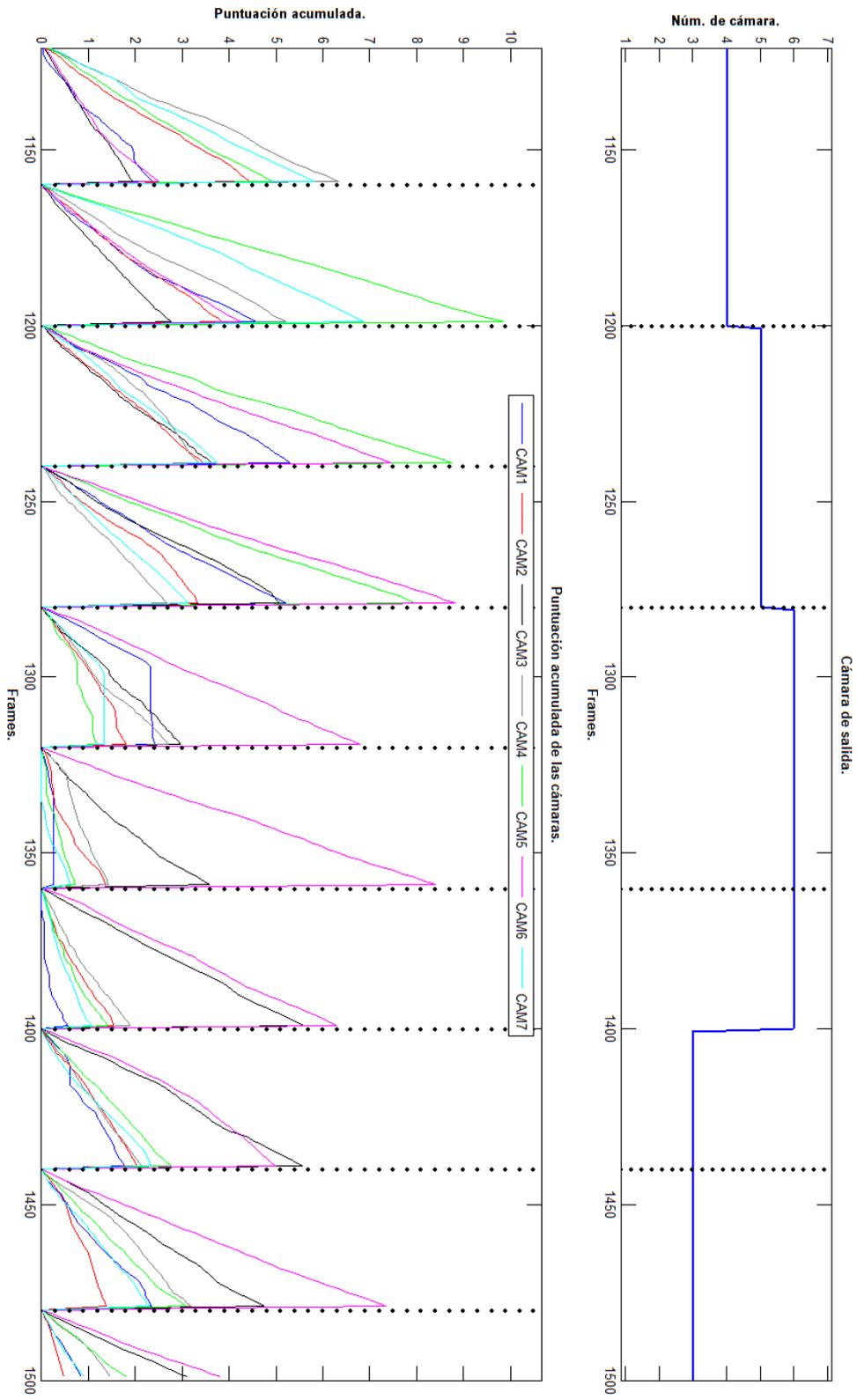


Figura 4.5: Puntuación acumulada (parcial)

4.2.2. Baloncesto (SET 2)

4.2.2.1. Parámetros elegidos

El cambio propuesto en el conjunto de parámetros es la disminución del número máximo de blobs, por debajo del valor esperado, recordemos que en este caso, partido de baloncesto, el número máximo de blobs (número de personas) coincidiría con 10 jugadores más dos árbitros, en total 12. Para esta prueba se decide reducir el parámetro a la mitad del óptimo.

- Número máximo de blobs (*blobs_max*) = 6
- *Control* = 100
- *AntiFlickering* = 80
- *Retardo* = 40

4.2.2.2. Número de blobs y área

Como se puede apreciar en la figura B.7, el hecho de seleccionar un número máximo de blobs más pequeño del valor real que se va a dar en el vídeo, provoca una pérdida de sensibilidad, ya que las veces que se supere ese número (6), el coeficiente se quedará a 1, que es su límite superior. Éste efecto se puede apreciar sobre todo en los 500 primeros *frames* de la cámara 3.

En líneas generales la reducción de éste parámetro supondrá el aumento del coeficiente del número de blobs, ya que se ha reducido el número por el que se normaliza.

El coeficiente de área no variará, ya que este coeficiente engloba todos los píxeles que están en movimiento de *frame* a *frame*, independientemente de los parámetros. Por lo tanto, será el mismo que en el SET 1, el cual se puede ver en la figura B.2.

4.2.2.3. Coeficiente de emborronamiento

El coeficiente de emborronamiento no depende de los parámetros sometidos a evaluación y, por lo tanto, en ninguna de éstas pruebas variará.

La variaciones de este coeficiente se evaluaron a lo largo del apéndiceA.1.

4.2.2.4. Coeficiente de atención

El coeficiente de atención no está directamente relacionado con el número máximo de blobs, sin embargo, como veremos en el siguiente apartado, sí que supone una variación en la puntuación de cada una de las cámaras, cambiando así la salida, lo que conlleva finalmente que éste coeficiente se vea alterado ante el cambio del parámetro de número de blobs.

4.2.2.5. Puntuación de las cámaras

Como se puede apreciar en la figura B.9, la puntuación de las cámaras varía, aumentando con respecto a la referente. Ya se ha dicho que el coeficiente de número de blobs aumentaba, y la puntuación es directamente proporcional a éste.

Se puede ver como en algunos casos la puntuación permanece constante, esto se debe, como se ha dicho anteriormente, a la pérdida de sensibilidad de uno de los coeficientes que participa en la función de cálculo de la puntuación: el coeficiente del número de blobs.

Cabe reseñar, que este problema de sensibilidad, puede llevar a situaciones como la que se puede ver en la cámara 5, donde en tramos del coeficiente de número de blobs igual a 1, se ve claramente la contribución del coeficiente de atención, en torno al *frame* 600 y también al 1100.

4.2.2.6. Puntuación acumulada

La variación del parámetro en cuestión hace que varíe la puntuación, de manera que la puntuación acumulada no será una excepción, lo cual como se ha visto, provoca una alteración en la salida.

4.2.2.7. Histograma de cámaras

Como ya se ha adelantado, se produce un cambio en la salida, que también se pone de manifiesto en el histograma del uso de las cámaras.

En este caso desaparece por completo la utilización de la cámara 4, en favor de la 5.

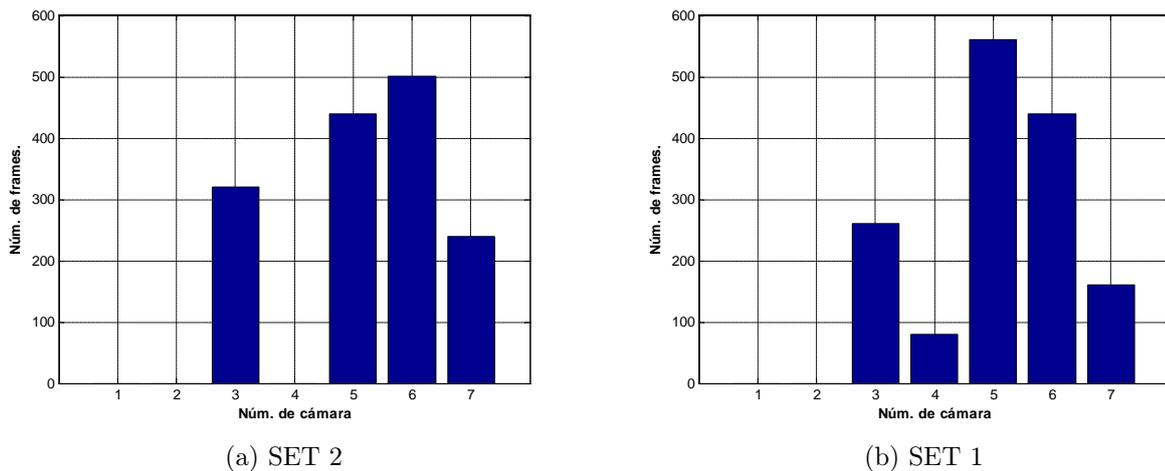


Figura 4.6: Histograma del uso de las cámaras. SET 2 vs SET 1

4.2.2.8. Conclusiones

La elección de un número inferior al esperado en este parámetro (número máximo de blobs), supone una pérdida de sensibilidad en el coeficiente de número de blobs, el cual repercute directamente en la puntuación de las cámaras. Por lo tanto afecta a la elección de las mismas, de tal manera que la decisión final no será la mejor posible.

Una mala elección de este parámetro puede llevar a no obtener la mejor salida posible.

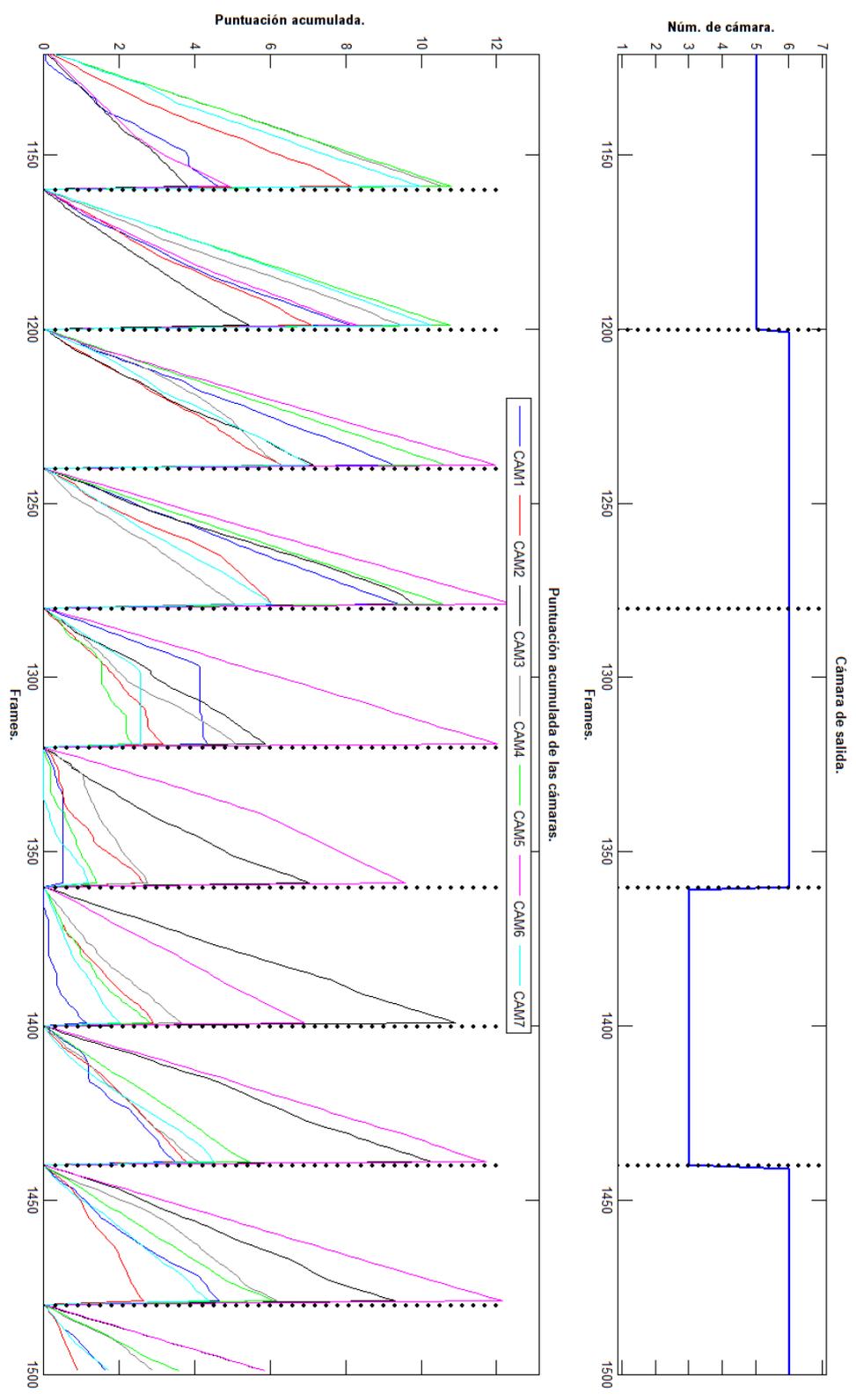


Figura 4.8: Puntuación acumulada (parcial). SET 2 vs SET 1

4.2.3. Baloncesto (SET 3)

4.2.3.1. Parámetros elegidos

En este set se va a analizar el caso opuesto al anterior, sobrepasar el número máximo de blobs esperado, si el esperado son 12, se usará el doble, 24.

- Número máximo de blobs ($blobs_max$) = 24
- $Control$ = 100
- $AntiFlickering$ = 80
- $Retardo$ = 40

4.2.3.2. Número de blobs

Al elegir un número mayor, lo que se consigue es que el coeficiente sea menor, ya que el número máximo de blobs es el encargado de normalizar este coeficiente. Sin embargo, la reducción de éste afectará a la puntuación obtenida en cada cámara.

4.2.3.3. Coeficiente de atención

Como se ha podido observar la salida en esta situación es exactamente igual. No obstante el cambio que provoca el coeficiente de número de blobs en la puntuación de las cámaras, tiene a su vez un efecto en el coeficiente de atención.

En la figura B.12, se ve que el coeficiente de atención no varía, salvo en dos excepciones, en la cámara 4 y 5. Esto se debe a la reducción de la puntuación.

Como se explicó en 3.2.4.7, hay un mecanismo que resetea el coeficiente de atención, si $Score_{nuevo\ frame} \geq 0,2 \cdot Score_{frame\ inicial} + Score_{frame\ inicial}$, donde el $frame$ inicial es en el que empezó a mostrarse esa cámara. Debido a que el valor de la puntuación es menor con respecto al valor que hay en el usado como referencia, puede ocurrir que un cambio en la puntuación, suponga en una un incremento del 20% mientras que en la referente no.

Para mayor claridad se muestran los datos en el cuadro, dichos datos corresponden a la cámara 5.

<i>Frames</i>	Actual			Referencia		
	<i>CoefAttention</i>	<i>Score</i>	Límite	<i>CoefAttention</i>	<i>Score</i>	Límite
1002	1.00000	0.11780	0.14136	1.00000	0.23044	0.276528
...	
1045	0.99536	0.15135	0.18162	0.99536	0.27468	
1046	1.00000	0.14055		0.99517	0.27439	

*** $Límite = 0,2 \cdot Score_{frame\ inicial} + Score_{frame\ inicial}$

Cuadro 4.1: Reinicio Coef. atención

Tras ver el cuadro se comprueba que en el caso actual, la puntuación en el *frame* 1045 supone un incremento superior al 20 % de la puntuación inicial (denotado por la variable Límite), lo cual significa que el coeficiente de atención será reiniciado. Sin embargo, en el caso de referencia no se llega a ese valor del 20 % por lo que no se producirá ningún reinicio.

Este cambio es provocado por la alteración del parámetro de número de blobs, ya que provoca una reducción general del valor del coeficiente de núm. de blobs, y éste a su vez una reducción general en el valor de la puntuación.

4.2.3.4. Puntuación de las cámaras

Estas imágenes ponen de manifiesto la contribución del coeficiente de número de blobs en el cálculo de la puntuación. Se aprecia como una disminución del primero causa, efectivamente, una disminución del segundo.

La diferencia entre ambas funciones es proporcional al valor del coeficiente en cada instante, sin embargo en la cámara 5 se puede apreciar claramente el efecto que provoca el distinto coeficiente de atención, explicado en el punto anterior.

4.2.3.5. Puntuación acumulada

La puntuación acumulada se ve alterada, si bien el máximo valor corresponderá a la misma cámara que en la referente, puesto que no varía la salida. Pero el rango de valores es más reducido, en torno a la mitad.

4.2.3.6. Histograma de cámaras

La salida no ha variado con este cambio de parámetros, por lo tanto el histograma de uso será el mismo que en el caso referente.

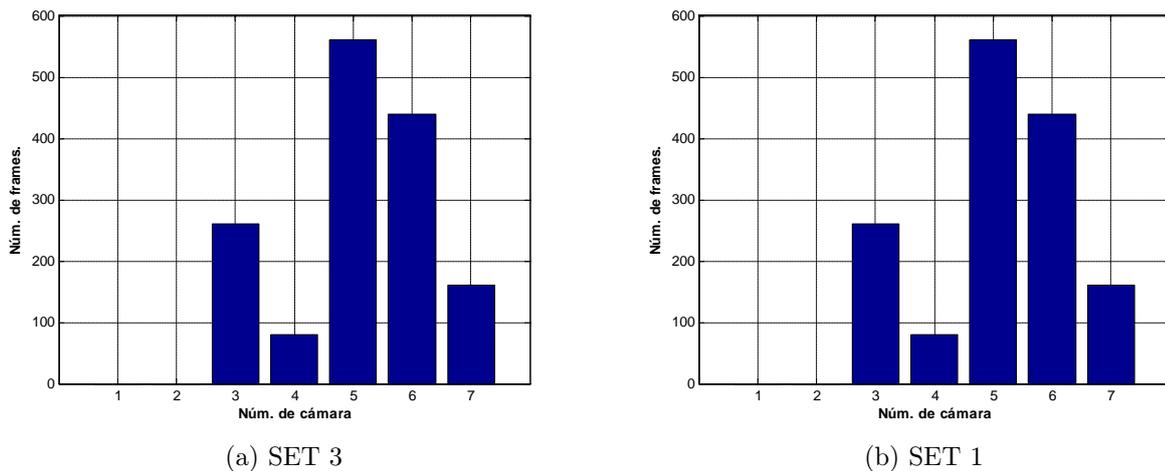


Figura 4.9: Histograma del uso de las cámaras. SET 3 vs SET 1

4.2.3.7. Conclusiones

La principal consecuencia del incremento en el parámetro, es la disminución del valor de la puntuación. Al reducir este valor se pueden dar casos de reinicio del coeficiente de atención, como se ha visto en el punto 4.2.3.3.

En definitiva, sobredimensionar el valor de número máximo de blobs genera diferencias con respecto al referente, pero las diferencias son reducidas. No obstante, aunque en el caso de ejemplo no suponen un cambio sustancial, pueden llegar a producir cambios en la salida.

Habiendo comprobado previamente el caso contrario (reducción del número máximo de blobs), parece preferible el exceso.

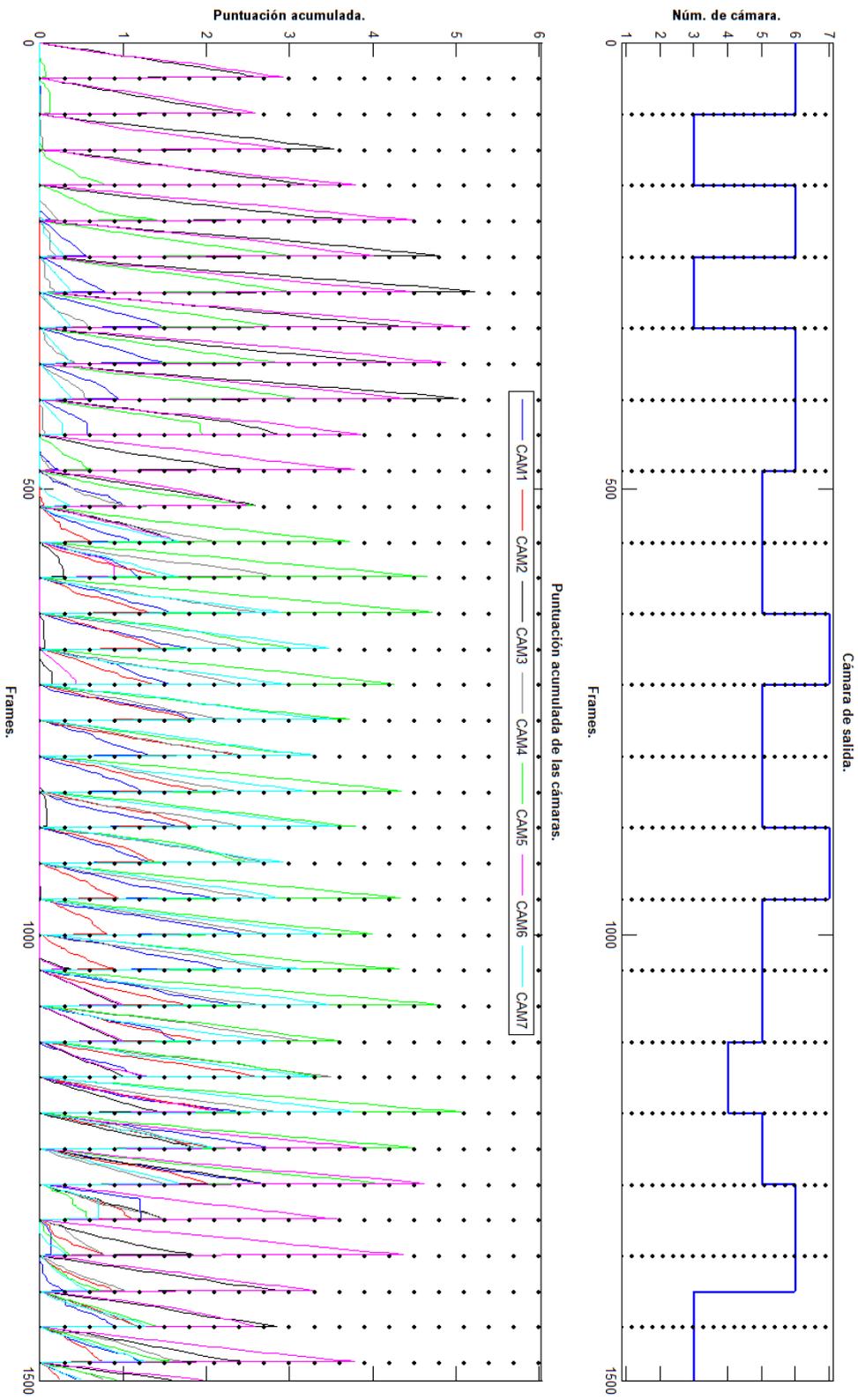


Figura 4.10: Puntuación acumulada (vídeo completo), SET 3 vs SET 1

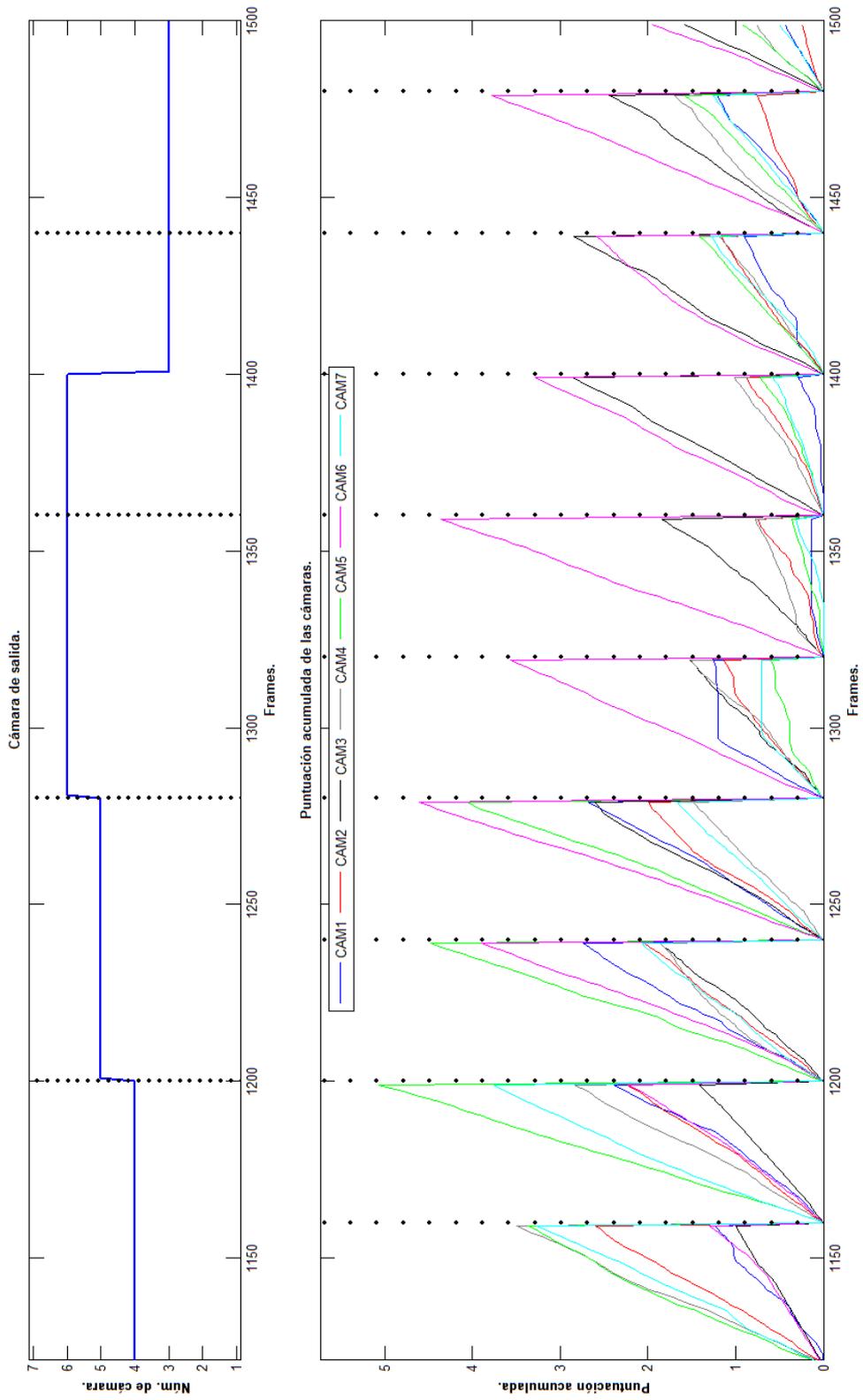


Figura 4.11: Puntuación acumulada (parcial). SET 3 vs SET 1

4.2.4. Baloncesto (SET 4)

4.2.4.1. Parámetros elegidos

En este análisis se va a reducir el valor del parámetro **Control**, el cual es el encargado de determinar a partir de que *frame* se comenzará a penalizar la puntuación de los siguientes.

En el caso elegido como referente, el valor seleccionado para este parámetro es 100. Para más información consultar la sección 3.2.4.7.

- Número máximo de blobs (**blobs_max**) = 12
- **Control** = 40
- **AntiFlickering** = 80
- **Retardo** = 40

En esta situación los coeficientes de número de blobs, área y emborronamiento permanecerán constantes puesto que la variable **Control** actúa directa y únicamente sobre el coeficiente de atención.

4.2.4.2. Coeficiente de atención

El parámetro **Control** determina cuál será la forma de la función (de la arcotangente), si su pendiente es más o menos pronunciada, y por lo tanto la penalización que sufre cada *frame*.

Al ver la figura B.15 se comprueba, sobre todo en los primeros *frames* donde la salida es la misma que en el referente, que la penalización provocada por el parámetro **Control** = 40 es mayor, provocando en algunos casos el cambio de cámara con respecto a la referente.

Al penalizar más rápido los *frames*, se promueve un cambio de cámara más rápido, como se aprecia en la imagen. El número de cambios de cámara ha aumentado, llegando al máximo, como en este caso donde se realizan cada "**Retardo**" *frames*, donde **Retardo** es el número mínimo de *frames* a analizar para que se pueda producir un cambio, ver 3.2.6.

4.2.4.3. Puntuación de las cámaras

Los cambios en la puntuación se deben únicamente al cambio del parámetro **Control**. El coeficiente de atención disminuye más rápido que en el caso de referencia, provocando que los *frames* mostrados pierdan importancia en favor de los novedosos.

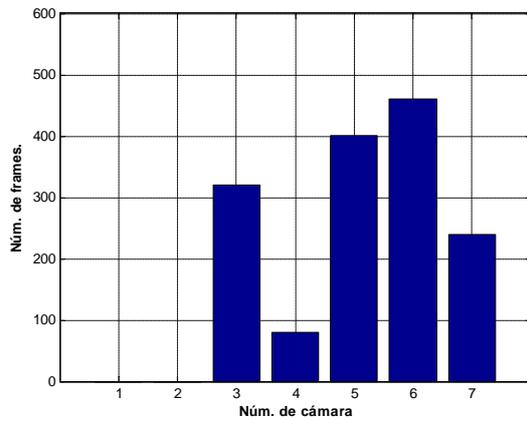
4.2.4.4. Puntuación acumulada

La puntuación acumulada es del mismo orden que la referente. La escasa variación que hay entre ambas situaciones se debe únicamente a la acción del coeficiente de atención.

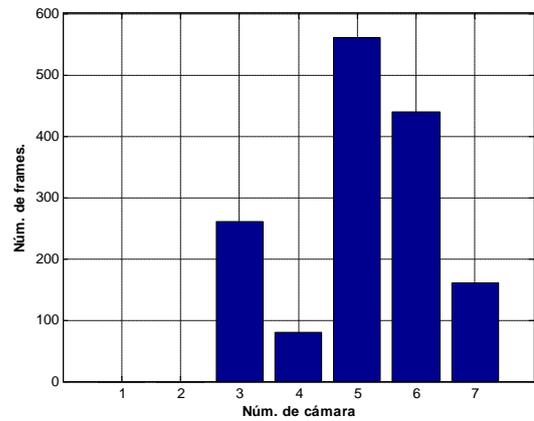
4.2.4.5. Histograma de cámaras

Debido a la alta penalización, hay un mayor cambio de cámara, produciéndose así mayor repartición de la salida entre las distintas cámaras, como indica el histograma.

Las diferencias entre las 3 cámaras más mostradas se ve notablemente reducida.



(a) SET 4



(b) SET 1

Figura 4.12: Histograma del uso de las cámaras. SET 4 vs SET 1

4.2.4.6. Conclusiones

El cambio del parámetro **Control**, permite realizar montajes con tránsitos más o menos cortos, según se elija menor o mayor, respectivamente.

Cuanto más pequeño sea el valor elegido, menor será el tiempo de estancia en una cámara, llegando al mínimo de “**Retardo**” frames. Esto se debe a que cuanto más pequeño sea el parámetro antes se empezará a penalizar los frames.

Por otro lado, un valor muy grande supone no penalizar apenas los frames. Se produciría en una situación en la que el usuario permanecería igual de atento en todo momento.

Según lo buscado por el usuario en cada montaje de vídeo, este parámetro nos dará la posibilidad de tránsitos largos o lentos.

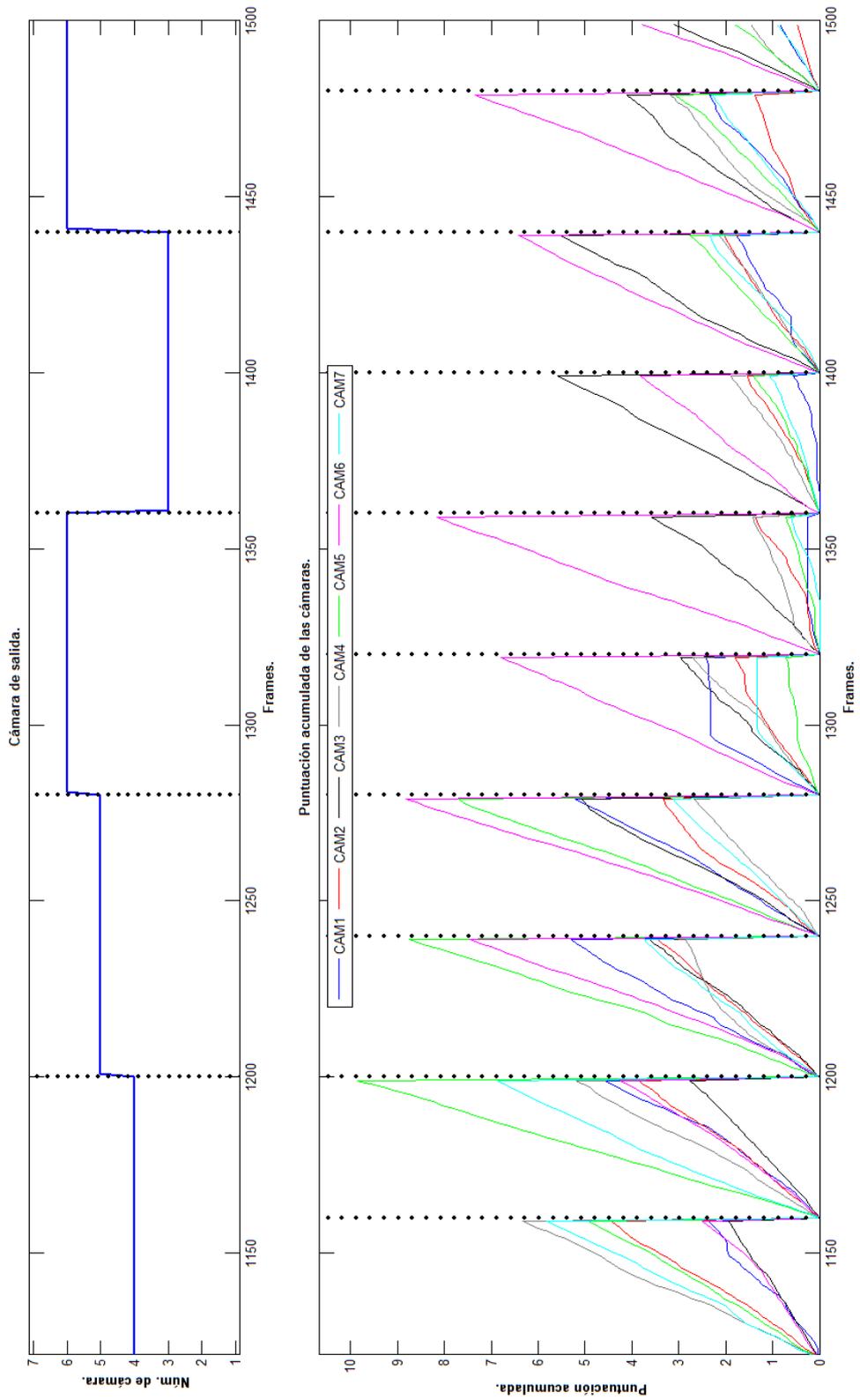


Figura 4.14: Puntuación acumulada (parcial), SET 4 vs SET 1

4.2.5. Baloncesto (SET 5)

4.2.5.1. Parámetros elegidos

Uno de los problemas que surgen a la hora de hacer un montaje de vídeo es encontrar una relación adecuada entre el mínimo tiempo que una cámara ha de ser mostrada, y el tiempo necesario para tomar una decisión sobre que cámara mostrar.

En este algoritmo esos dos tiempos se ven representado mediante los parámetros *AntiFlickering* y *Retardo*, respectivamente.

Con este SET se pretende ver un ejemplo práctico de lo explicado teóricamente en la sección 3.2.6.3.

- Número máximo de blobs (*blobs_max*) = 12
- *Control* = 100
- *AntiFlickering* = 80
- *Retardo* = 70

4.2.5.2. Coeficiente de atención

El coeficiente de atención va variando según la cámara que vaya saliendo. En este caso, independientemente de la puntuación, la cámara elegida en primer lugar saldrá un mínimo de 140 frames. Esto se debe a que el mínimo tiempo de estancia en una misma cámara lo denota el parámetro *AntiFlickering* (en este caso 80), pero por otro lado el parámetro *Retardo* indica cuantos *frames* analiza el algoritmo para tomar una decisión. De tal manera que cuando el algoritmo analiza los primeros 70 *frames* y toma una decisión sobre la cámara que saldrá aún no ha cumplido el tiempo mínimo de estancia, por lo que la siguiente cámara en salir será la misma que acaba de ser elegida, para así cumplir el *AntiFlickering*. Por lo tanto, el primer cambio de cámara se podrá producir, ahora sí, dependiendo únicamente de la puntuación acumulada, en el *frame* 140.

Como se aprecia en la función de salida, hay muchos menos cambios de cámara y el tiempo de estancia es mayor.

Los cambios en el coeficiente de atención se deben a la alteración en los tiempos mínimos de estancia en una misma cámara, que provoca que una cámara esté como salida más tiempo por lo que será más penalizada por el coeficiente de atención.

4.2.5.3. Puntuación de las cámaras

Las variaciones apreciadas en la puntuación con respecto a la referente, se deben únicamente a la acción del coeficiente de atención.

Hay que destacar que, pese a los escasos cambios que hay en la puntuación, la función de salida es muy diferente a la de referencia. Esto es debido a la relación existente entre los parámetros *AntiFlickering* y *Retardo*.

4.2.5.4. Puntuación acumulada

Analizando la figura se observa el efecto comentado anteriormente, entre *Retardo* y *AntiFlickering*. En la segunda puntuación acumulada, *frame* 140, se observa que la cámara con mayor puntuación es la 3. Sin embargo, la mostrada en la salida sigue siendo la cámara 6. Este efecto se puede seguir viendo a lo largo de todo el vídeo.

4.2.5.5. Histograma de cámaras

Analizando este caso, la cámara 4 corresponde a un periodo de 80 *frames* en el caso referente, es decir, que fue seleccionada una vez como la más importante, pero en el caso actual no aparece. Esto se debe a lo citado anteriormente, que la cámara elegida ha de mostrarse durante 140 *frames* para cumplir el mínimo tiempo de estancia, ignorando durante el segundo periodo de 70 *frames* la cámara que haya sido más importante. Por lo tanto, es más probable que las cámaras más usadas sean las que se “beneficien” de este efecto, y así se incrementa más su uso en detrimento de las menos usadas, como en este caso la cámara 4.

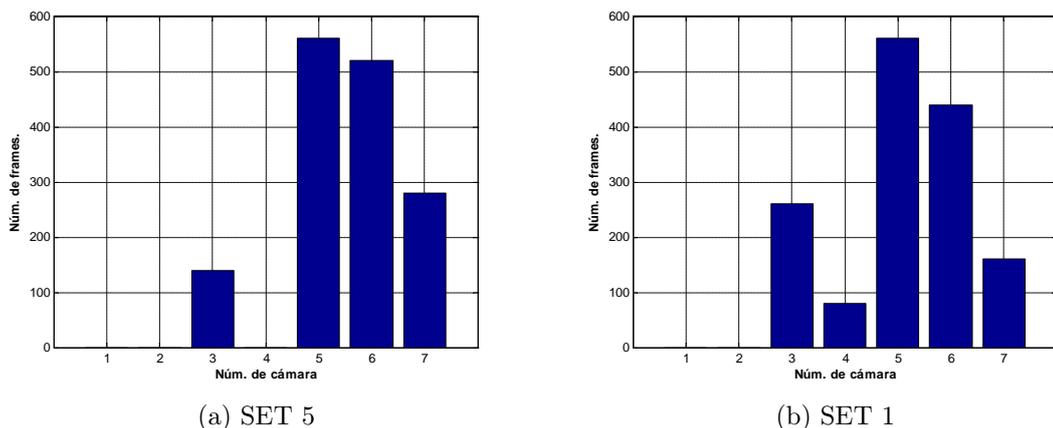


Figura 4.15: Histograma del uso de las cámaras. SET 5 vs SET 1

4.2.5.6. Conclusiones

La elección de un valor de *Retardo* cercano al valor de *AntiFlickering* es un error, ya que aumenta el tiempo mínimo de estancia en una cámara al doble del valor de *Retardo*, haciendo inútil el valor de *AntiFlickering*.

Esta situación provoca pérdida de información, ya que para cumplir ese tiempo mínimo ha de mostrar una misma cámara sea o no sea la más importante en ese momento.

Por otro lado, se ve cómo incrementa la utilización de las cámaras con mayor índice de uso, perdiendo así distintas vistas, que según el algoritmo eran calificadas de importantes (puntuación alta en ciertos momentos).

En conclusión una relación de $Retardo \lesssim AntiFlickering$ no es adecuada para la realización de un montaje de vídeo.

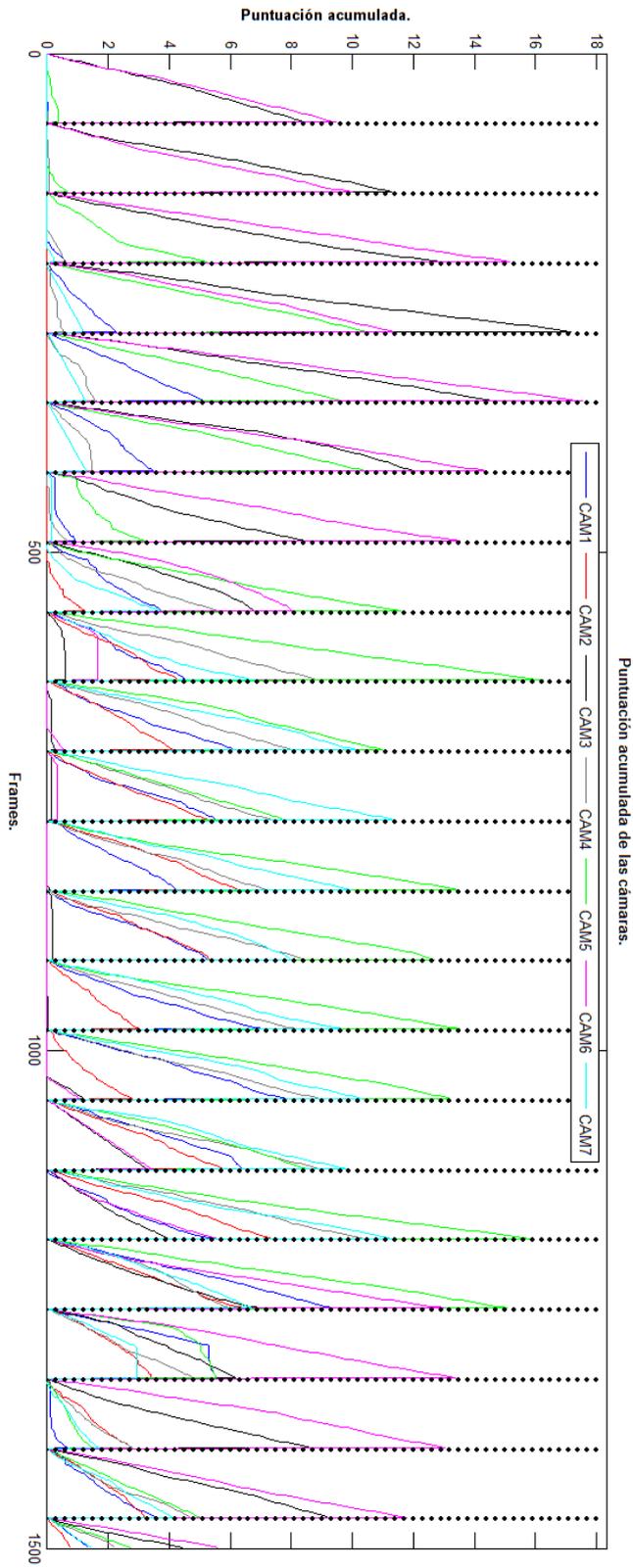
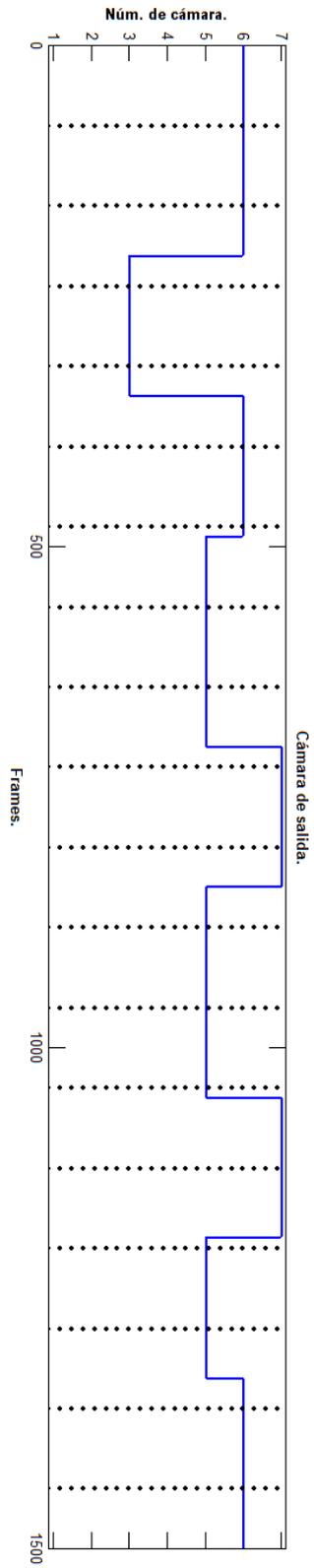


Figura 4.16: Puntuación acumulada (vídeo completo). SET 5 vs SET 1

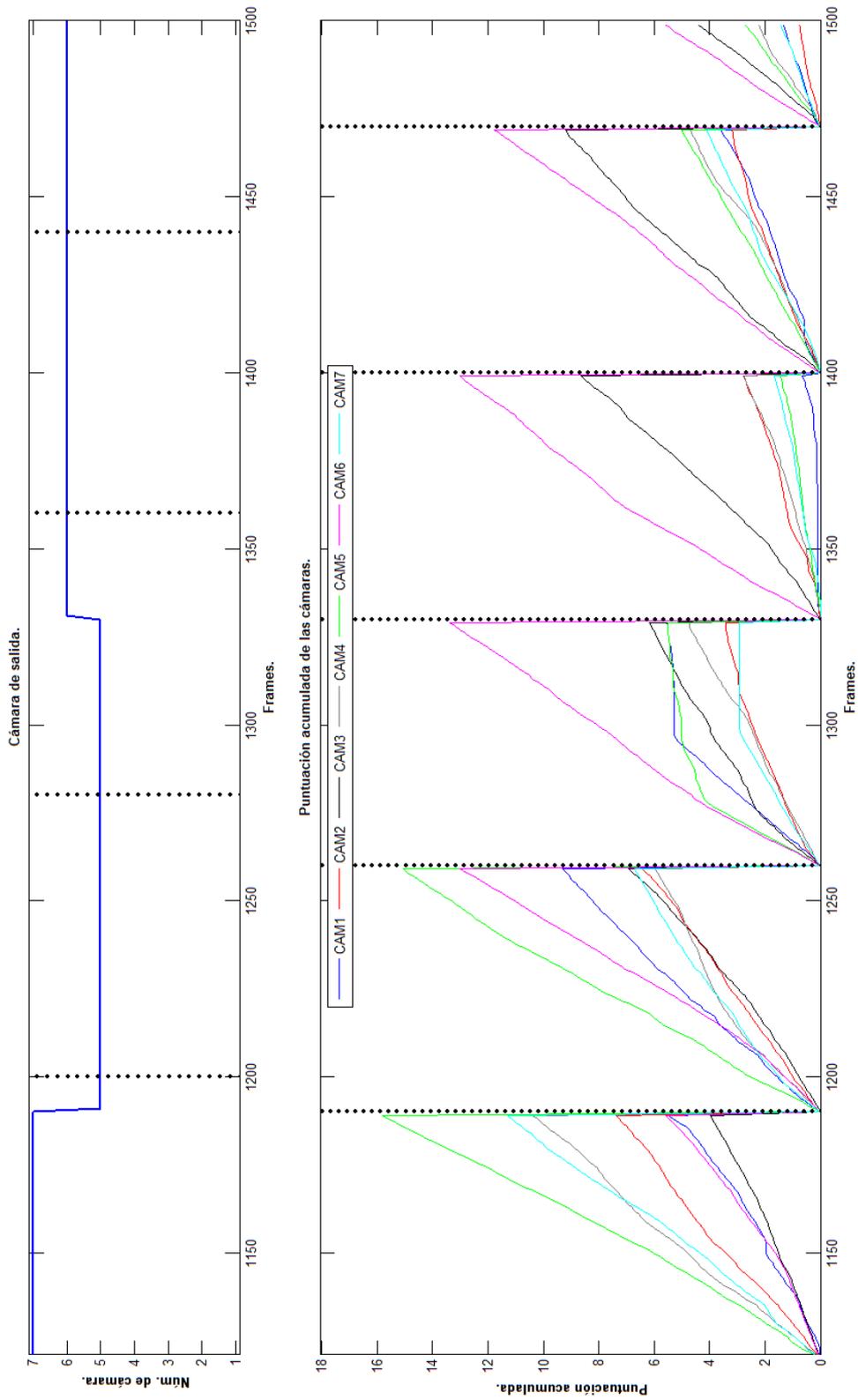


Figura 4.17: Puntuación acumulada (parcial), SET 5 vs SET 1

4.2.6. Baloncesto (SET 6)

4.2.6.1. Parámetros elegidos

En el caso referente se ha estudiado la relación $2 \cdot \text{Retardo} = \text{AntiFlickering}$, y en el caso anterior $\text{Retardo} \approx \text{AntiFlickering}$. Ahora se va a hacer aún más pequeño el **Retardo**, usando una relación $3 \cdot \text{Retardo} = \text{AntiFlickering}$. Para ello se ha cambiado el valor de **AntiFlickering** a 90, para que sea múltiplo de 3.

- Número máximo de blobs (**blobs_max**) = 12
- **Control** = 100
- **AntiFlickering** = 90
- **Retardo** = 30

Hay que tener cuidado al realizar la comparación con la referente, ya que se han cambiado tanto **AntiFlickering** como **Retardo**, y no solo uno de ellos.

4.2.6.2. Coeficiente de atención

Aparecen cambios en el coeficiente de atención con respecto a la referente, debido a que el tiempo mínimo de estancia en una misma cámara ha cambiado, ha pasado de ser 80 *frames* a 90, lo que introduce cambios en este coeficiente.

Por otro lado también habrá cambios, ya que al variar el parámetro **Retardo**, las decisiones sobre qué cámara será la salida en cada momento cambiarán, haciendo así que el coeficiente también varíe.

4.2.6.3. Puntuación de las cámaras

La única variación que se producirá en la puntuación será la correspondiente al coeficiente de atención, puesto que los demás coeficientes se mantienen en los mismos valores que en el caso referencia.

4.2.6.4. Puntuación acumulada

Al elegir valores tan pequeños de **Retardo**, se le da más importancia de la debida a los cambios puntuales de las cámaras. Ya que en los 30 *frames* elegidos en este ejemplo como **Retardo**, puede que una cámara tenga mucha puntuación, pero en los 90 totales en los que va a ser mostrada, esa importancia se diluya.

Este efecto, se puede apreciar levemente en torno al *frame* 1000, cuando la salida corresponde a la cámara 7, esta decisión ha sido tomada ya que los 30 primeros *frames* analizados dan una alta importancia a la cámara 7, pero en los dos tramos siguientes de 30 *frames* (hasta 90) se observa que la cámara más importante es la 5.

Sin duda alguna una elección algo más alta del valor **Retardo** permite reducir estos casos.

4.2.6.5. Histograma de cámaras

Como se puede apreciar en el histograma, las cámaras minoritarias han aumentado su uso, esto se debe en mayor medida a la situación descrita con anterioridad, a menor valor del parámetro *Retardo*, más posibilidades tienen de ser elegidas como salida, porque aunque en un intervalo mayor no sean tan importantes, puede que en un margen menor, 30 frames como el del ejemplo, sí sean de gran importancia.

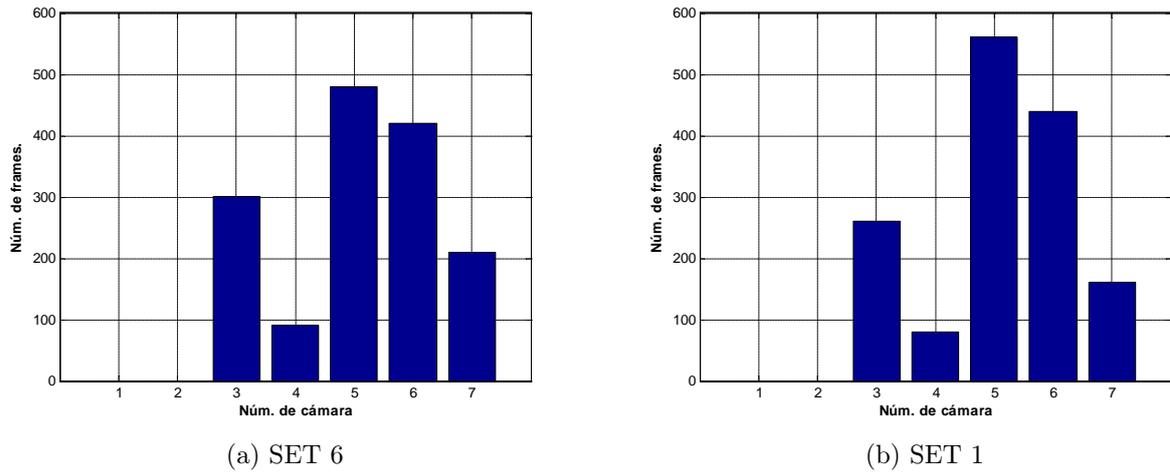


Figura 4.18: Histograma del uso de las cámaras. SET 6 vs SET 1

4.2.6.6. Conclusiones

La elección de la relación entre *AntiFlickering* y *Retardo* es muy importante, como se ha visto a lo largo de los ejemplos y en el caso referente. Un valor muy pequeño como el caso de este ejemplo, puede ocasionar que se elijan cámaras, que en el rango que son mostradas no sean tan importantes, pero que al ser analizadas en el intervalo denotado por *Retardo* sí lo eran.

Una elección errónea de esta relación puede ocasionar un mal montaje del vídeo de salida.

Por un lado hay que evitar lo que pasaba en el SET 5, un exceso de tiempo mínimo de estancia, pero sin pasar al extremo como pasa en este caso. El punto medio entre ambos es el caso referente, donde la relación entre ambos parámetros es $2 \cdot \text{Retardo} = \text{AntiFlickering}$.

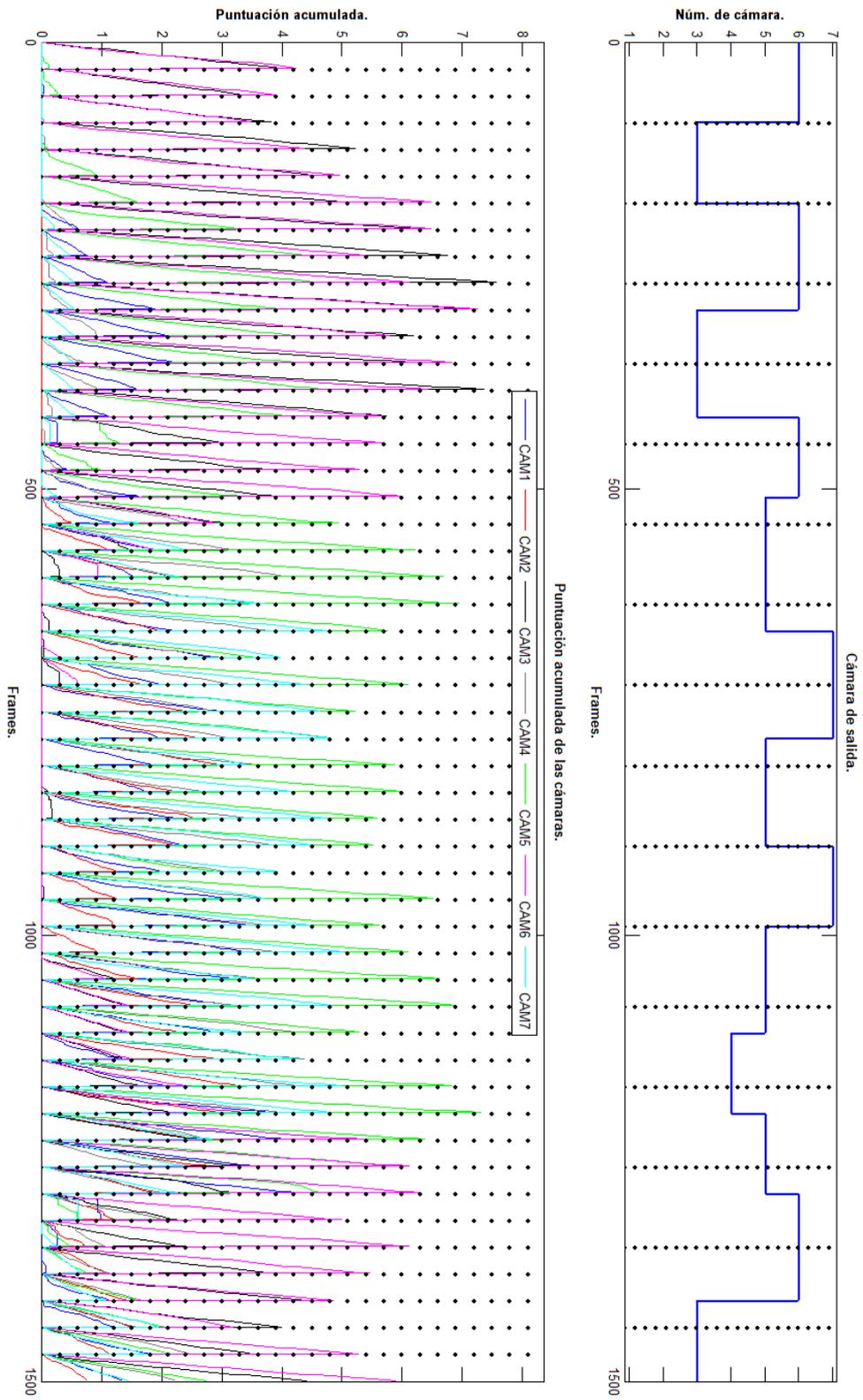


Figura 4.19: Puntuación acumulada (vídeo completo). SET 6 vs SET 1

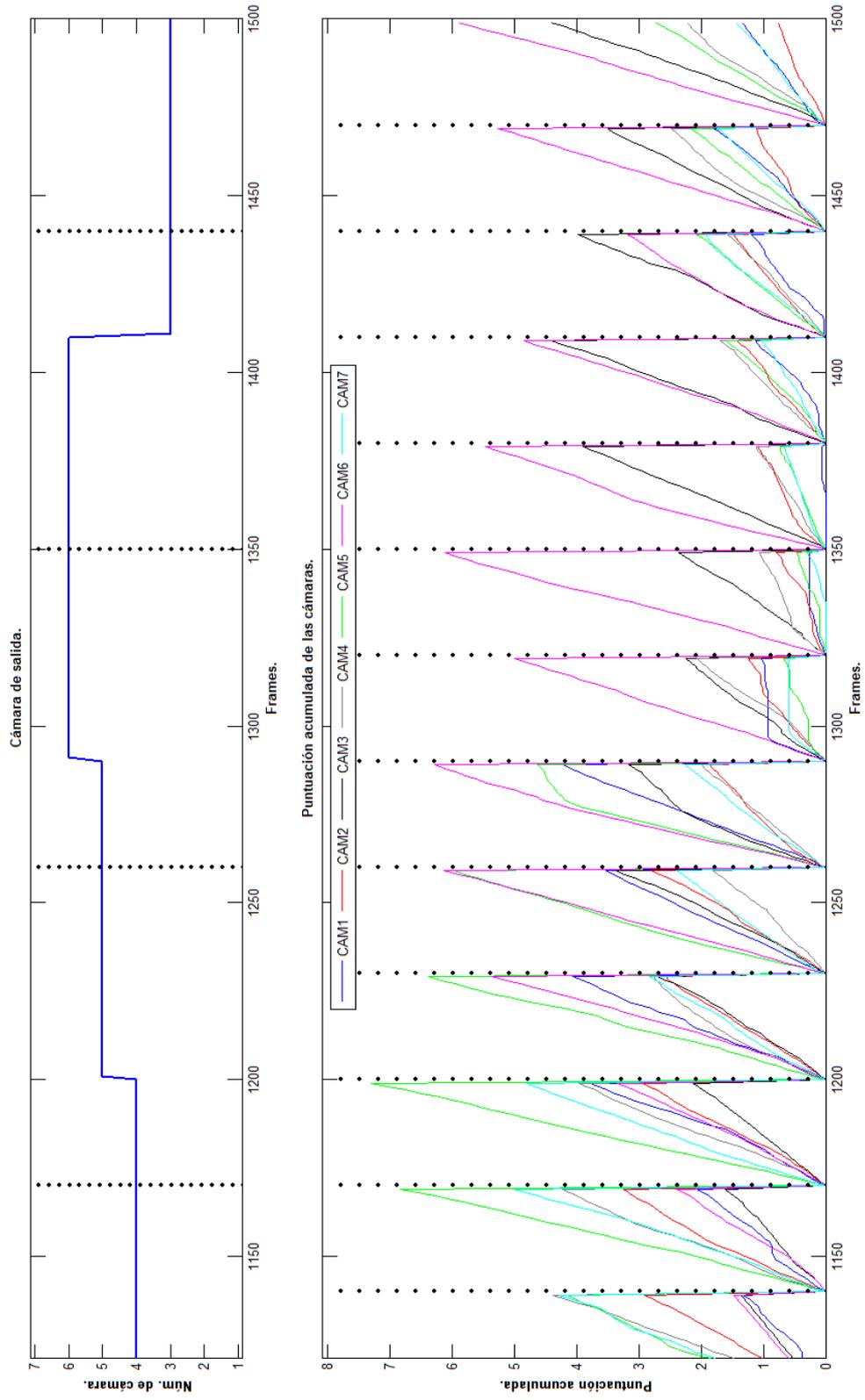


Figura 4.20: Puntuación de las cámaras (parcial). SET 6 vs SET 1

4.2.7. Conclusiones finales sobre los parámetros

De estas seis pruebas realizadas con distintos parámetros se pueden sacar las siguientes conclusiones:

- Una elección errónea del número máximo de blobs (*blob_max*) supone un falseamiento de la variable score, lo cual repercute directamente en la elección de cámara impidiendo la realización del montaje más adecuado.
- La variable *Control* es responsable, en parte, de la velocidad de transición entre cámaras, ya que representa a partir de que frame empieza a aburrirse el espectador. Un número muy pequeño producirá transiciones muy rápidas, mientras que uno muy grande generará transiciones más lentas.
- Por último la relación entre *AntiFlickering* y *Retardo*.
 - Si el *Retardo* \ll *AntiFlickering*, se selecciona la cámara evaluando un conjunto pequeño de frames lo cual implica que se pueda elegir una cámara muy importante en ese pequeño intervalo pero poco importante para el intervalo de frames durante el que va a ser mostrada (*AntiFlickering*)
 - Si el *Retardo* \lesssim *AntiFlickering*, la misma cámara será mostrada como mínimo el doble del valor de *Retardo*, independientemente de cual sea la cámara más importante durante ese intervalo.
 - Si el *Retardo* $= \frac{1}{2}$ *AntiFlickering* supone un compromiso intermedio entre los dos problemas citados anteriormente.

4.3. Resultados para distintos dataset

Como se ha dicho y argumentado a lo largo de todo el punto anterior el set elegido como óptimo será el siguiente:

- Número máximo de blobs (*blobs_max*) = 12
- *Control* = 100
- *AntiFlickering* = 80
- *Retardo* = 40

A continuación, se pasará a realizar las pruebas del sistema completo con dichos parámetros, sobre los tres diferentes dataset de los que se dispone, cambiando únicamente el número máximo de blobs.

4.3.1. Baloncesto

Este dataset ha sido analizado a lo largo del punto anterior, tanto para el set de parámetros elegido como para los demás que han compuesto el banco de pruebas.

4.3.2. Fútbol

Este dataset cuenta con 6 cámaras distribuidas por todo el campo, 3 en cada banda de forma que cubran todas las zonas del terreno de juego.

Para este set de cámaras únicamente se va a hacer el análisis con los parámetros elegidos como óptimos, por las pruebas realizadas en el anterior dataset.



Figura 4.21: Presentación de fútbol

4.3.2.1. Parámetros elegidos

El único cambio realizado será en el número máximo de blobs, puesto que en este deporte el número de jugadores cambia.

- Número máximo de blobs (*blobs_max*) = 25
- *Control* = 100
- *AntiFlickering* = 80
- *Retardo* = 40

4.3.2.2. Número de blobs y área

Tanto en área como en número de blobs, se aprecia que las cámaras con mayor movimiento son la 3 y la 4, ya que es donde más tiempo se desarrolla el juego en un partido de fútbol.

4.3.2.3. Coeficiente de emborronamiento

En general se trata de una calidad de vídeo bastante mala. La media del coeficiente de emborronamiento se encuentra por debajo del 30 %, lo cual provocará unas puntuaciones bajas.

Hay que tener cuidado con los dataset que tengan mucha diferencia entre los coeficientes de emborronamiento de cada cámara, ya que puede provocar que una cámara con contenido importante se pierda en el montaje por falta de calidad. Por lo tanto, es importante que las cámaras tengan más o menos el mismo orden de coeficiente de emborronamiento.

4.3.2.4. Coeficiente de atención

Se puede apreciar como las cámaras más penalizadas son aquellas más mostradas.

4.3.2.5. Puntuación de las cámaras

Se aprecia en la figura como las cámaras situadas en los extremos tienen actividad puntual, sin embargo, las cámaras centrales, la 3 y la 4, casi no pasan por cero, ya que como se ha dicho anteriormente, en este deporte siempre hay movimiento en la parte central del campo.

4.3.2.6. Puntuación acumulada

Llama la atención, comparado con el dataset de baloncesto, el valor tan bajo de las puntuaciones acumuladas, esto es producto de la escasa calidad del vídeo, como se ve en el coeficiente de emborronamiento, lo cual reduce su puntuación.

4.3.2.7. Histograma de cámaras

Las cámaras con mayor uso son las 3 y la 4, esto se debe a que son las cámaras que graban el medio del campo que es donde suele haber más movimiento en un partido de fútbol.

También habría que señalar que las cámaras más mostradas corresponden, con las cámaras que tienen mayor calidad.

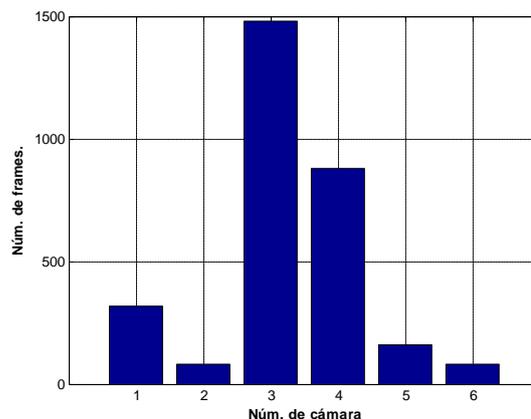


Figura 4.22: Histograma del uso de cámaras

4.3.2.8. Conclusiones

Resulta un montaje adecuado, no obstante hay que resaltar que en un deporte como el fútbol, no siempre es lo más indicado seguir la mayor cantidad de movimiento, ya que en ocasiones no coincide con lo más importante, como por ejemplo en un contraataque.

Es por esto por lo que en la sección de trabajo futuro, 5.2, se propondrán cuestiones como reconocimiento y seguimiento de balón.

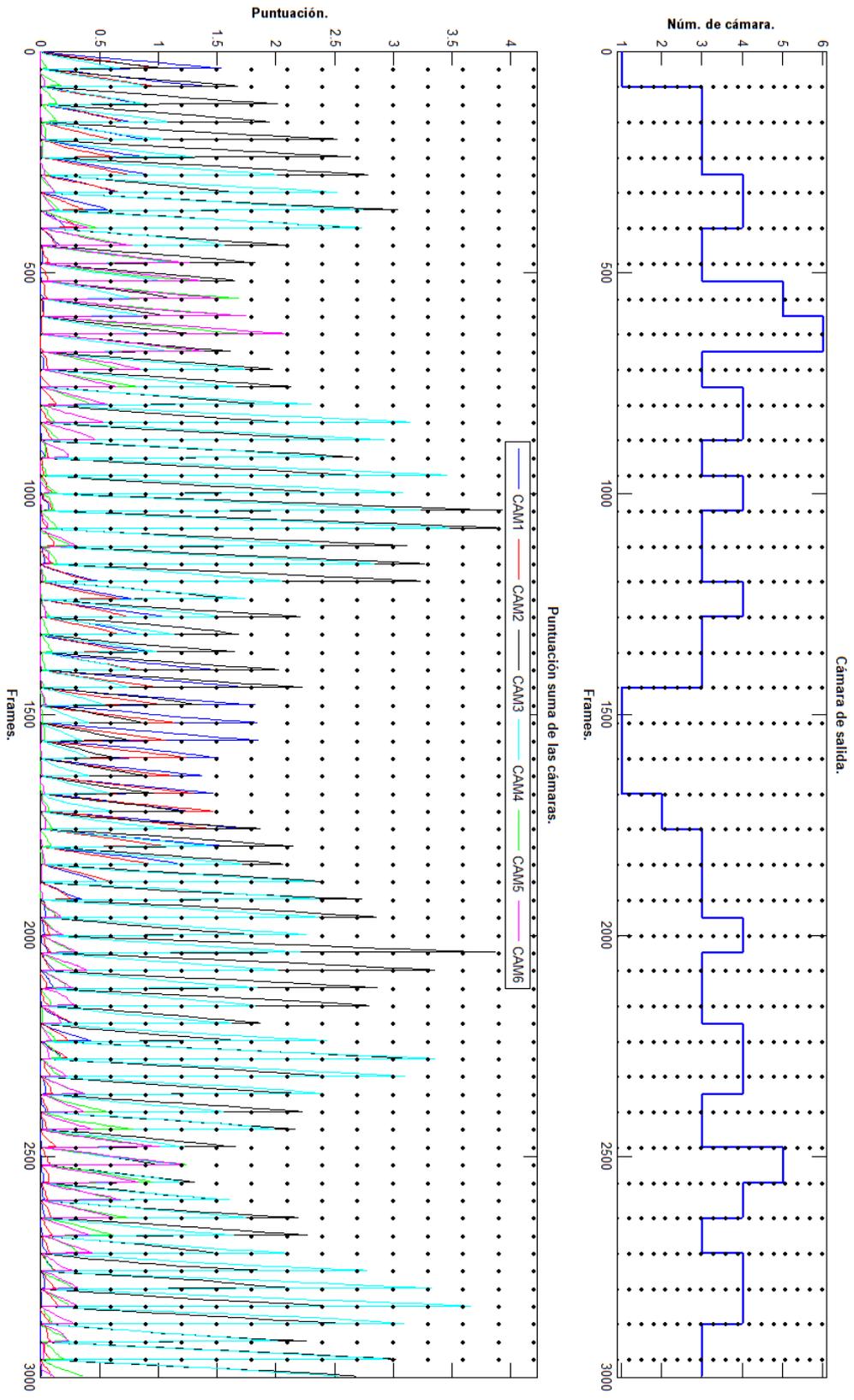


Figura 4.23: Puntuación acumulada (vídeo completo)

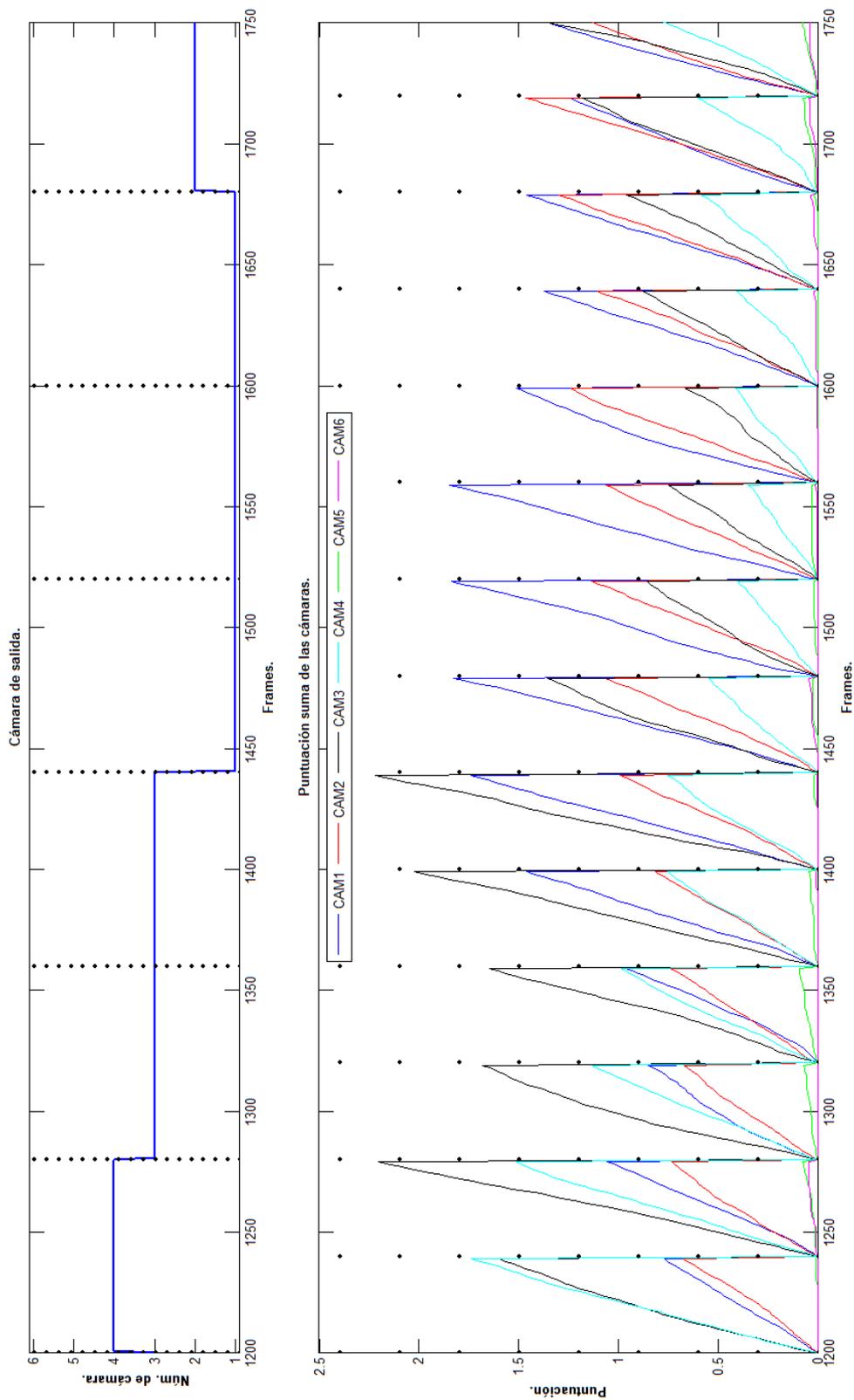


Figura 4.24: Puntuación acumulada (parcial)

4.3.3. Tenis

Dataset formado por un sistema de 9 cámaras distribuídas por todo el terreno de juego.



Figura 4.25: Presentación de tenis

4.3.3.1. Parámetros elegidos

Otra vez, como en el caso del fútbol, el único cambio que se realizará será en el número máximo de blobs.

- Número máximo de blobs (*blobs_max*) = 2
- *Control* = 100
- *AntiFlickering* = 80
- *Retardo* = 40

4.3.3.2. Número de blobs y área

Hay que destacar la forma de la figura de número de blobs, son cambios drásticos entre 0 y 1. Esto se debe a que en el tenis, solo hay dos jugadores, por lo que solo hay tres posibilidades para el coeficiente de número de blobs: que salgan los dos, uno o ninguno. De ahí que se vea tanto salto en esta figura.

4.3.3.3. Coeficiente de emborronamiento

Se aprecia como hay cámaras con mucha más calidad que otras. Esto también se debe a la colocación de las mismas, ya que hay algunas de ellas, que no están colocadas adecuadamente para seguir el juego de la pista de tenis, y que por lo tanto están grabando cosas de poco interés, como es el caso de la cámara 5, que parte de ella enfoca a la lejanía .

4.3.3.4. Coeficiente de atención

Se puede ver que gran cantidad de cámaras mantienen su coeficiente de atención a 1 porque en ningún momento son mostradas.

4.3.3.5. Puntuación de las cámaras

Hay cámaras que pese a tener actividad han obtenido una puntuación baja, esto se debe a la calidad de la imagen.

Otras sin embargo, como el caso de la cámara 3, tienen baja puntuación no solo por la mala calidad, sino también por la mala colocación de la cámara, en la que la mitad del *frame* se ve el final de la pista, sin que haya nada de interés en esa zona.

4.3.3.6. Puntuación acumulada

Después de lo visto en el caso del fútbol, aquí vuelve a subir el valor de la puntuación, en torno al doble de lo que valía en el caso anterior, esto se debe al aumento de la calidad de los *frames*, en comparación.

4.3.3.7. Histograma de cámaras

Por el histograma vemos que de las nueve cámaras hay cuatro que no han sido mostradas ni una sola vez.

En el caso de la cámara 5, por ejemplo, se debe a que no hay apenas actividad en su zona, como se ve representado en el coeficiente de área y en el de número de blobs.

Sin embargo en la cámara 7 el problema no es ese, sino que la calidad de la cámara es muy baja.

La cámara 1 y 3, son una mezcla de las dos anteriores, viéndose así siempre superadas por la puntuación de otras cámaras.

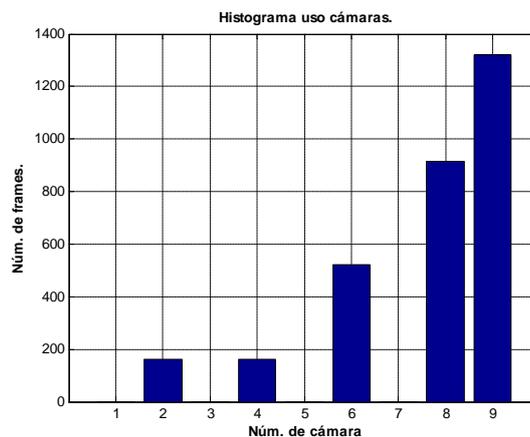


Figura 4.26: Histograma del uso de cámaras

4.3.3.8. Conclusiones

Para el tenis resulta un montaje adecuado, ya que la mayoría de los planos son amplios, y al ser un terreno pequeño se ve todo sin problemas. No obstante, en ciertas ocasiones, la cámara se centra en el jugador que corre y no en el que golpea. Esto se podría resolver usando un detector de pelota.

Hay que destacar que los problemas citados anteriormente se podrían resolver eligiendo una mejor colocación de las cámaras, ya que su posicionamiento es muy deficiente, llegando incluso a casos en los que se ve más proporción de terreno de fuera de pista que de dentro.

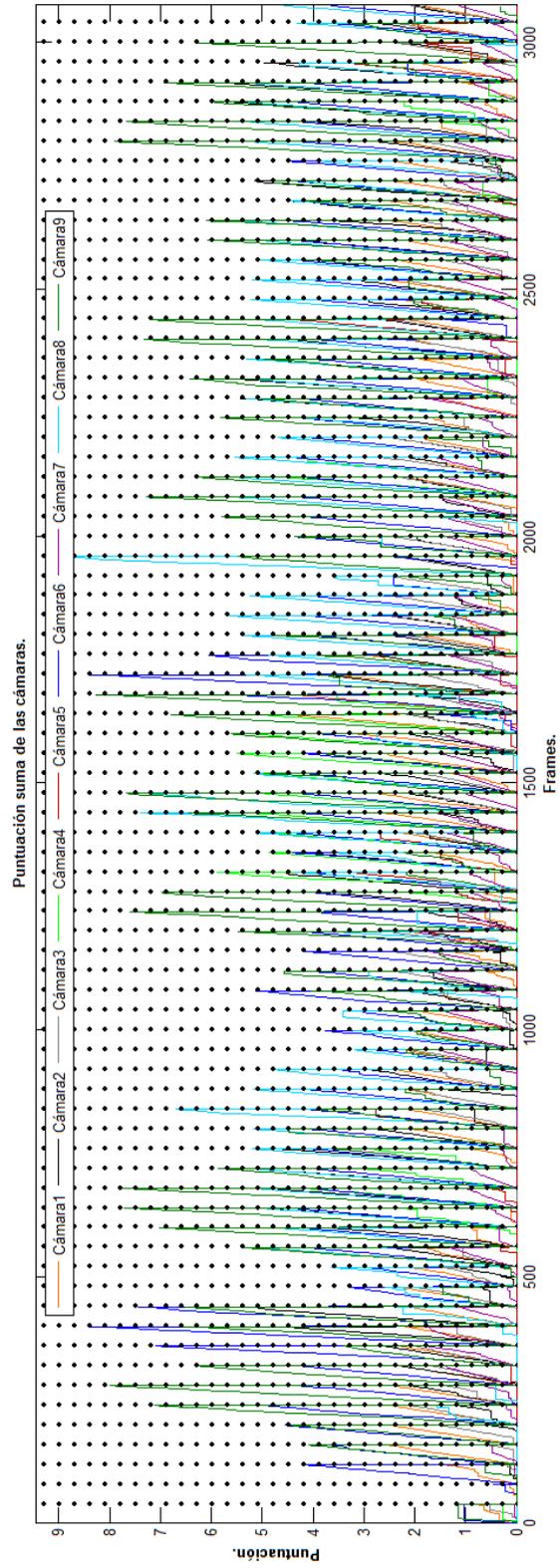
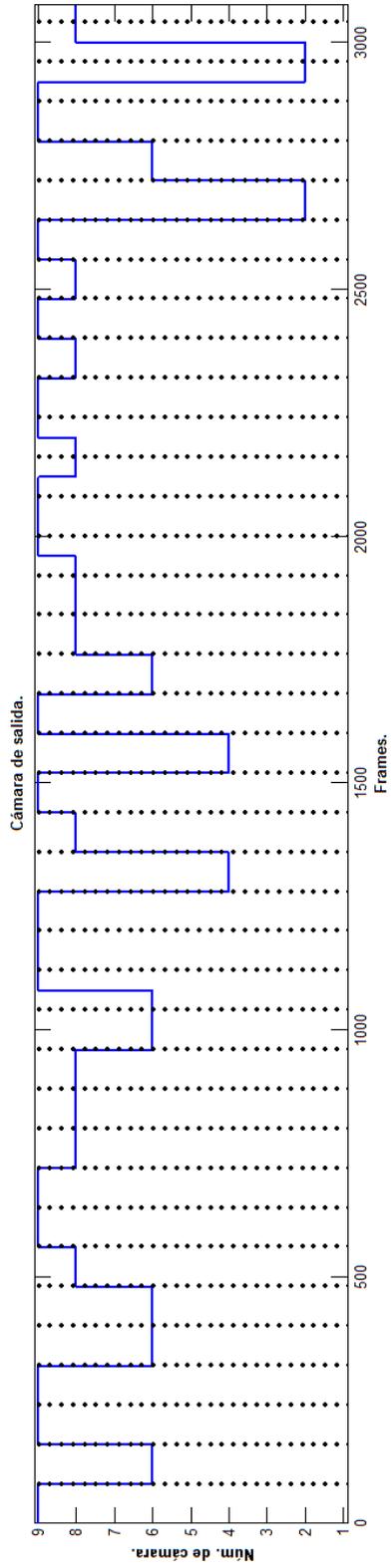


Figura 4.27: Puntuación acumulada (vídeo total)

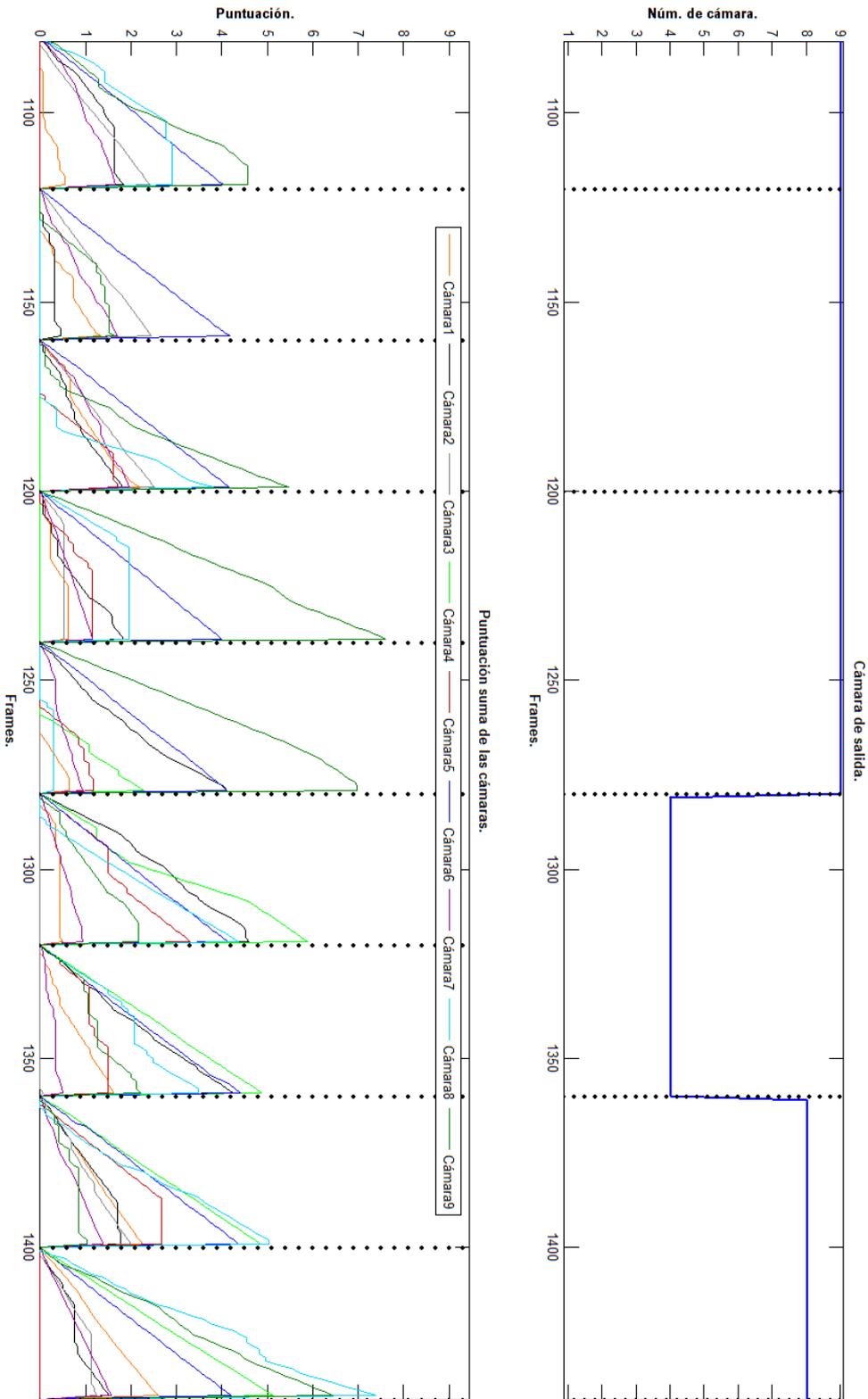


Figura 4.28: Puntuación acumulada (parcial)

4.4. Aportación de QualityBlur y userAttention

En este caso se va a usar el dataset de fútbol para mostrar la aportación de estos dos bloques, tanto por separado como juntos.

Para ello se ha ejecutado el sistema en cuatro nuevas situaciones, pero con los parámetros seleccionados antes como óptimos:

- Ejecución sin QualityBlur ni userAttention.
- Ejecución con QualityBlur pero sin userAttention.
- Ejecución sin QualityBlur pero con userAttention.
- Ejecución con QualityBlur y con userAttention.

4.4.1. QualityBlur

El coeficiente de emborronamiento es un valor entre $[0, 1]$ pero al observar el cuadro 4.4.1 (mostrado en tanto por ciento), se observa que las imágenes procedentes de los datasets no son de alta calidad, de hecho las obtenidas del dataset de fútbol son de mala calidad, esto repercute en la puntuación obtenida para cada *frame*, esto se aprecia en la figura 4.4.2, que representa la puntuación total, lo cuál finalmente alterará la salida, como se ve en la figura 4.4.2.

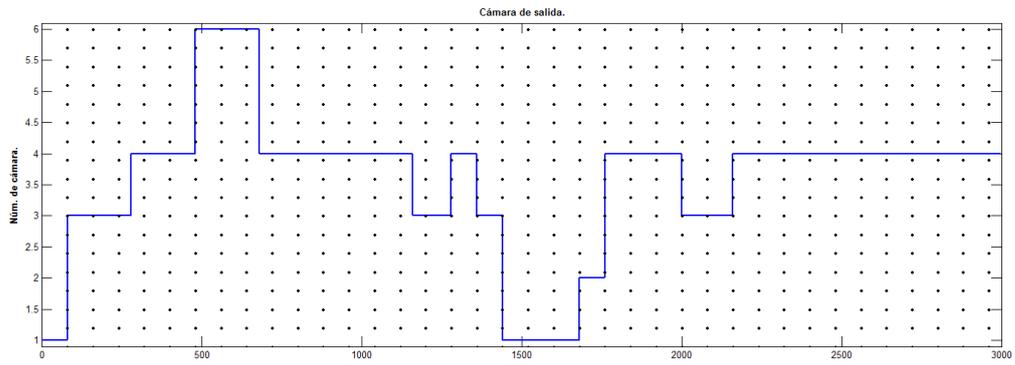
Cámara	Coef. de emborronamiento (%)		
	Fútbol	Baloncesto	Tenis
1	20,349	49,419	21,802
2	16,860	45,349	24,419
3	26,453	57,558	18,605
4	16,860	48,256	41,860
5	14,826	54,070	38,372
6	13,081	64,244	37,791
7	-	38,372	13,953
8	-	-	37,791
9	-	-	36,337

Cuadro 4.2: Coeficientes emborronamiento.

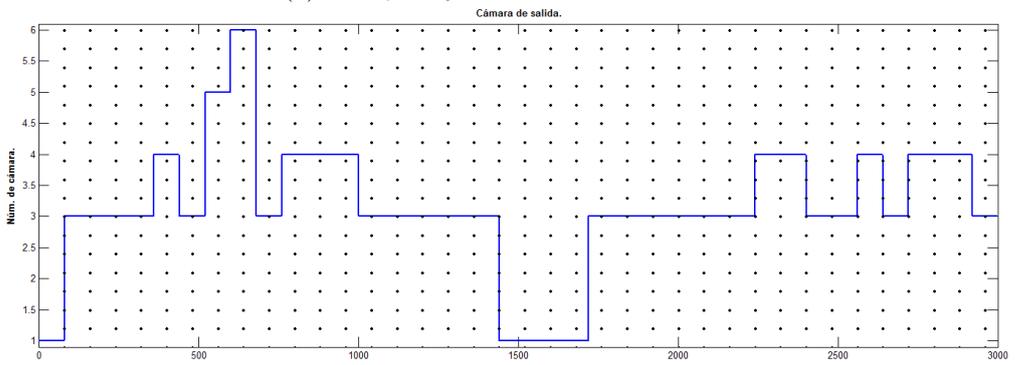
4.4.2. userAttention

Como se puede ver a lo largo de la figura 4.4.2, en los momentos de monotonía es el bloque de **userAttention** el que busca una cámara diferente que aporte innovación al montaje del vídeo. Sin este bloque la salida podría permanecer bloqueada en la misma cámara durante numerosos *frames*.

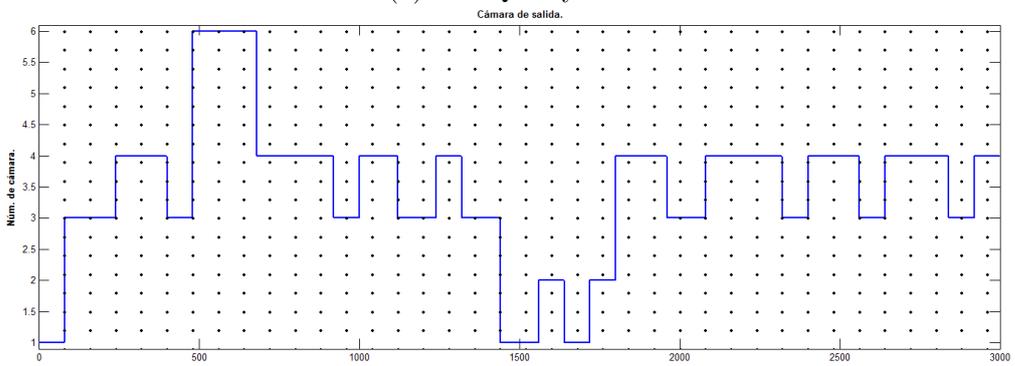
Se han citado los casos de monotonía ya que son los más sencillos de comprobar, no obstante el bloque está en funcionamiento durante todo el montaje.



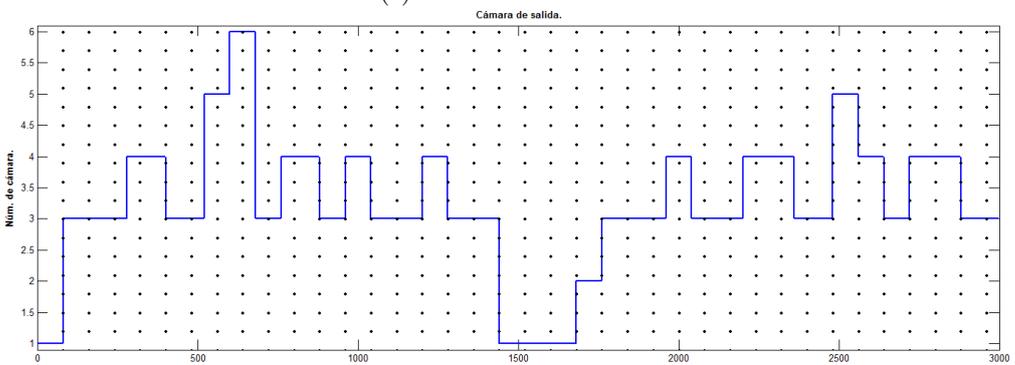
(a) Sin QualityBlur. Sin userAttention



(b) Con QualityBlur

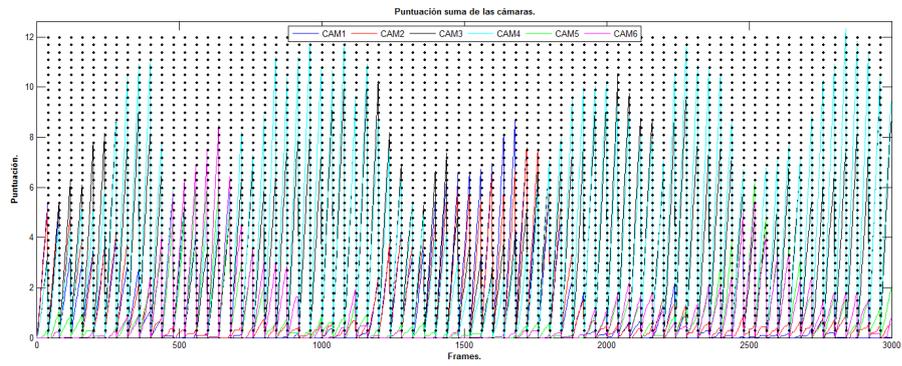


(c) Con userAttention

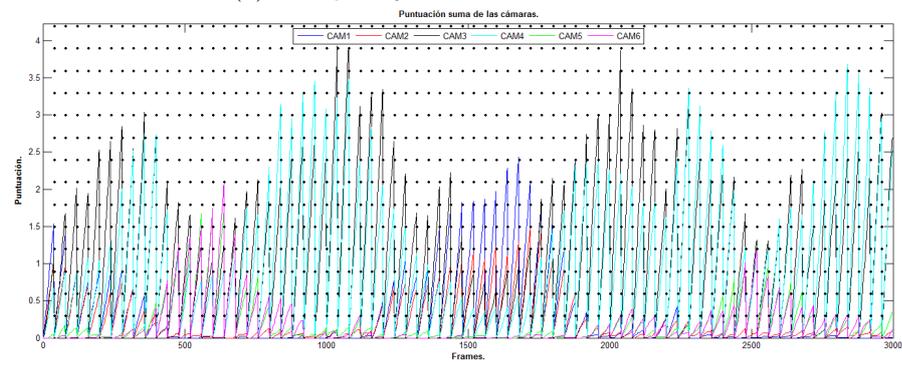


(d) Con QualityBlur. Con userAttention

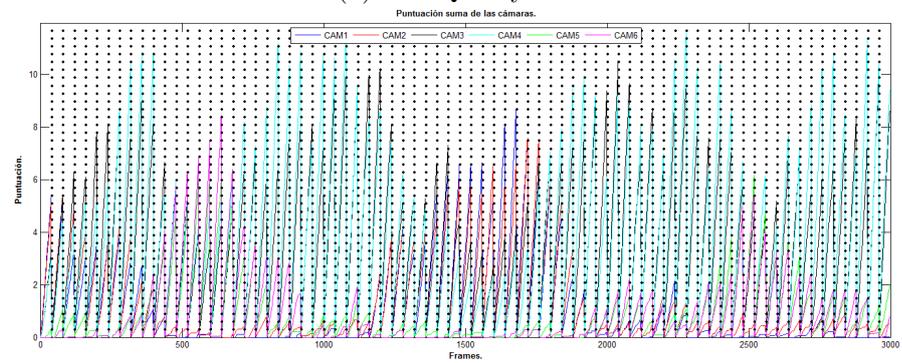
Figura 4.29: Distintas salidas. Dataset fútbol.



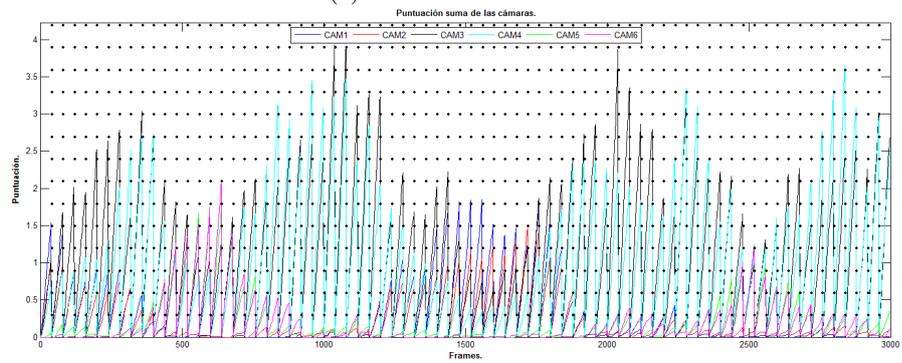
(a) Sin QualityBlur. Sin userAttention



(b) Con QualityBlur



(c) Con userAttention



(d) Con QualityBlur. Con userAttention

Figura 4.30: Distintas salidas. Dataset fútbol.

4.5. Respuesta en tiempo del sistema

Para analizar el tiempo que emplea el sistema, se va a utilizar una vez más el SET 1 del dataset de baloncesto.

Los tiempos que se tendrán en cuenta serán los correspondientes a cada uno de los bloques expuestos en la sección 3.2, y el tiempo total.

Resize	71.56
AreaNBlobs	329.8
QualityBlur	111.2
userAttention	0.003729
Score	0.0004826
Decisor	0.0001773
Presentation	51.43

Cuadro 4.3: Valores de tiempo medio por *frame* (ms)

Resize	111000
AreaNBlobs	495011
QualityBlur	166505
userAttention	5.812
Score	0.851
Decisor	0.6261
Presentation	79700

Cuadro 4.4: Valores de tiempo empleado por los bloques en los 1499 *frames* del vídeo (ms)

La suma total del tiempo empleado por los bloques es de 852.350 ms, sin embargo el tiempo total del sistema asciende a 983.000 ms. Esta diferencia entre tiempos se debe a ciertos algoritmos que no forma parte específicamente de ninguno de los bloques evaluados, pero que también ha de ser tenido en cuenta.

Por lo tanto los bloques en cuestión suponen un 86.7% del tiempo total de ejecución.

Al analizar el tiempo total, 983 segundos, se pueden obtener más estadísticas, como por ejemplo el número de *frames* por segundo que son analizados.

$$\frac{\text{Frames totales}}{\text{Tiempo total}} = \frac{1499}{983} = 1,5249\text{fps}$$

Como se puede observar, con estos tiempos el sistema queda lejos del uso en tiempo real, tema que será propuesto en el trabajo futuro.

El objetivo ha sido en todo momento la realización del sistema completo, no la optimización del mismo. Es por ello que los tiempos empleados en cada uno de los bloques son ampliamente mejorables y son una de las propuestas de trabajo futuro.

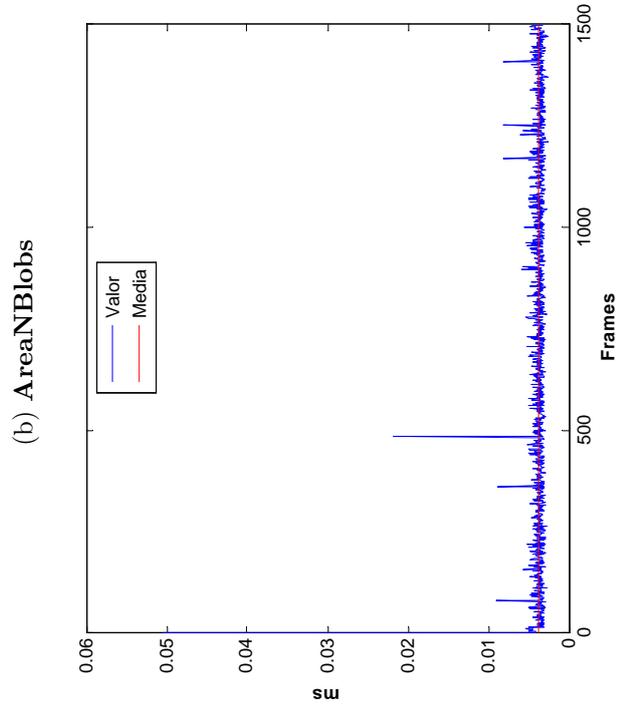
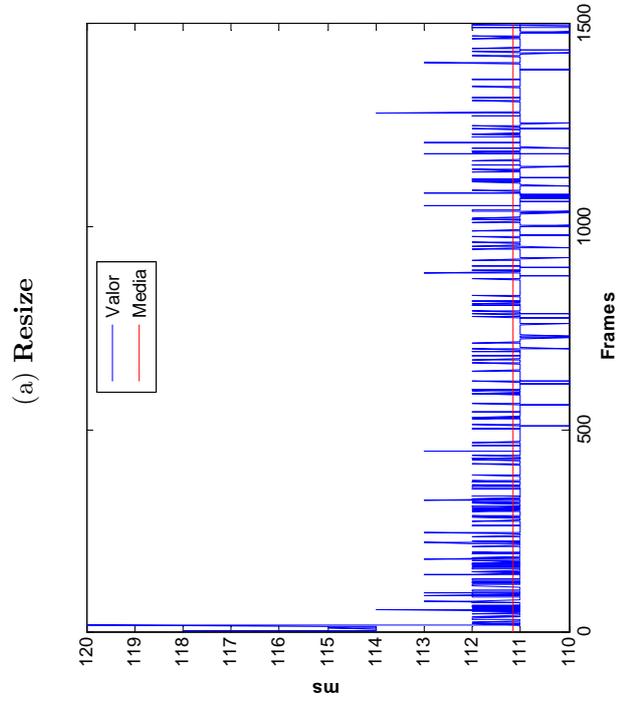
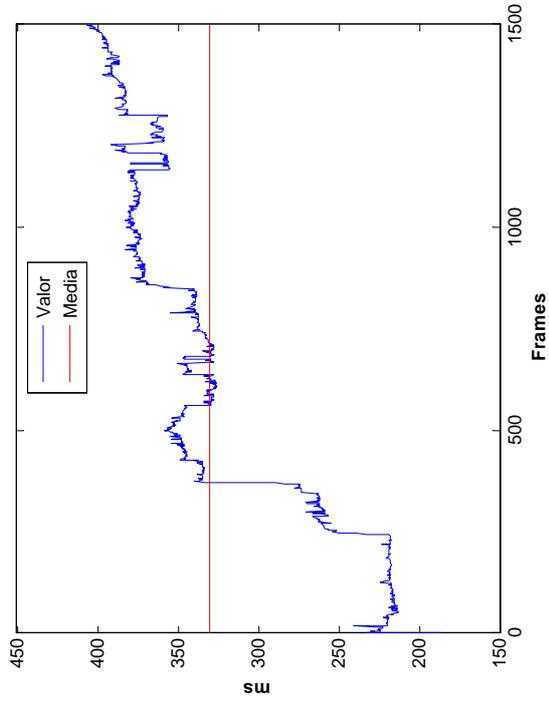
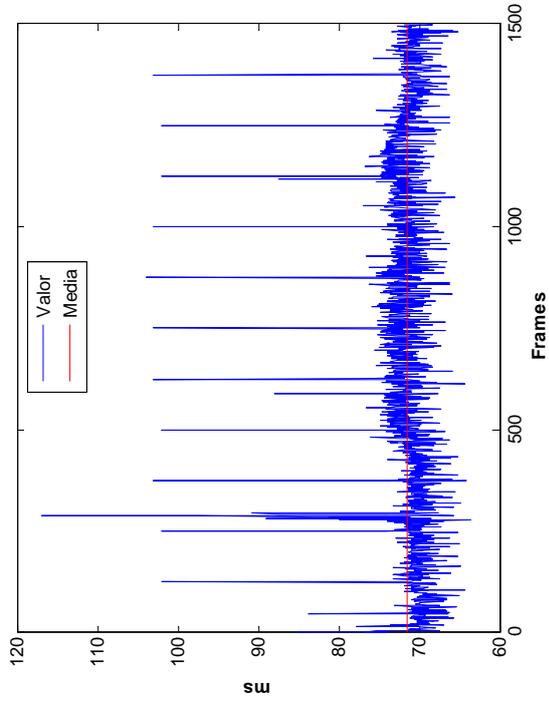
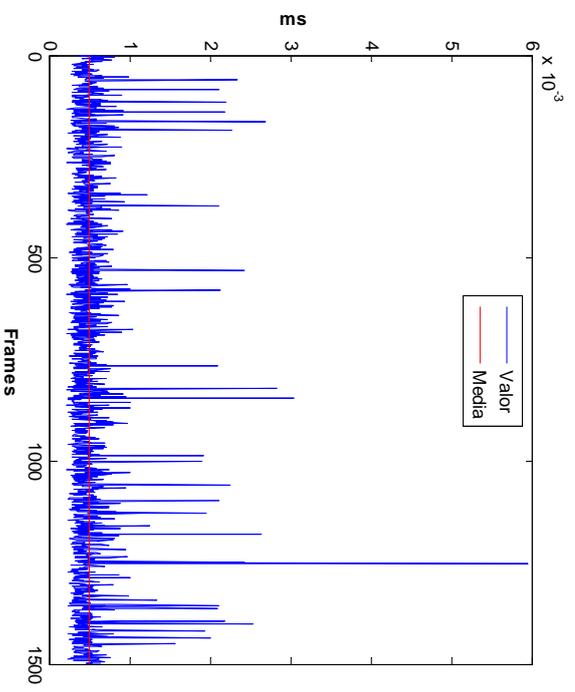
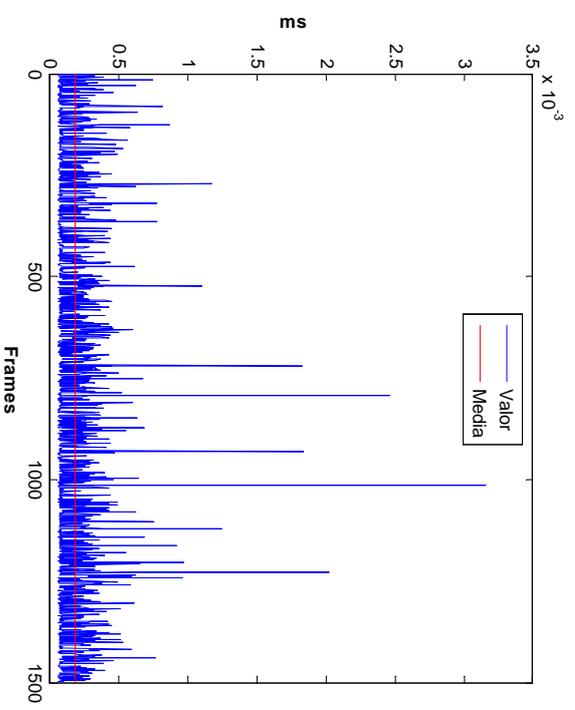


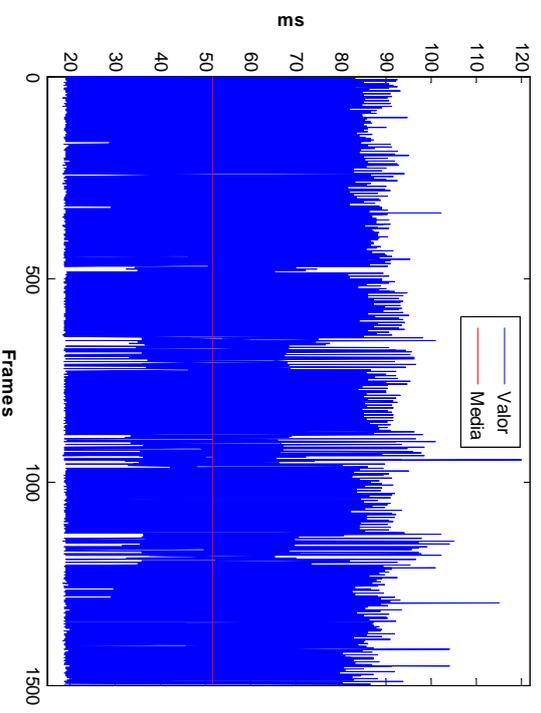
Figura 4.31: Tiempo empleado por cada bloque



(e) Score



(f) Decisor



(g) Presentation

Figura 4.32: Tiempo empleado por cada bloque (II)

Capítulo 5

Conclusiones y trabajo futuro

5.1. Conclusiones

A lo largo de este proyecto se han desarrollado e implementado algoritmos, y se han integrado conformando un sistema completo, con el cual se han realizado pruebas con imágenes y con vídeos. Después de todo esto, la conclusión es que funciona correctamente dentro de la limitación en tiempo y esfuerzo que supone un Proyecto de Fin de Carrera, ya que realiza montajes adecuados para su visualización, mostrando lo más significativo de cada una de las cámaras. Cabe reseñar que no se ha hecho un estudio subjetivo de la calidad del mismo, usando para ello, distintos montajes y varias pruebas en un número representativo de usuarios. Esto se explica por las limitaciones en tiempo que conlleva un PFC. No obstante este análisis se propondrá en trabajos futuros.

El montaje de un vídeo depende totalmente del contenido, y para decidir como se monta, primeramente, hay que extraer distintas características de cada una de las cámaras, para después compararlas. En este PFC se evalúan tres características: número de blobs, área en movimiento y emborronamiento.

Por otro lado se ha añadido un aspecto nuevo, como es el coeficiente de atención, el cual ha sido completamente diseñado desde cero y supone un aporte novedoso. Su función es la de simular la atención del usuario, lo cual realiza correctamente. En definitiva, este bloque innova teniendo en cuenta una característica no usada hasta ahora, abriendo así nuevos caminos.

Por lo tanto, como ya se ha dicho al principio de este punto el proyecto ha cumplido con los objetivos que se planteaban al comienzo del mismo.

El sistema tiene una gran potencialidad de uso, que, aunque en estas primeras fases de desarrollo es más difícil darse cuenta de ello, a medida que se vayan añadiendo módulos de análisis y reduciendo el tiempo de acción, su utilidad se multiplicará.

No hay que olvidar que la finalidad de este proyecto era la creación de un sistema completo, ya que la propia naturaleza del proyecto, un PFC, impone unas limitaciones en tiempo, es ahora, con esta base cuando hay que empezar a trabajar para ir mejorando cada uno de los apartados e ir añadiendo algoritmos y bloques para poder tener más características a evaluar.

5.2. Trabajo futuro

Durante todas estas páginas se ha mostrado la realización de un sistema completo de selección de la mejor vista en un entorno multicámara, pero esto sólo es el comienzo, aún queda muchas cosas por hacer, y parte de este trabajo es el propuesto a continuación como líneas generales de futuro trabajo.

5.2.1. Parametrizable para cualquier número de cámaras

En principio este sistema ha sido diseñado para datasets de 6, 7 y 9 cámaras, pero su funcionamiento es totalmente extrapolable a otro número de cámaras.

Lo adecuado sería realizar una adaptación del algoritmo para convertirlo en dinámico, de tal manera que al usuario se le preguntará cuál es el número de cámaras del dataset, y ya con ello, el bloque actuará en consecuencia.

5.2.2. Estudio interpolación

Como se comentó en la parte de diseño, las imágenes van a sufrir un redimensionado, para trabajar en las mismas condiciones. Para realizar este cambio de tamaño es necesario un proceso de interpolación.

Convendría realizar un estudio sobre los distintos métodos de interpolación, para saber cual de ellos es el más adecuado para cada caso.

5.2.3. Función de Score

Esta función ha sido solo una aproximación, una posibilidad, pero sería necesario un estudio sobre ella: sobre qué otros tipos hay, sobre qué tipo de contenidos funciona mejor una u otra, etc. porque como ya se ha dejado claro, el contenido del vídeo es muy importante a la hora de hacer el montaje, y si bien es cierto que se puede crear una función general que funcione con la mayoría de contenidos, también puede ser interesante tener funciones particulares para un entorno determinado.

5.2.4. Algoritmo de orden de cámaras

A la hora de realizar un montaje con varias cámaras, es posible, que para facilitar la visión al espectador sea mejor mostrar unas cámaras seguidas de otras, para que se vea cierta progresión. Es por eso que un algoritmo que diera más importancia a las cámaras secuenciales a la que se está mostrando sería de gran ayuda.

La idea sería que cada set de vídeo viniera con un archivo de texto en el que se concretara la relación entre las distintas cámaras el cual el programa interpretase y ponderase la puntuación según estas relaciones.

5.2.5. Reducción tiempo de ejecución

Uno de los posibles usos del sistema sería en tiempo real, para lo cual habría que reducir el tiempo que requieren cada uno de los bloques, en definitiva, la optimización de cada uno de ellos. Habría que llegar a tasas de análisis de 25 fps.

5.2.6. Detección de objetos

En ocasiones la importancia de un *frame* sobre otro depende de un único objeto. En el caso del fútbol por ejemplo, lo más importante es la pelota, y por lo tanto sería necesario detectarla para poder marcarla, asignándole una puntuación alta, y así seguirla a través del dataset de cámaras.

5.2.7. Detección de eventos

En un principio suena ligeramente a ciencia ficción, pero el objetivo principal sería conocer el entorno que estamos estudiando para a través de ciertas características observadas poder decir que cierto evento está ocurriendo. Por ejemplo, en el caso del tenis, si los dos jugadores están quietos, y están alrededor de unas zonas previamente delimitadas, se podría extrapolar que se está realizando un saque.

5.2.8. Sonido

Se puede utilizar el audio de los vídeos como característica de selección de cámara como otro parámetro más. También podría ser usado en el apartado de reconocimiento de eventos, en el caso de un partido de fútbol o baloncesto, el sonido del silbato o la bocina: podrían denotar eventos desde faltas, gol, fin de posesión, etc.

5.2.9. Análisis subjetivo de funcionamiento del sistema.

Se debería llevar a cabo un estudio para discernir el buen o mal funcionamiento del sistema, poniéndolo a prueba mediante el visionado por parte de un grupo representativo de usuarios, donde se les presentarían distintos montajes realizados a partir de este sistema variando sus parámetros, y se recogerían sus opiniones.

Bibliografía

- [1] Fahad Daniyal, Murtaza Taj, and Andrea Cavallaro. Content and task-based view selection from multiple video streams. *Multimedia Tools and Applications*. 3, 4, 5, 6, 7, 16
- [2] Murtaza Taj, Emilio Maggio, and Andrea Cavallaro. Multi-feature graph-based object tracking. In *Proceedings of the 1st international evaluation conference on Classification of events, activities and relationships*, CLEAR'06, pages 190–199, Berlin, Heidelberg, 2007. Springer-Verlag. 3
- [3] Xuhui Zhou, Robert T. Collins, Takeo Kanade, and Peter Metes. A master-slave system to acquire biometric imagery of humans at distance. In *ACM International Workshop on Video Surveillance*, pages 113–120. ACM Press, 2003. 4
- [4] Changsong Shen, Chris Zhang, and Sidney Fels. A multi-camera surveillance system that estimates quality-of-view measurement. In *In IEEE International Conference on Image Processing*, 2007. 4
- [5] Han-Saem Park, Soojung Lim, Jun-Ki Min, and Sung-Bae Cho. *Optimal View Selection and Event Retrieval in Multi-Camera Office Environment*, pages 106–110. 2008. 5
- [6] R Goshorn, J Goshorn, Deborah Goshorn, and Hamid Aghajan. Architecture for cluster-based automated surveillance network for detecting and tracking multiple persons. *Proc of ICDCS*, 7:219–226, 2007. 6
- [7] Alberto Del Bimbo and Federico Pernici. Towards on-line saccade planning for high-resolution image sensing. *Pattern Recogn. Lett.*, 27:1826–1834, November 2006. 7
- [8] Hao Jiang, Sidney Fels, and James J Little. Optimizing multiple object tracking and best view video synthesis. *IEEE Transactions on Multimedia*, 10(6):997–1012, 2008. 7
- [9] P. Peixoto J. Batist and H. Aralújo. Real-time active visual surveillance by integrating peripheral motion detection with foveated tracking. In *Proceedings of the 1998 IEEE Workshop on Visual Surveillance*, pages 18–, Washington, DC, USA, 1998. IEEE Computer Society. 7
- [10] Mohammad Rezaeian. Sensor scheduling for optimal observability using estimation entropy. *Fifth Annual IEEE International Conference on Pervasive Computing and Communications Workshops PerComW07*, pages 307–312, 2007. 7
- [11] Cash J Costello, Christopher P Diehl, Amit Banerjee, and Hesky Fisher. Scheduling an active camera to observe people. *International Multimedia Conference*, page 39, 2004. 7
- [12] Cantata, video and image datasets index. <http://www.multitel.be/cantata/>, 2008.
- [13] APIDIS basket ball dataset. <http://www.apidis.org/Dataset/>, 2009.
- [14] OpenCV 2.0 C Reference. <http://opencv.willowgarage.com/documentation/>, 2009.
- [15] OpenCVWiki. <http://opencv.willowgarage.com/wiki/>, 2010.
- [16] GIMP. <http://www.gimp.org/>, 2001-2011.

- [17] Ciaran O' Conaire, Philip Kelly, Damien Connaghan, and Noel E. O'Connor. Tennissense: A platform for extracting semantic information from multi-camera tennis data. In *DSP 2009 - 16th International Conference on Digital Signal Processing*, pages 1062–1067, 2009.
- [18] X. Marichal, W.Y. Ma, and H.J. Zhang. Blur determination in the compressed domain using dct information. pages II:386–390, 1999. 85
- [19] X. Chen and J.E. Davis. An occlusion metric for selecting robust camera configurations. 19(4):xx–yy, July 2008.
- [20] Fahad Daniyal, Murtaza Taj, and Andrea Cavallaro. Content and task-based view selection from multiple video streams. *Multimedia Tools Appl.*, 46:235–258, January 2010.
- [21] E. Kurutepe, M.R. Civanlar, and A.M. Tekalp. Client-driven selective streaming of multiview video for interactive 3dtv. 17(11):1558–1565, November 2007.
- [22] Konstantinos Tarabanis, Roger Y. Tsai, and Peter K. Allen. Engineering systems with intelligence. chapter Overview of the MVP sensor planning system for robotic vision tasks, pages 285–293. Kluwer Academic Publishers, Norwell, MA, USA, 1992.
- [23] T. D'Orazio, M. Leo, N. Mosca, P. Spagnolo, and P. L. Mazzeo. A semi-automatic system for ground truth generation of soccer video sequences. In *Proceedings of the 2009 Sixth IEEE International Conference on Advanced Video and Signal Based Surveillance, AVSS '09*, pages 559–564, Washington, DC, USA, 2009. IEEE Computer Society.
- [24] Xian sheng Hua, Lie Lu, and Hong jiang Zhang. Automatically converting photographic series into video. In *12th ACM International Conference on Multimedia*, pages 708–715. Press, 2004.

Apéndice A

Parametrización

A.1. Aplicación a imágenes

Se han realizado pruebas con imágenes para comprobar el correcto funcionamiento de algunos algoritmos, como es el caso del algoritmo **QualityBlur**, el cual es el encargado de devolver un coeficiente de emborronamiento, recibiendo únicamente la imagen a evaluar. Las distintos valores en el coeficiente son causa de la variación de los parámetros de control de este algoritmo.

- *MinDCTValue*: Mínimo valor de la DCT a tener en cuenta.
- *MaxHistValue*: Valor relativo del histograma que puede alcanzar.

Sobre este algoritmo se realizarán dos comprobaciones, la primera sería para el valor máximo de definición, es decir, una imagen perfectamente nítida sin emborronamiento, en cuyo caso el algoritmo ha de devolver valor 1. Para comprobar esto se va a seleccionar una imagen de alta definición, procedente de un Blu-Ray. La cual se tomará como referencia, para poder controlar que parámetros del algoritmo funcionan mejor o peor.

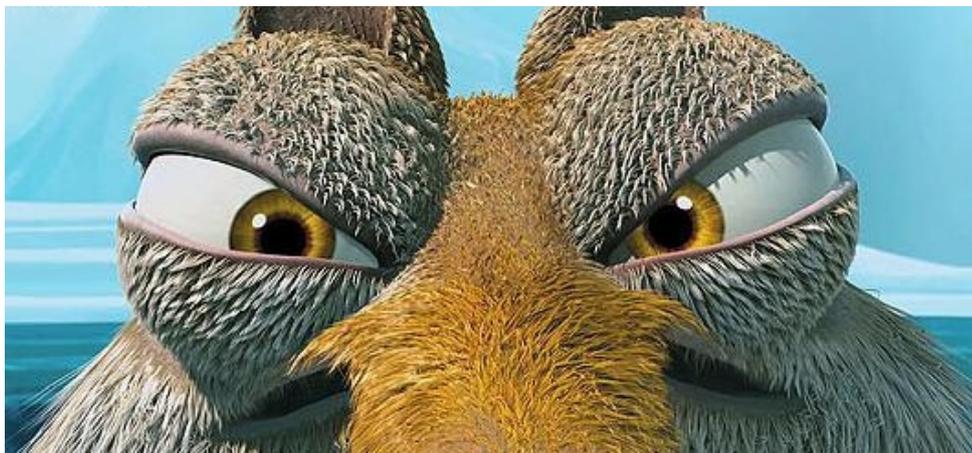


Figura A.1: Imagen referencia.

Una vez visto para qué valores de los parámetros del algoritmo se obtienen valores cercanos al 1, en la imagen referente, se pasará a la segunda comprobación, la cual se llevará a cabo verificando que, efectivamente, la misma imagen más emborronada devuelve un coeficiente de emborronamiento menor.

Para comprobar que esto se cumple se ha desarrollado un banco de pruebas específico. Consta de dos grupos de cuatro imágenes, que han sido emborronadas artificialmente usando un editor de imágenes, cada una con distintos grados de emborronamiento, numerados del 1-4, de menor a mayor grado.

Como aclaración hay que decir que aunque en el sistema el bloque QualityBlur devuelva un valor entre $[0, 1]$, en los siguientes cuadros se mostrará este valor en tanto por ciento.



(a) Grado 1.



(b) Grado 2.



(c) Grado 3.



(d) Grado 4.

Figura A.2: Prueba 1.



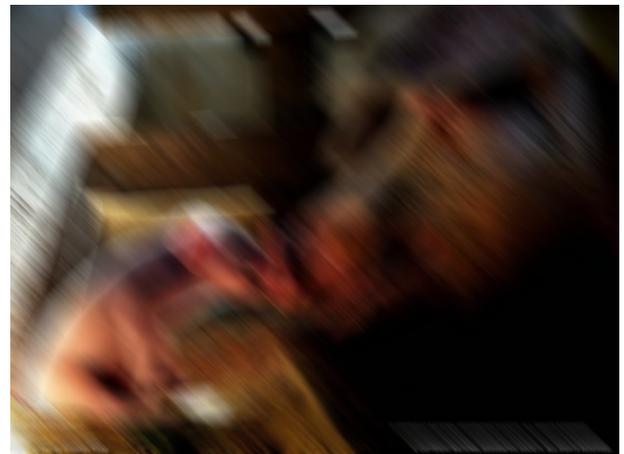
(a) Grado 1.



(b) Grado 2.



(c) Grado 3.



(d) Grado 4.

Figura A.3: Prueba 2.

En primer lugar, antes de analizar más específicamente, se puede apreciar, que para ciertos valores en las imágenes totalmente desenfocadas como el caso del grado 4: el algoritmo no responde como se esperaba, devolviendo un valor del coeficiente incluso superior al del grado 3. Para el uso que se le dará a este algoritmo no supone un impedimento, puesto que las imágenes que se usarán estarán siempre entre los tres primeros grados de emborronamiento. No obstante este efecto es evitable, escogiendo los valores adecuados.

A la vista de los resultados se observa que a medida que se aumenta el parametro MaxHistValue, la imagen referencia más se aleja del 100 % de nitidez, también valores mayores de MinDCTValue tienen el mismo efecto, aunque en menor grado.

Para los valores en los que se da el mayor porcentaje de nitidez (en la imagen de referencia), se produce un problema: la sensibilidad del algoritmo se ve reducida, como se puede comprobar, del grado 1 al 2 sin variar el coeficiente.

MinDCTValue	6,00	8,00	10,00	6,00	8,00	10,00	6,00	8,00	10,00	6,00	8,00	10,00
MaxHistValue	0,05	0,05	0,05	0,10	0,10	0,10	0,15	0,15	0,15	0,20	0,20	0,20
Imagen	blur											
Referencia	95,640	91,860	88,081	89,535	86,337	79,651	86,337	77,907	71,221	79,651	70,349	63,081

(a) Imagen referencia.

MinDCTValue	6,00	8,00	10,00	6,00	8,00	10,00	6,00	8,00	10,00	6,00	8,00	10,00
MaxHistValue	0,05	0,05	0,05	0,10	0,10	0,10	0,15	0,15	0,15	0,20	0,20	0,20
Imagen	blur											
Grado 1	68,895	68,895	68,895	54,651	54,651	54,651	46,802	46,802	45,349	43,023	43,023	39,535
Grado 2	68,895	68,895	67,733	54,651	52,326	45,930	46,802	35,756	24,709	36,047	22,093	13,663
Grado 3	15,116	13,663	12,209	13,663	12,210	12,209	12,209	10,465	10,465	12,209	10,465	8,140
Grado 4	26,453	18,605	16,860	18,605	12,500	8,721	14,535	8,721	8,721	8,721	8,721	6,395

(b) Prueba 1

MinDCTValue	6,00	8,00	10,00	6,00	8,00	10,00	6,00	8,00	10,00	6,00	8,00	10,00
MaxHistValue	0,05	0,05	0,05	0,10	0,10	0,10	0,15	0,15	0,15	0,20	0,20	0,20
Imagen	blur											
Grado 1	58,140	58,140	58,140	47,674	47,674	50,291	42,151	40,116	38,372	39,244	31,686	29,070
Grado 2	58,140	56,977	51,163	47,674	36,337	26,163	35,174	23,256	21,512	25,000	17,733	16,279
Grado 3	17,151	15,116	13,663	15,116	12,209	12,209	12,209	12,209	12,209	12,209	8,721	8,721
Grado 4	27,907	18,605	18,605	18,605	16,279	8,721	16,279	8,721	8,721	8,721	8,721	8,721

(c) Prueba 2

Cuadro A.1: Coeficiente de emborronamiento.

En [18], los valores recomendados son “8” para MinDCTValue y “0,1” para MaxHistValue. Como se puede ver en la tabla, estos valores de los parámetros, dan resultados adecuados, ya que la imagen de referencia queda muy cercana al 100 %, con un 86,337 %, y las distintas gradaciones de emborronamiento quedan escaladas en el orden correcto, solucionando el problema de sensibilidad que se presentaba para valores más bajos. No obstante el problema del último grado persiste.

El primer valor en solventar todos los problemas, y aún dando un buen porcentaje de nitidez de la imagen de referencia sería

- MinDCTValue = 10
- MaxHistValue = 0,1
- Nitidez imagen referencia = 0.7965

Las pruebas anteriores demuestran el correcto funcionamiento del algoritmo, pero a la hora de la utilización del mismo, no hay porqué elegir los parámetros seleccionados como “mejores”, ya que dependiendo del contenido del vídeo puede interesar variar esos parámetros.

También la comparación entre las distintas cámaras proporcionará un orden de calidad, lo cual permite comprobar si los valores obtenidos por los parámetros elegidos respeta ese orden. Hay que tener en cuenta que a la hora de ejecutar este algoritmo en las distintas cámaras, éste no compara imágenes iguales, sino que puede tratarse de imágenes totalmente distintas.

Por ese motivo, a continuación, se muestra un estudio de los coeficientes de emborronamiento de frames de cada uno de los dataset sometidos a prueba.

A.1.1. Dataset de 6 cámaras (Fútbol)

Para este dataset se evalúa el coeficiente de emborronamiento de cada una de las cámaras en el mismo instante de tiempo. Para ello se ejecuta el bloque QualityBlur sobre las imágenes que se pueden ver en la figura A.4.

Los valores del coeficiente de emborronamiento para los distintos parámetros y cámaras queda reflejado en el cuadro A.2.

A.1.2. Dataset de 7 cámaras (Baloncesto)

Aquí se procede como en la sección anterior pero con 7 imágenes, ya que el dataset dispone de 7 cámaras. Las imágenes se encuentran en las figuras A.5 y A.6. Los valores del coeficiente de emborronamiento según los parámetros elegidos y la cámara están en el cuadro A.3.

A.1.3. Dataset de 9 cámaras (Tenis)

Como en los casos anteriores se evalúan las cámaras correspondientes a este dataset, en este caso 9, las imágenes sometidas a prueba se pueden ver en las figuras A.7 y A.8. Los valores del coeficiente de emborronamiento se encuentran en el cuadro A.4.



(a) Cámara 1.



(b) Cámara 2.



(c) Cámara 3.



(d) Cámara 4.



(e) Cámara 5.



(f) Cámara 6.

Figura A.4: Dataset fútbol.



(a) Cámara 1.



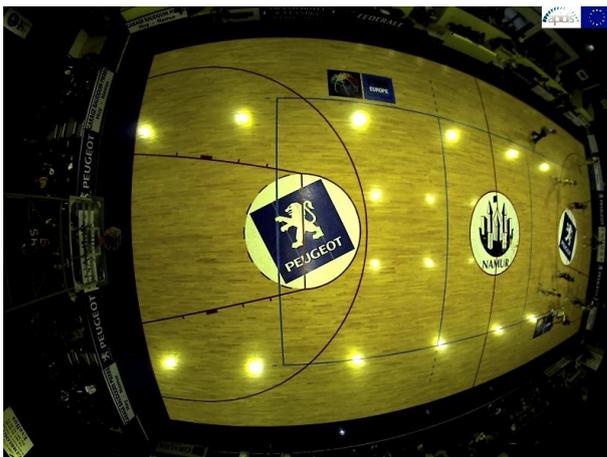
(b) Cámara 2.



(c) Cámara 3.



(d) Cámara 4.



(e) Cámara 5.



(f) Cámara 6.

Figura A.5: Dataset baloncesto.



(g) Cámara 7.

Figura A.6: Dataset Baloncesto (II)

MinDCTValue	2	3	4	5	8	2	3	4	8	10
MaxHistValue	0,05	0,05	0,05	0,05	0,05	0,1	0,1	0,1	0,1	0,1
Imagen	blur									
Cámara 1	77,907	65,698	60,465	54,942	44,186	54,360	44,186	36,047	25,581	20,349
Cámara 2	75,581	62,209	57,558	49,419	38,953	54,651	45,349	36,628	20,349	16,860
Cámara 3	89,535	79,651	71,221	63,953	52,326	70,349	54,651	45,930	29,651	26,453
Cámara 4	82,267	71,802	60,465	56,977	44,186	61,628	47,384	40,116	20,349	16,860
Cámara 5	82,267	68,895	61,628	52,616	41,570	54,360	43,895	32,267	17,733	14,826
Cámara 6	71,221	58,140	49,128	45,640	28,779	54,360	39,535	32,267	16,860	13,081

Cuadro A.2: Coeficientes emborronamiento. Dataset fútbol.

MinDCTValue	2	3	4	5	8	2	3	4	8	10
MaxHistValue	0,05	0,05	0,05	0,05	0,05	0,1	0,1	0,1	0,1	0,1
Imagen	blur	blur	blur	blur	blur	blur	blur	blur	blur	blur
Cámara 1	100,00	95,640	89,535	86,337	72,965	91,860	84,303	74,709	55,523	49,419
Cámara 2	100,00	95,640	88,081	83,140	70,058	91,860	78,488	72,674	51,744	45,349
Cámara 3	100,00	100,000	97,674	95,640	87,791	100,00	93,605	87,791	67,151	57,558
Cámara 4	100,00	95,640	88,081	81,395	69,477	91,860	81,395	72,674	51,744	48,256
Cámara 5	100,00	100,00	95,640	93,605	80,814	95,640	87,791	80,814	59,302	54,070
Cámara 6	100,00	100,00	100,00	97,674	87,791	100,00	95,640	87,791	71,221	64,244
Cámara 7	100,00	93,605	87,791	80,814	64,826	88,081	77,035	66,279	45,930	38,372

Cuadro A.3: Coeficientes emborronamiento. Dataset baloncesto.



(a) Cámara 1.



(b) Cámara 2.



(c) Cámara 3.



(d) Cámara 4.



(e) Cámara 5.



(f) Cámara 6.

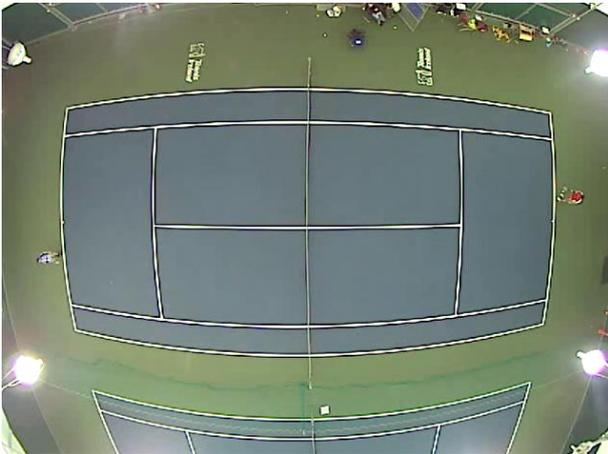
Figura A.7: Dataset tenis.



(g) Cámara 7.



(h) Cámara 8.



(i) Cámara 9.

Figura A.8: Dataset tenis (II)

MinDCTValue	2	3	4	5	8	2	3	4	8	10
MaxHistValue	0,05	0,05	0,05	0,05	0,05	0,1	0,1	0,1	0,1	0,1
Imagen	blur	blur	blur	blur	blur	blur	blur	blur	blur	blur
Cámara 1	90,116	83,721	75,000	67,733	57,558	66,279	55,233	46,221	21,802	21,802
Cámara 2	76,163	63,953	57,558	53,779	40,988	62,209	53,779	45,640	30,233	24,419
Cámara 3	65,407	54,070	51,744	49,709	41,860	50,291	40,407	38,081	23,547	18,605
Cámara 4	93,895	85,174	77,907	73,837	64,826	79,360	71,802	64,826	50,291	41,860
Cámara 5	90,116	78,488	73,837	67,442	58,140	77,907	64,826	63,081	41,860	38,372
Cámara 6	89,535	79,651	73,256	67,733	59,593	77,907	67,442	58,721	43,314	37,791
Cámara 7	67,767	61,919	51,453	44,767	35,465	51,453	40,988	33,721	17,442	13,953
Cámara 8	100,000	93,605	89,535	86,337	74,709	86,337	78,779	69,477	48,256	37,791
Cámara 9	87,791	82,267	74,128	68,314	61,337	74,709	64,244	55,523	39,826	36,337

Cuadro A.4: Coeficientes emborronamiento. Dataset tenis.

Apéndice B

Gráficas de resultados intermedios

B.1. Introducción

A lo largo del capítulo 4, Integración y resultados, se mostraron las figuras más significativas de las pruebas a las que se sometió el sistema. En este apéndice se van a mostrar el resto de figuras, las cuales representan el valor de todas las características extraídas para cada dataset a lo largo del vídeo.

Para el caso del Baloncesto también se incluyen las distintas variantes cambiando los parámetros.

Las características que se mostrarán a continuación son:

- Número de blobs.

- Coeficiente de área.

- Coeficiente de emborronamiento.

- Coeficiente de atención.

- Puntuación de las cámaras (vídeo total).

- Puntuación de las cámaras (parcial).

Como ya se ha mencionado, los datasets tendrán más o menos figuras según las que ya se hayan mostrado en el capítulo anterior o no.

Para el dataset del baloncesto las figuras distintas del SET1 de parámetros incluyen en cada figura dos señales, una azul y una roja, la azul corresponde a la señal respuesta a los parámetros de entrada, mientras que la roja corresponde con la señal recogida con los parámetros de entrada del SET1, que es el considerado como óptimo.

B.2. Distinta elección de los parámetros

B.2.1. Baloncesto (SET1)

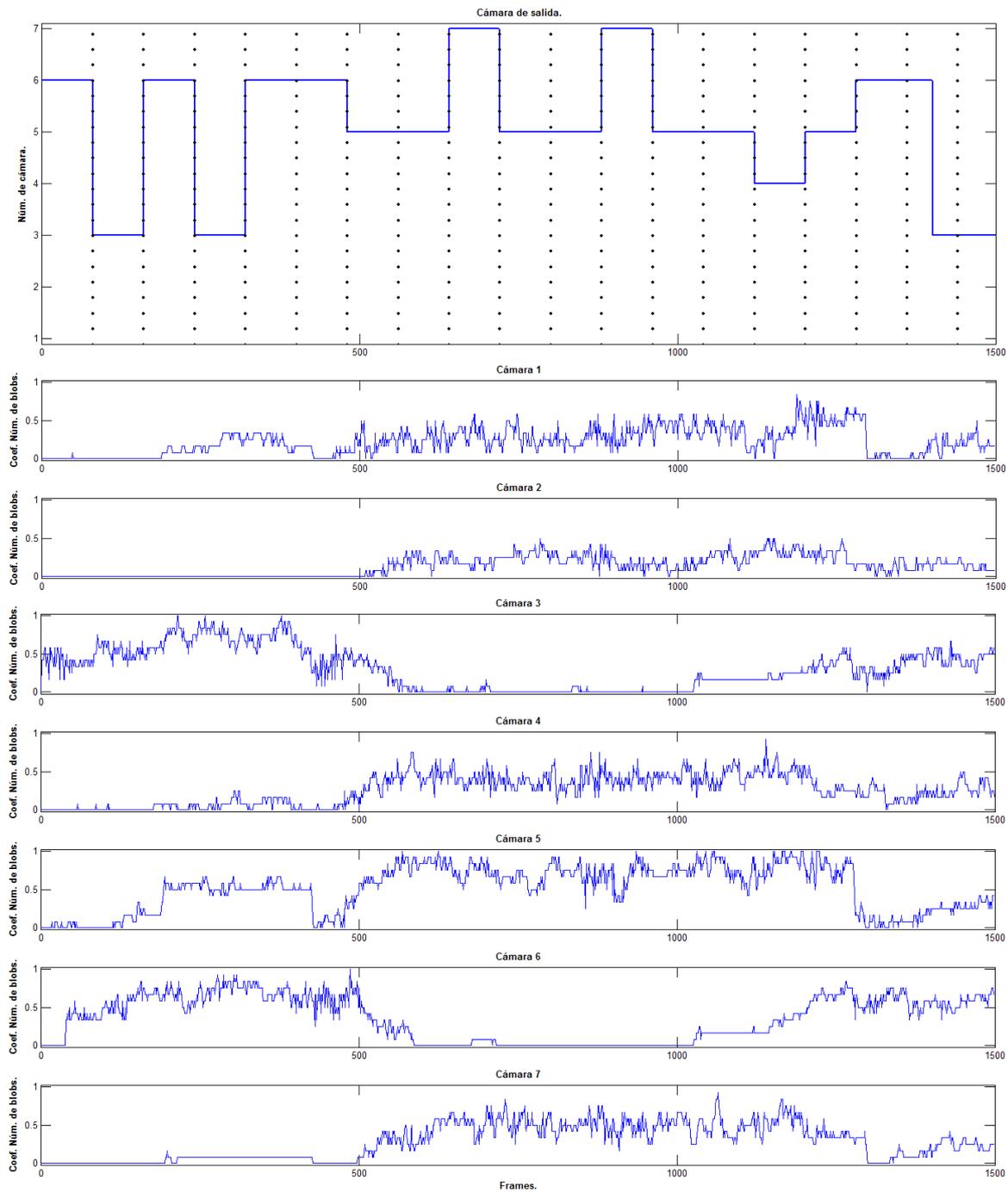


Figura B.1: Número de blobs

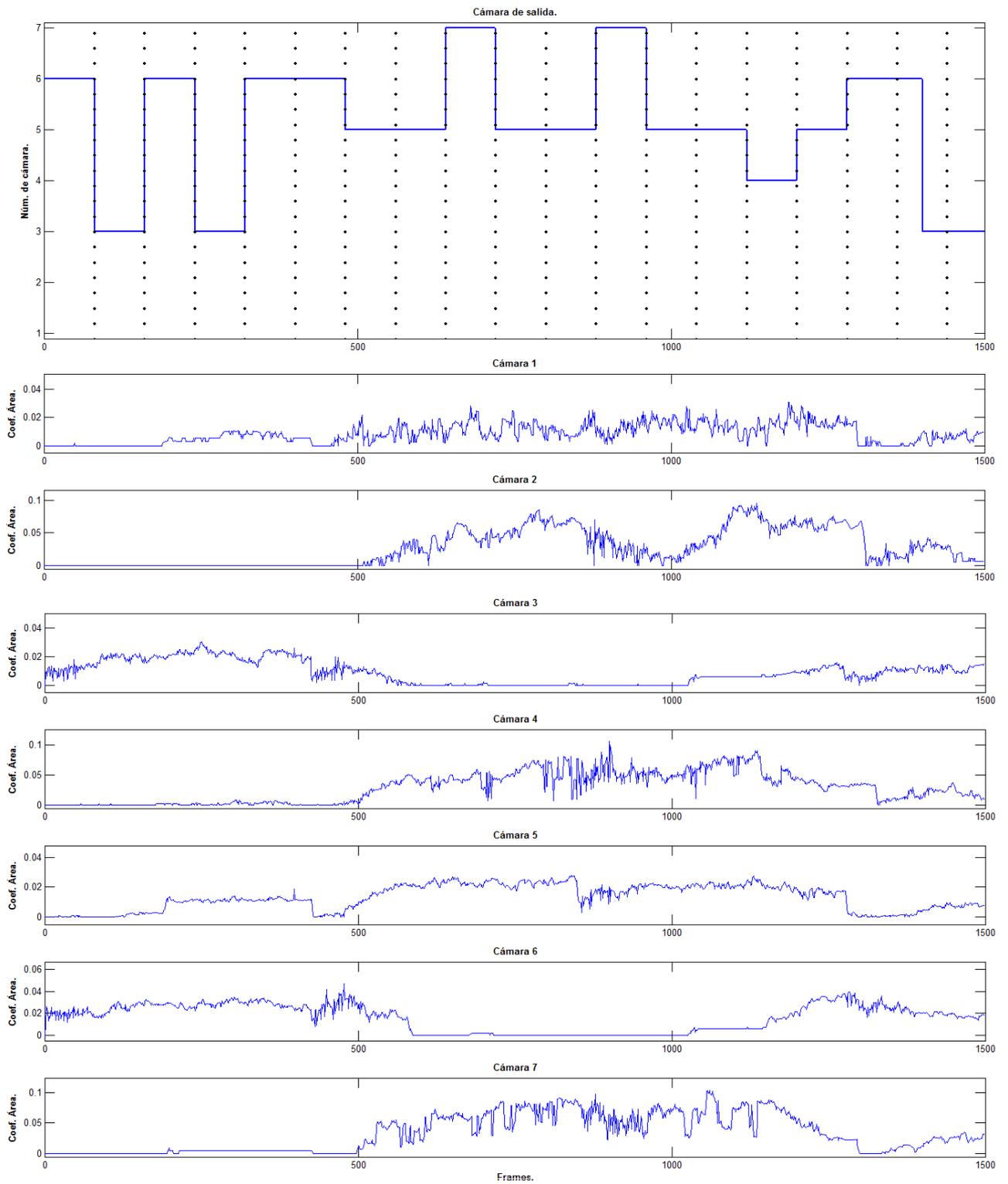


Figura B.2: Coeficiente de área

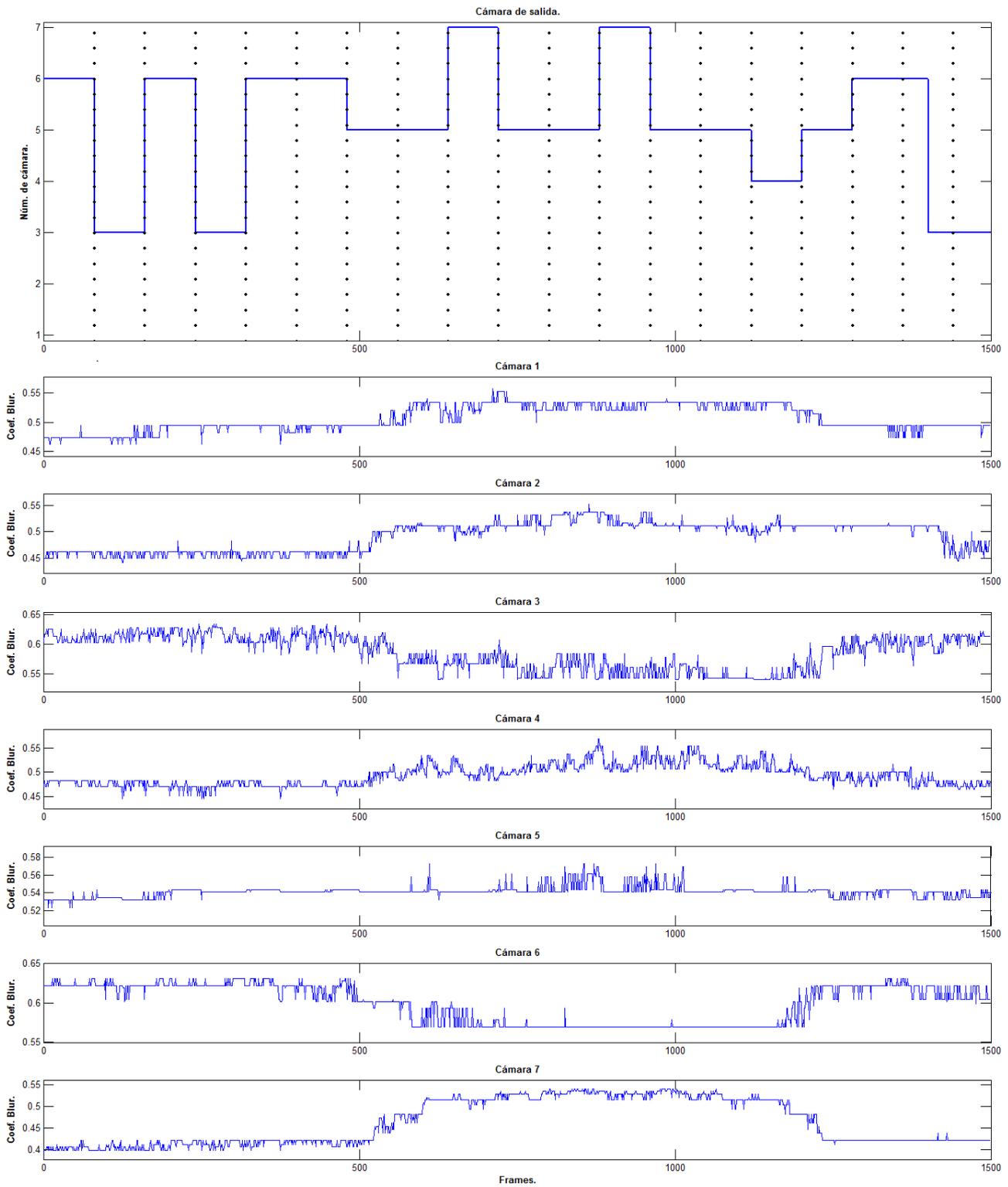


Figura B.3: Coeficiente de emborronamiento

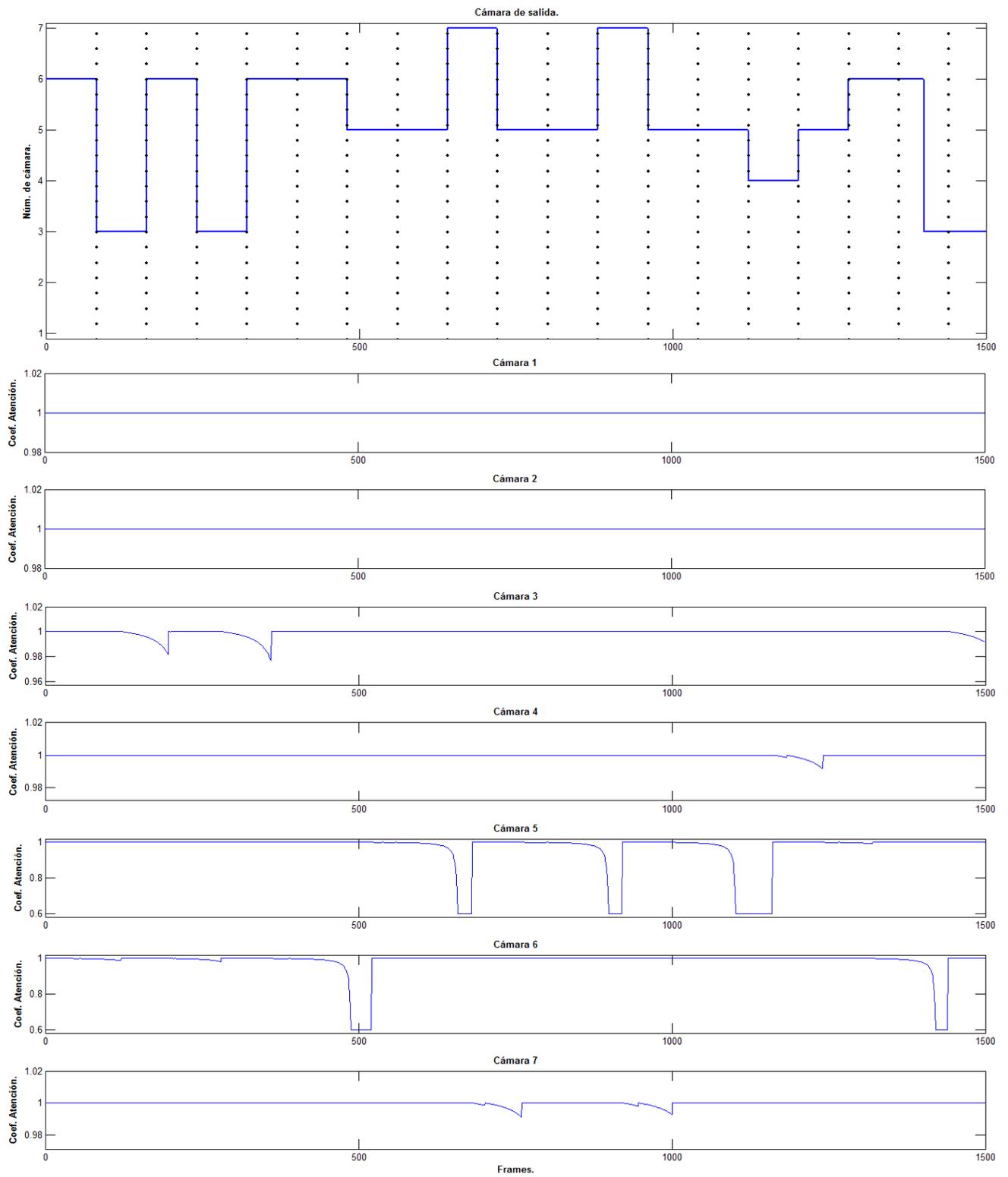


Figura B.4: Coeficiente de atención

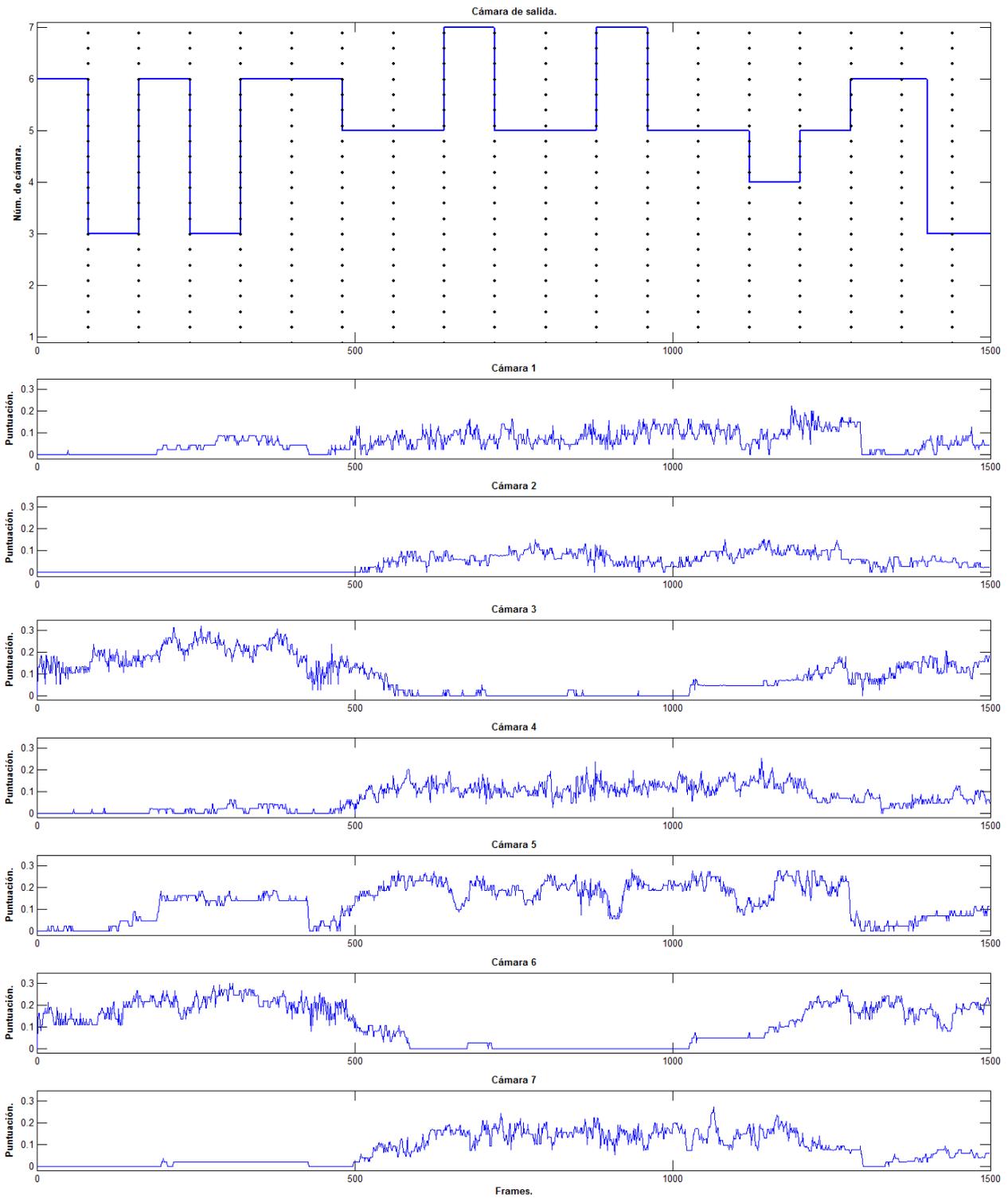


Figura B.5: Puntuación de las cámaras (vídeo completo)

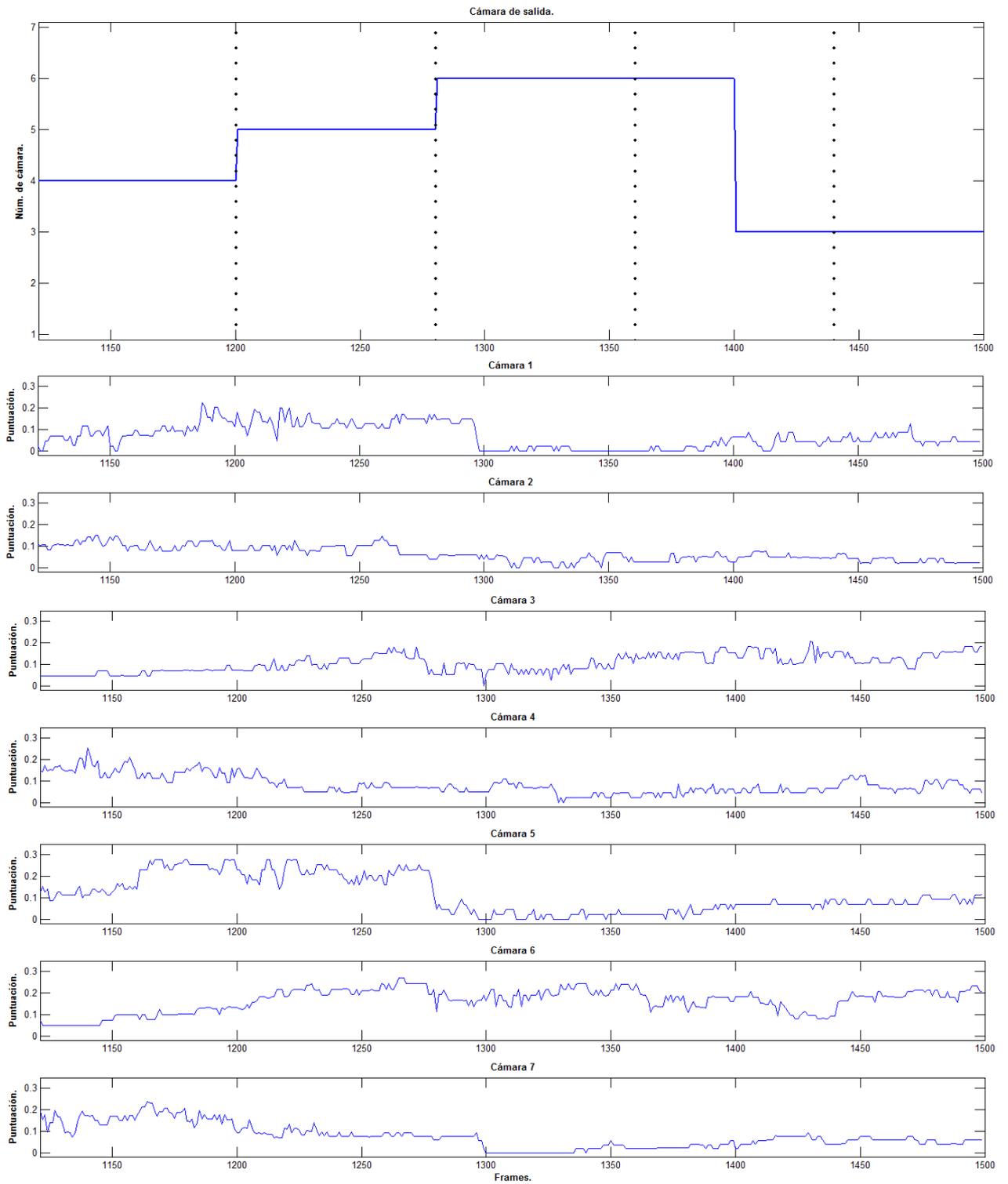


Figura B.6: Puntuación de las cámaras (parcial)

B.2.2. Baloncesto (SET2)

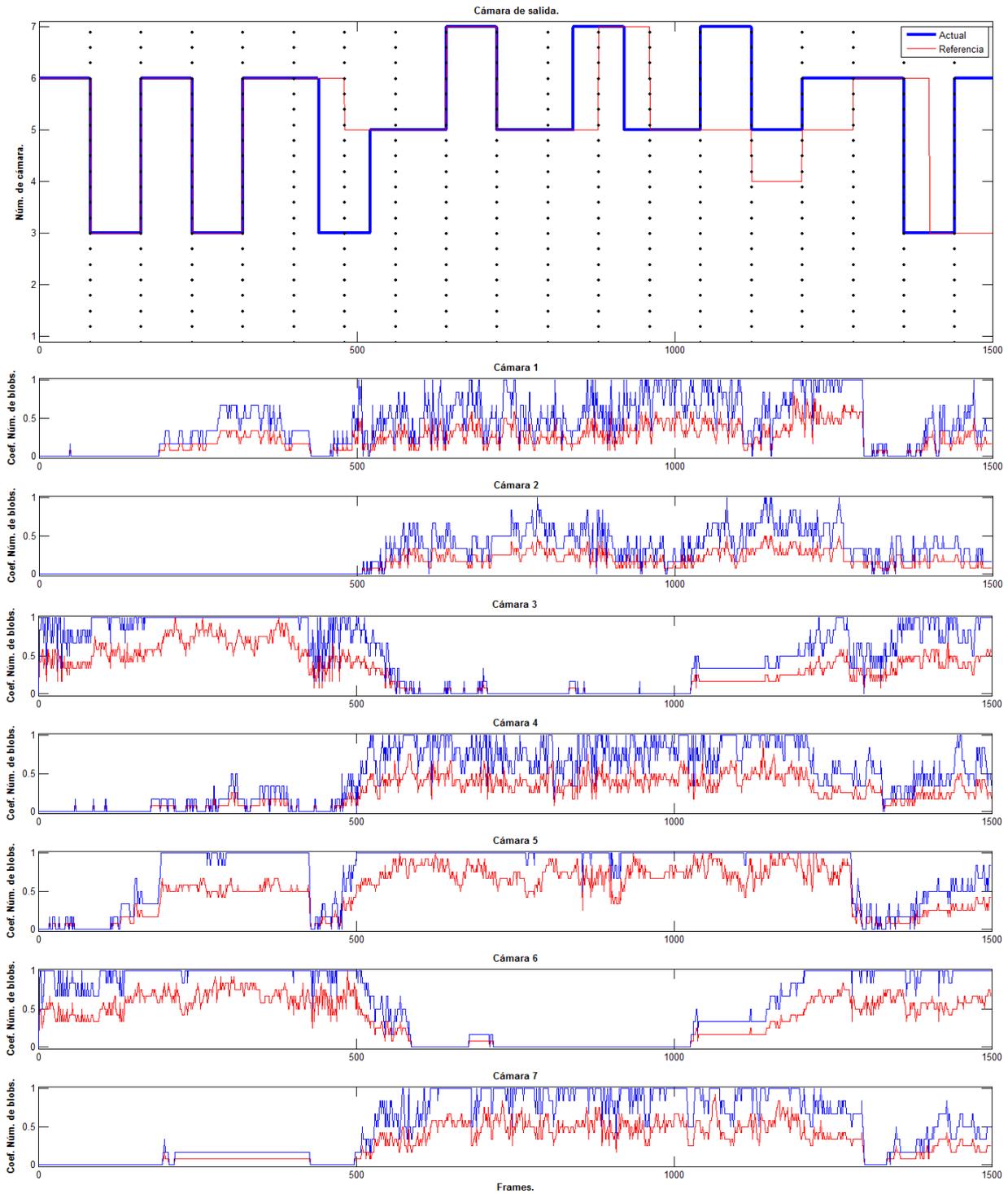


Figura B.7: Número de blobs. SET 2 vs SET 1

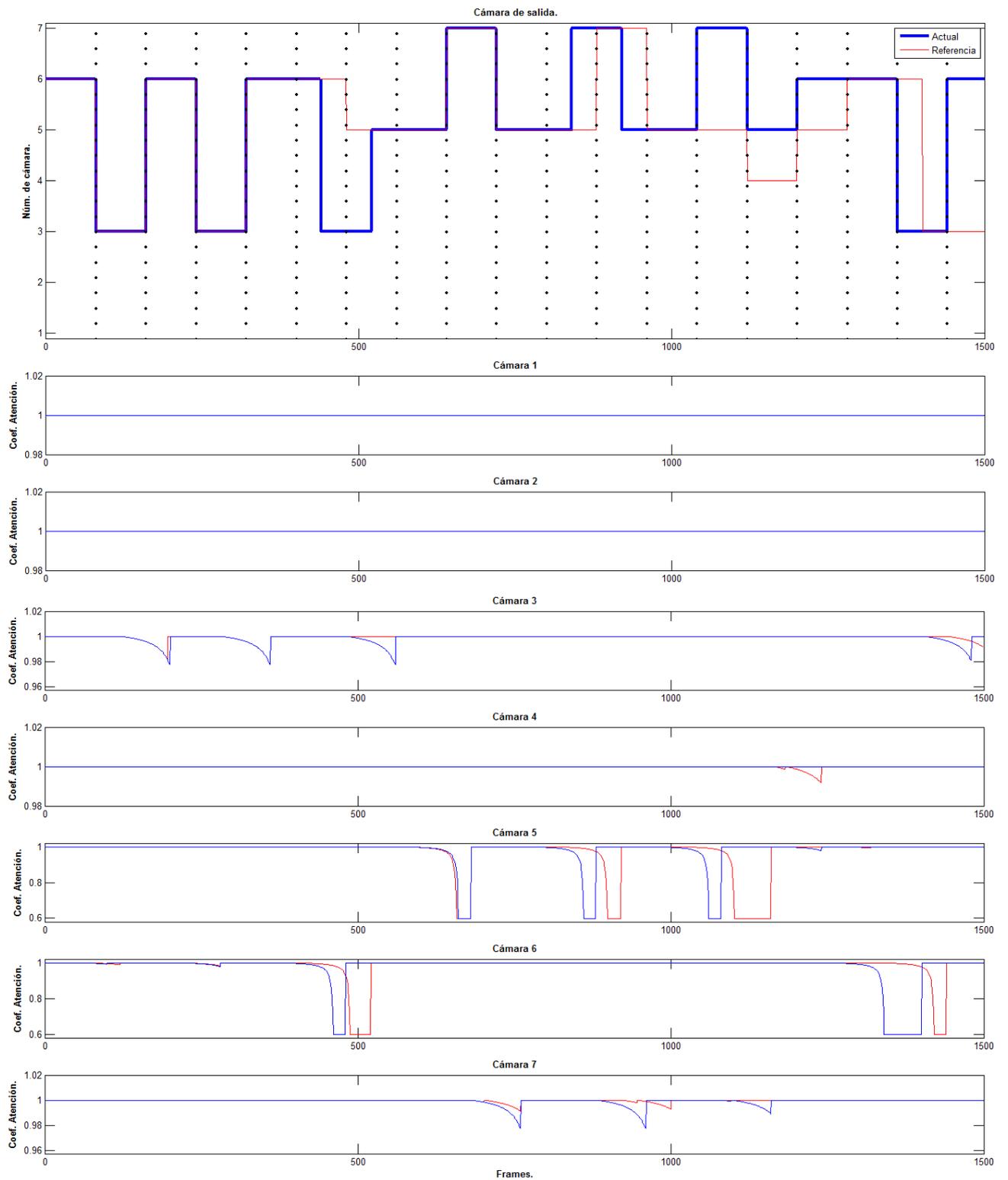


Figura B.8: Coeficiente de atención. SET 2 vs SET 1

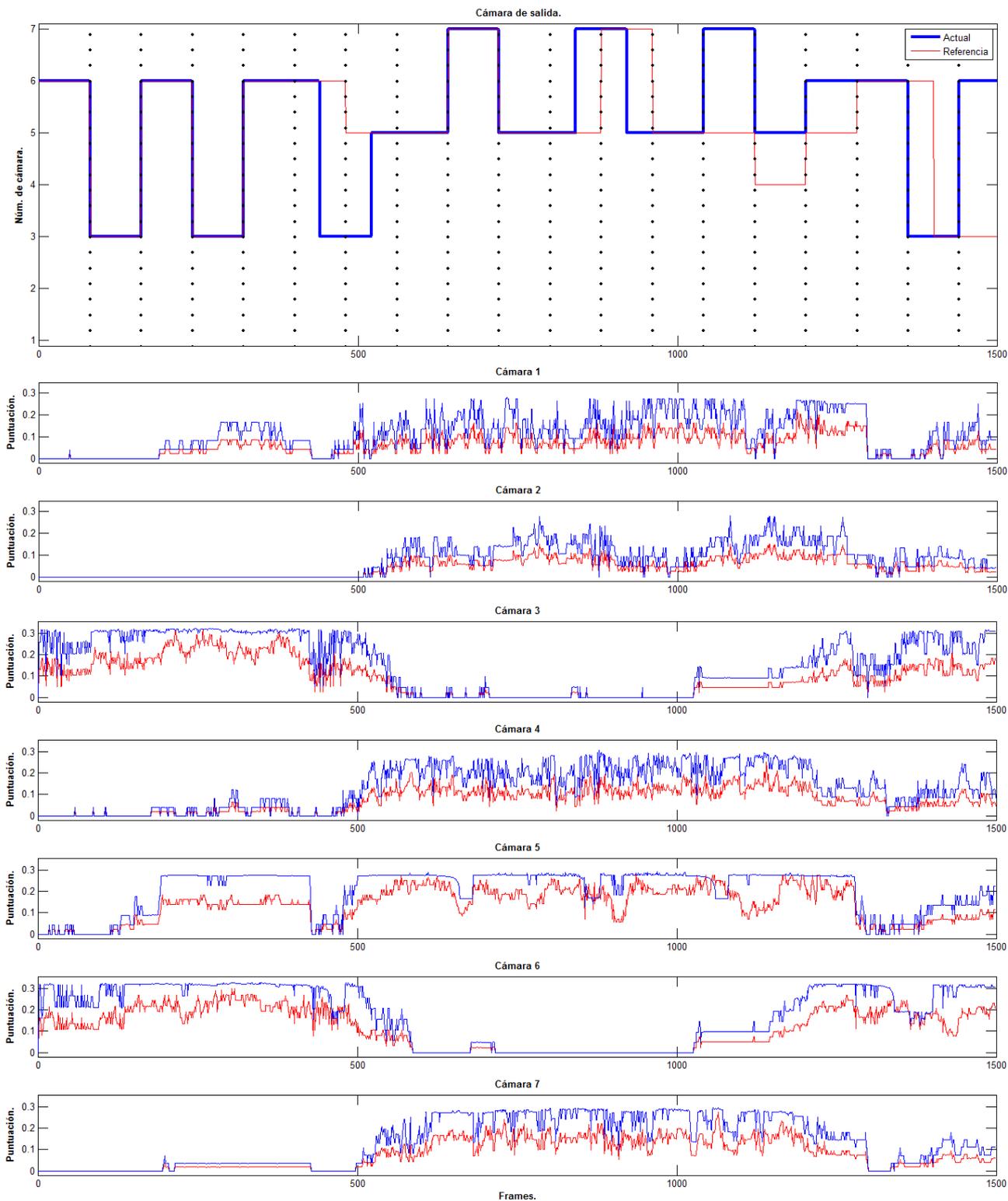


Figura B.9: Puntuación de las cámaras (vídeo completo). SET 2 vs SET 1

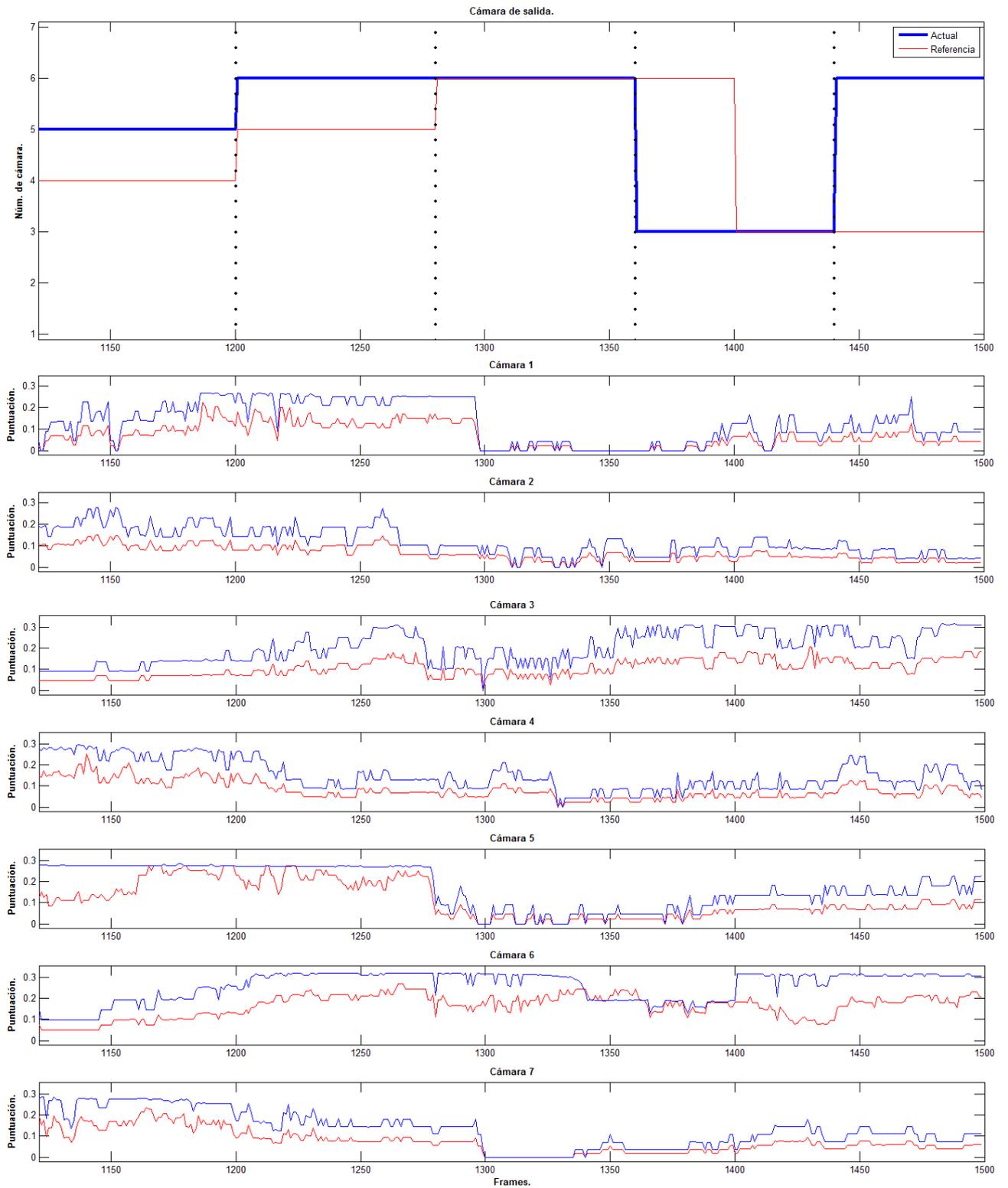


Figura B.10: Puntuación de las cámaras (parcial). SET 2 vs SET 1

B.2.3. Baloncesto (SET3)

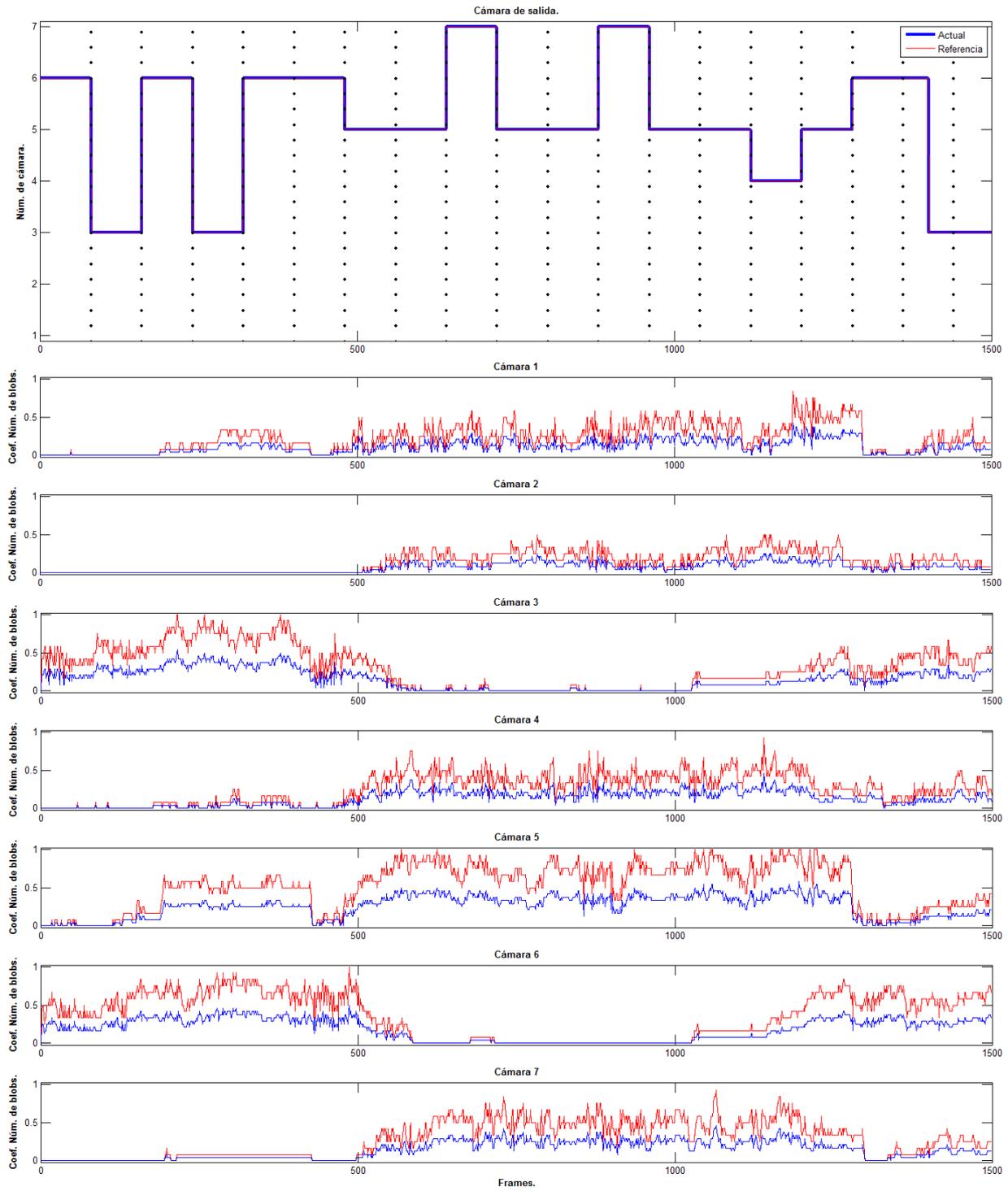


Figura B.11: Número de blobs. SET 3 vs SET 1

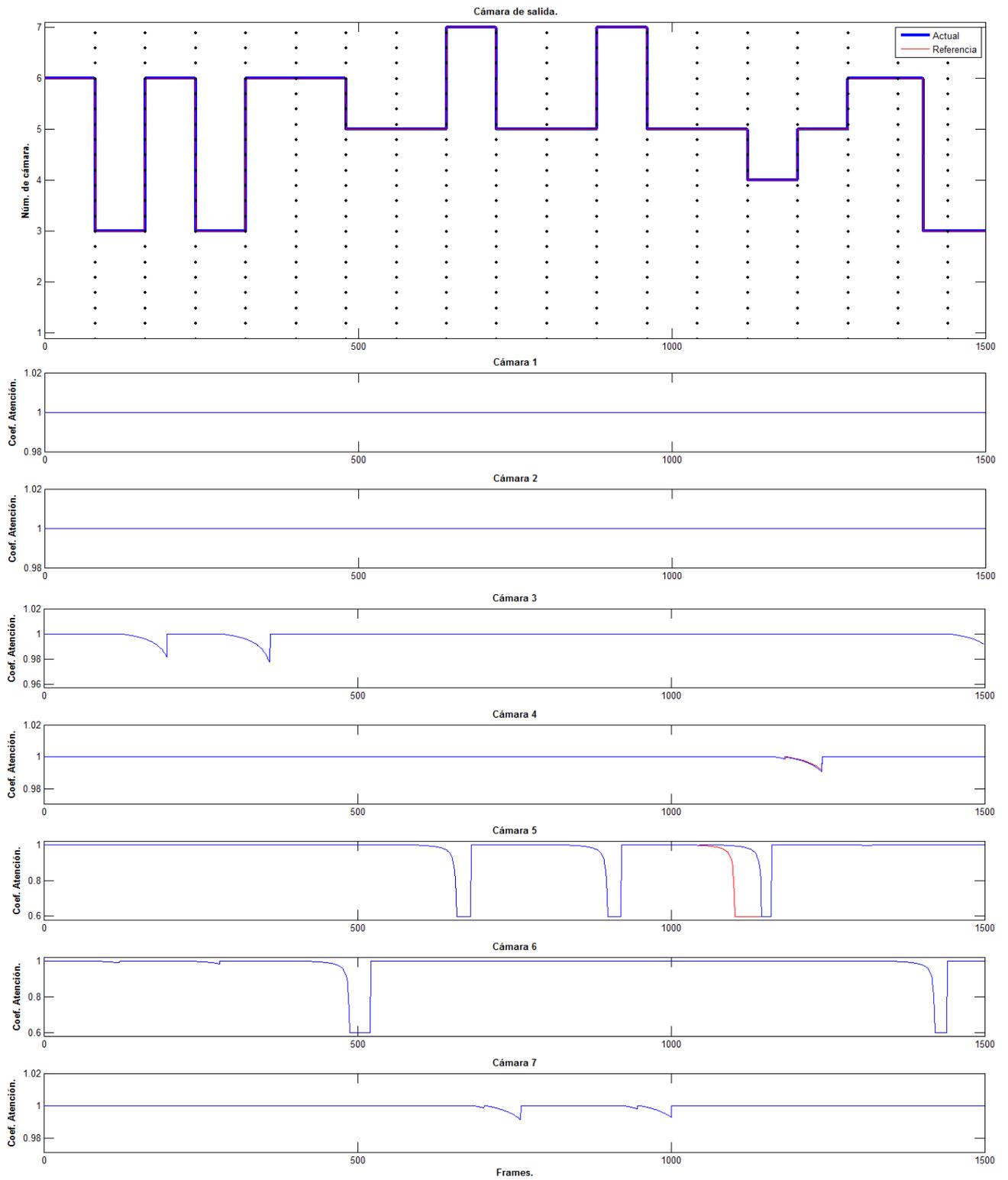


Figura B.12: Coeficiente de atención. SET 3 vs SET 1

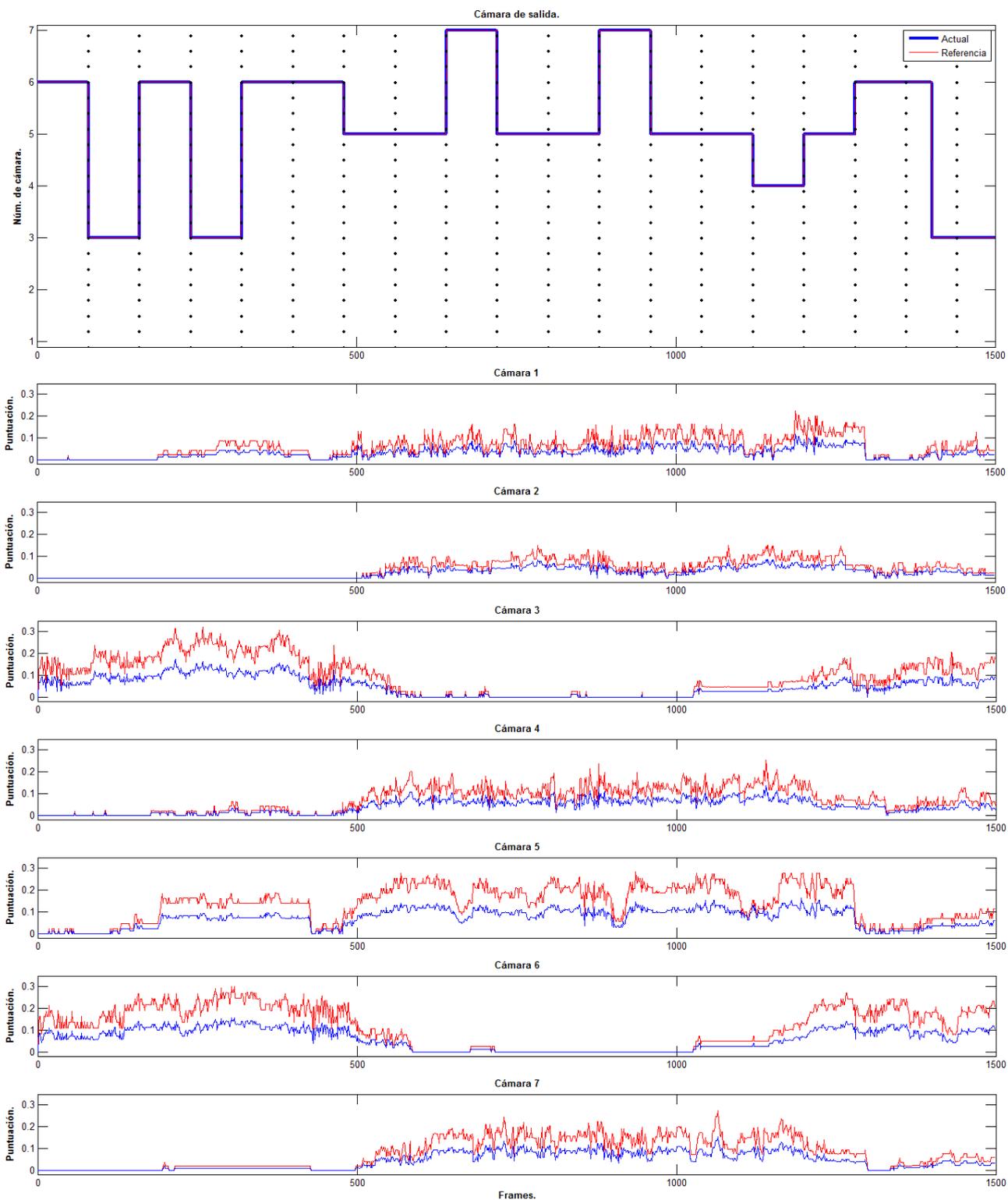


Figura B.13: Puntuación de las cámaras (vídeo completo). SET 3 vs SET 1

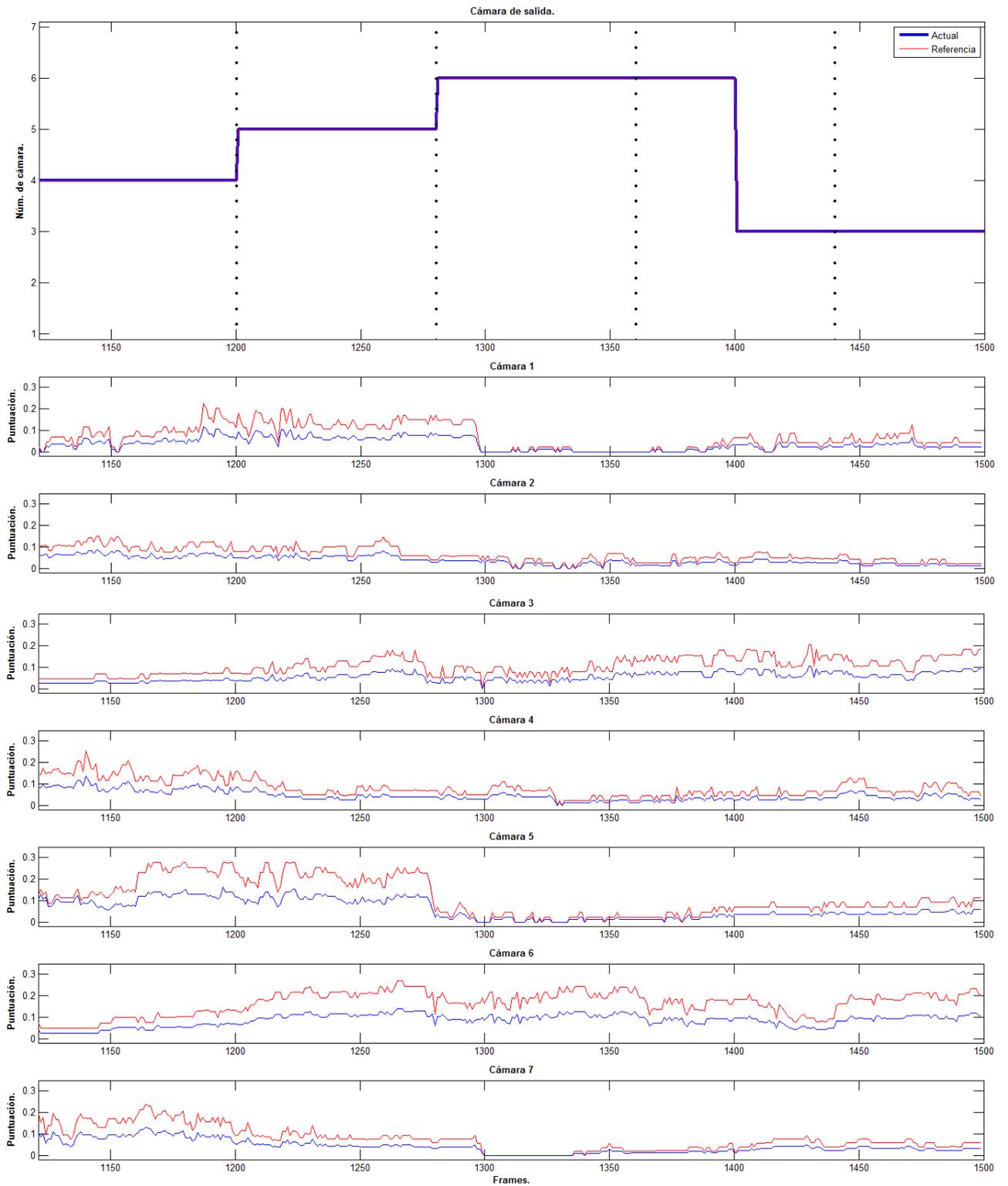


Figura B.14: Puntuación de las cámaras (parcial). SET 3 vs SET 1

B.2.4. Baloncesto (SET4)

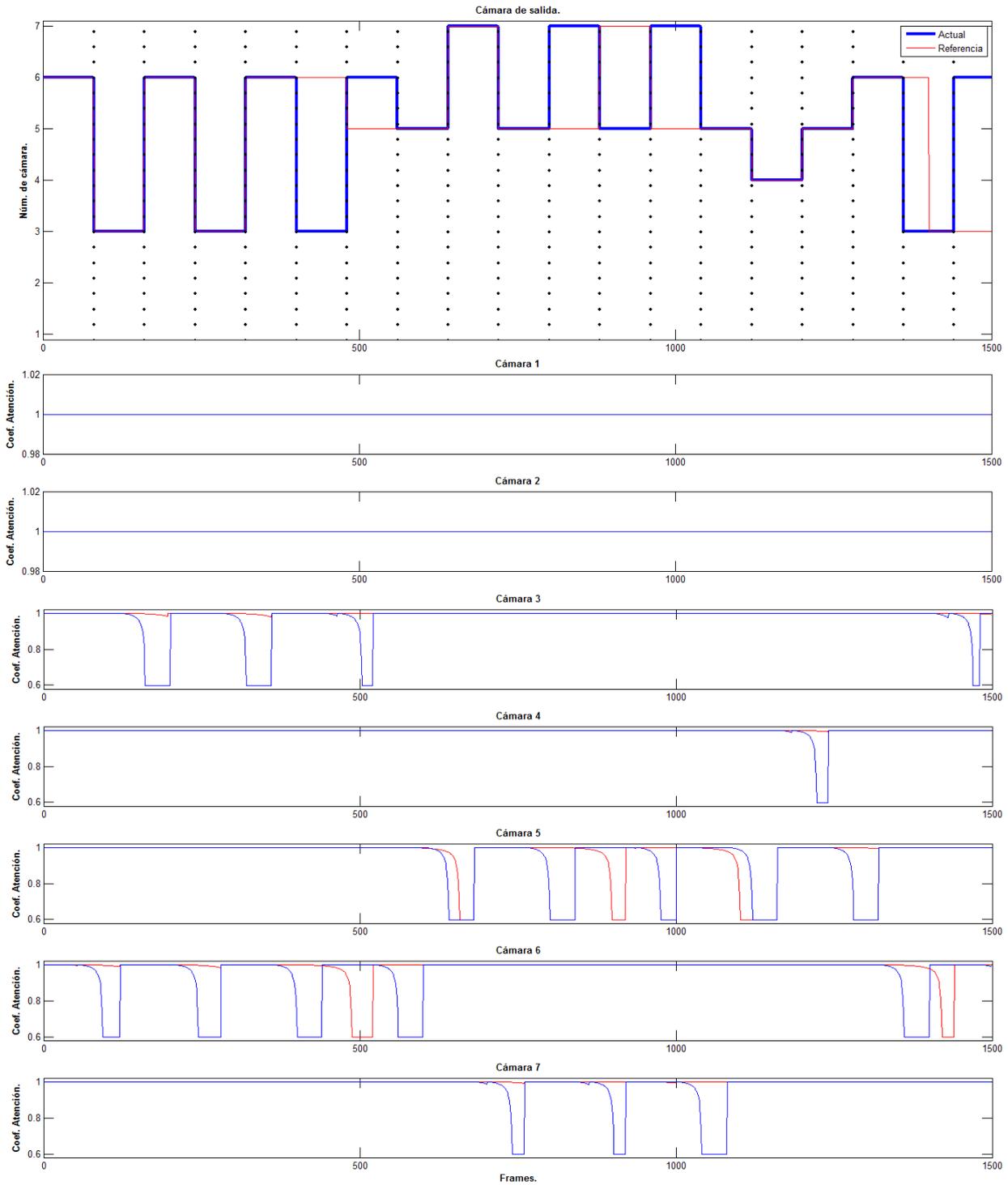


Figura B.15: Coeficiente de atención. SET 4 vs SET 1

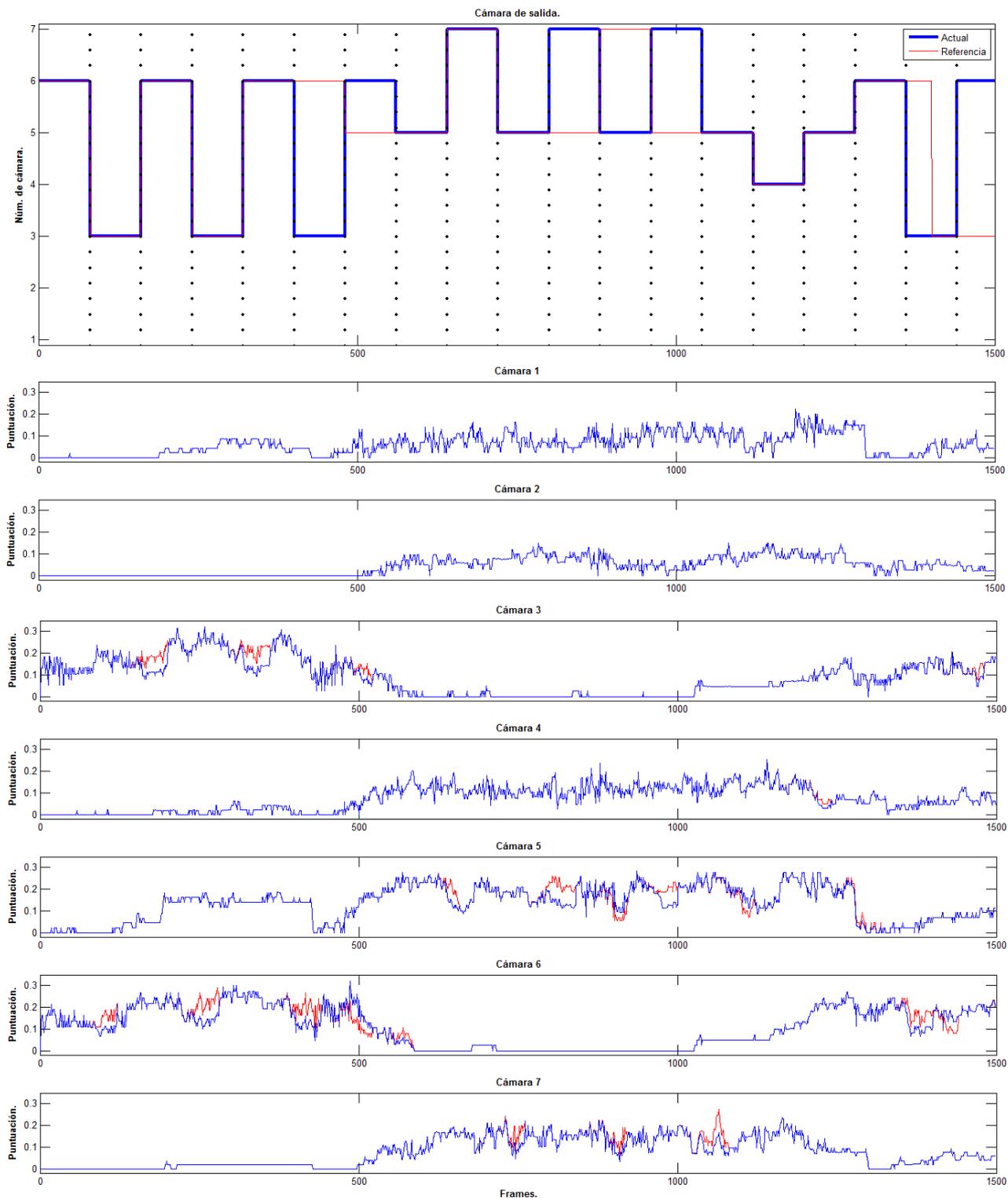


Figura B.16: Puntuación de las cámaras (vídeo completo). SET 4 vs SET 1

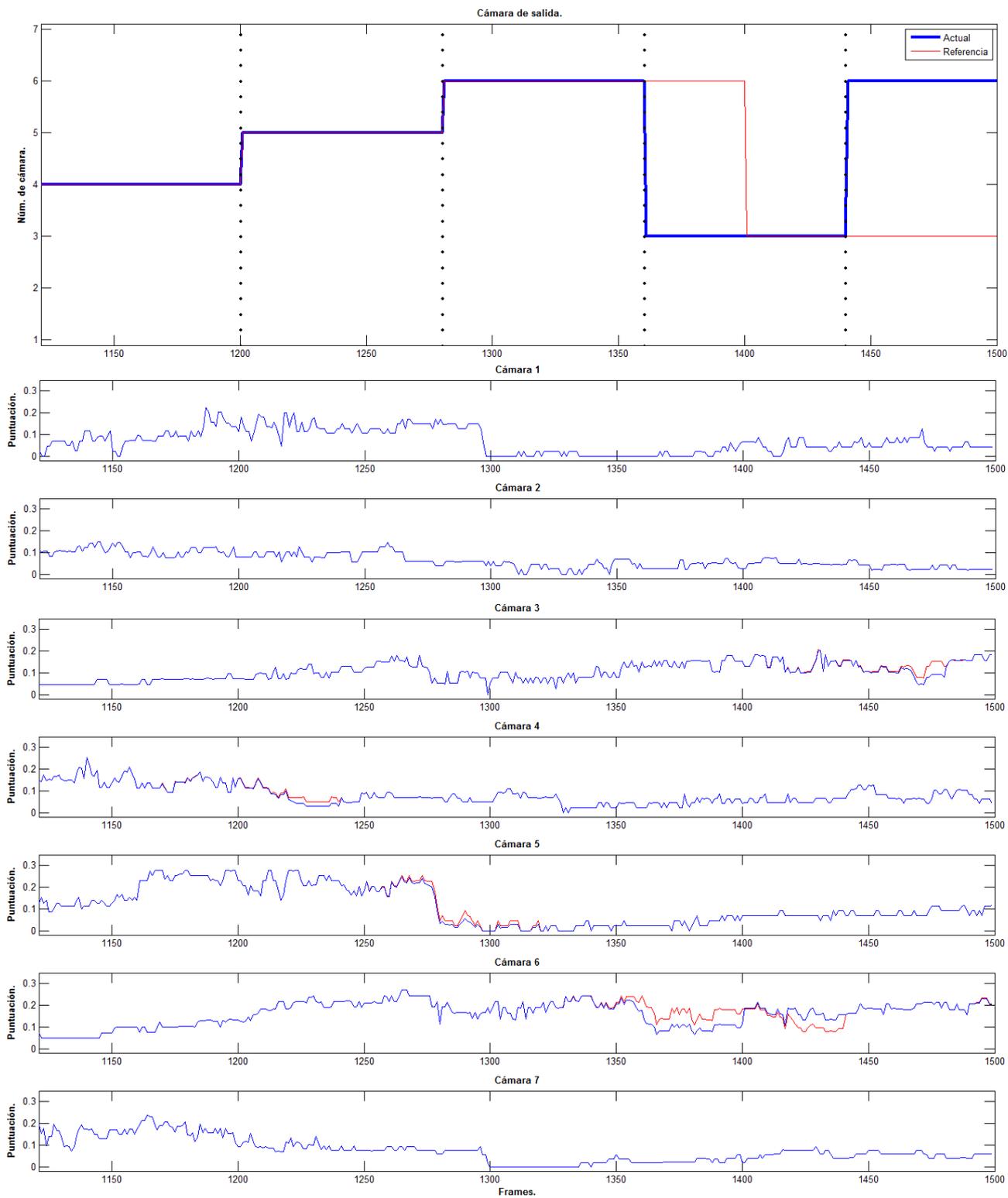


Figura B.17: Puntuación de las cámaras (parcial). SET 4 vs SET 1

B.2.5. Baloncesto (SET5)

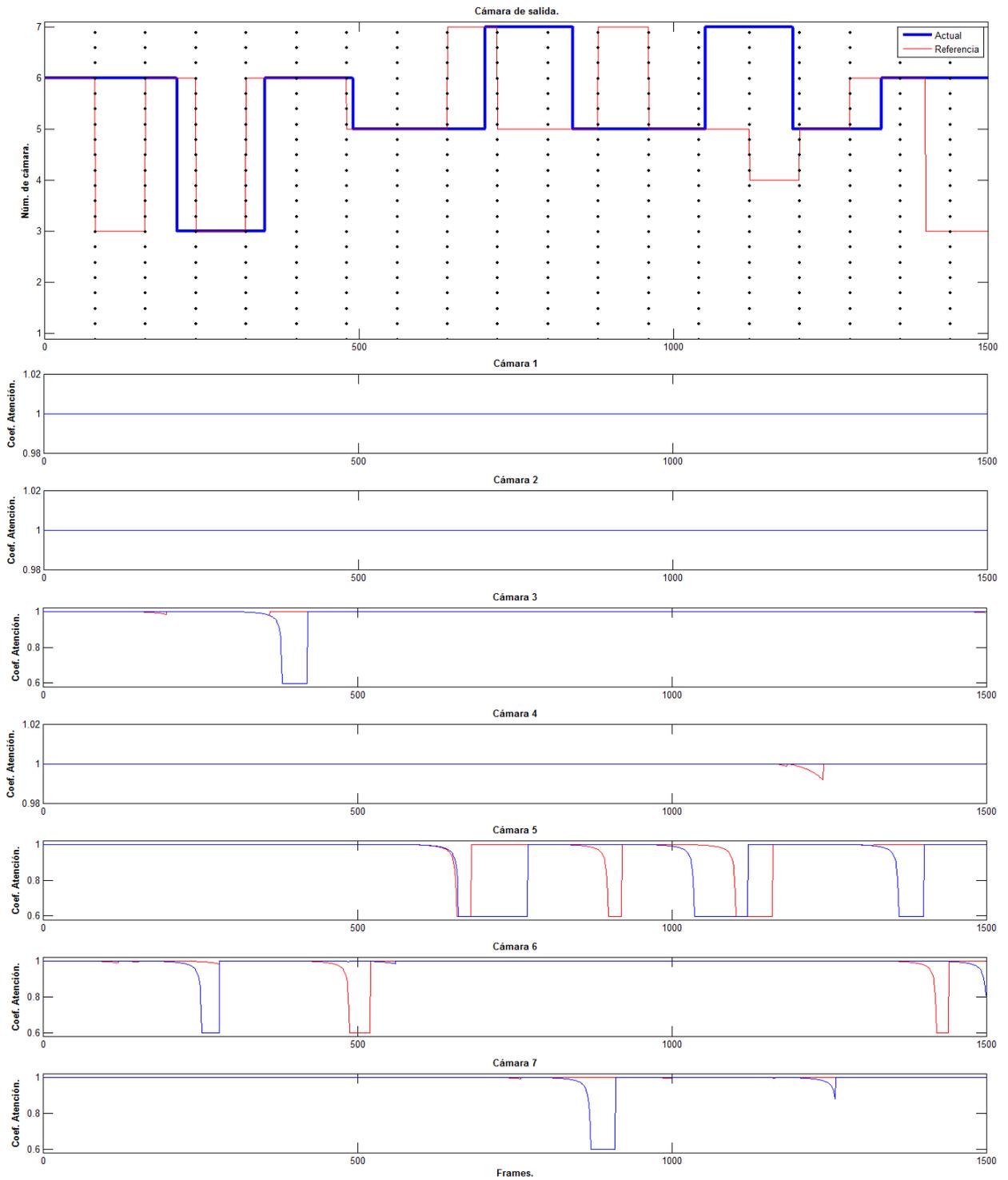


Figura B.18: Coeficiente de atención. SET 5 vs SET 1

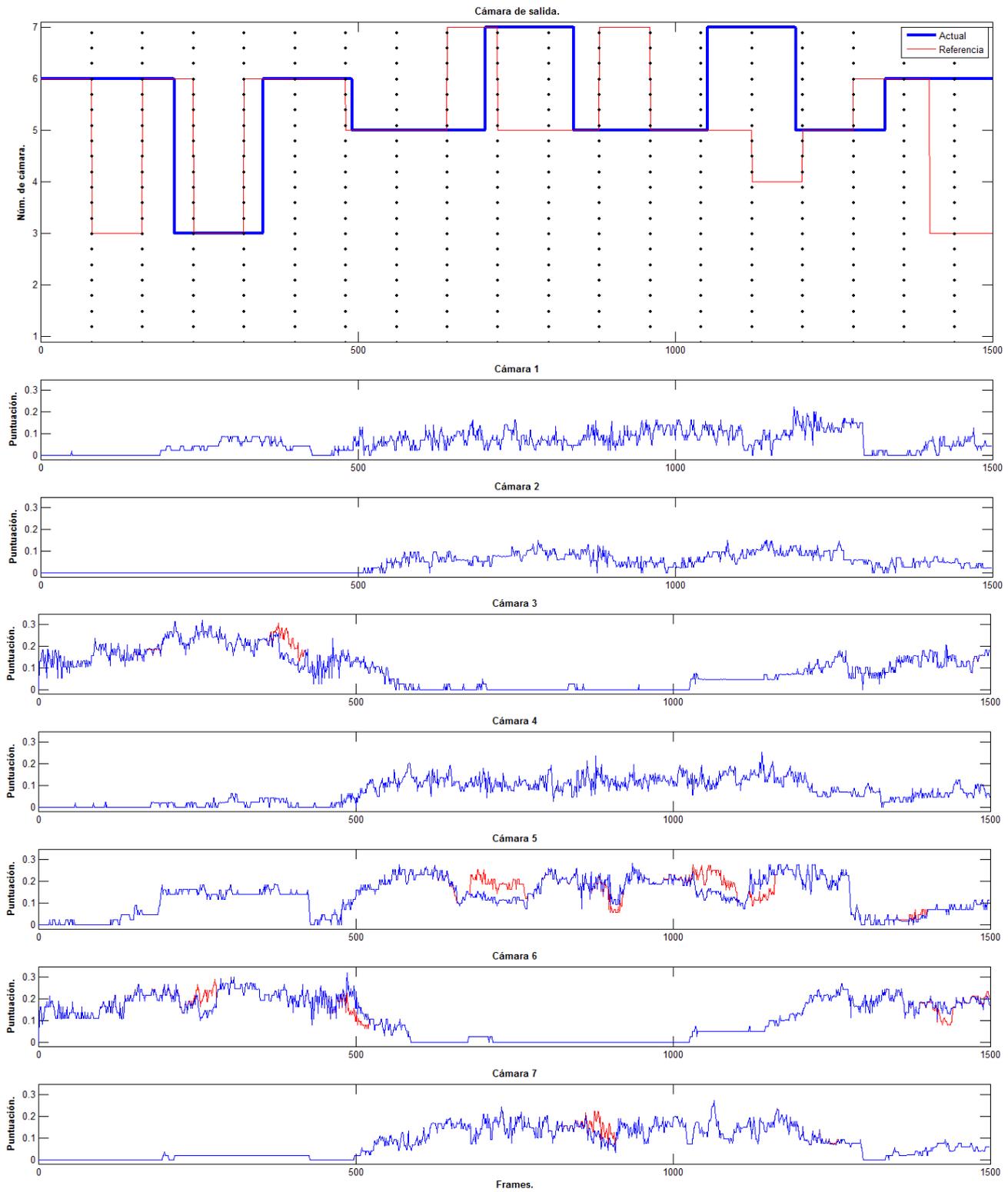


Figura B.19: Puntuación de las cámaras (vídeo completo). SET 5 vs SET 1

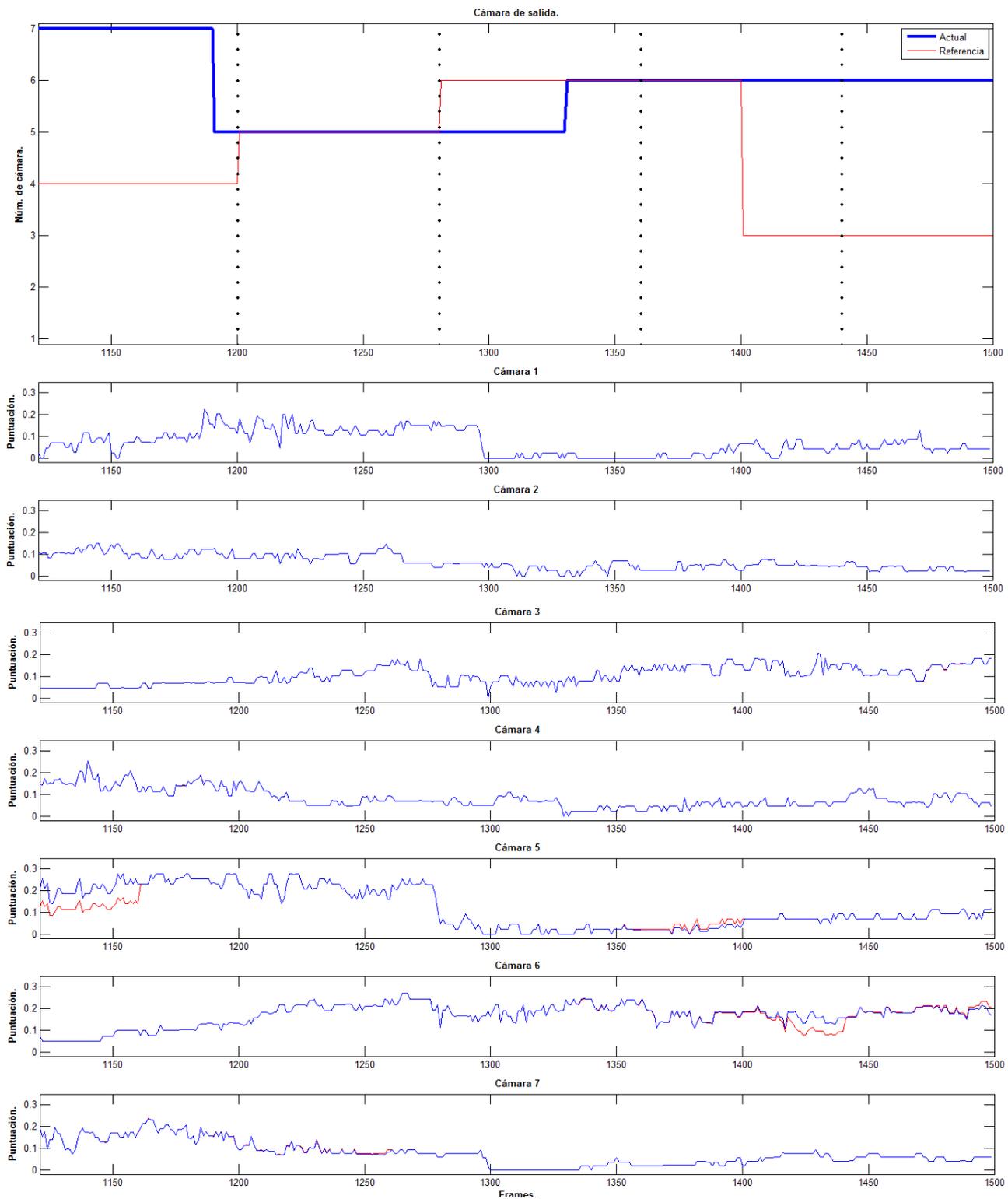


Figura B.20: Puntuación de las cámaras (parcial). SET 5 vs SET 1

B.2.6. Baloncesto (SET6)

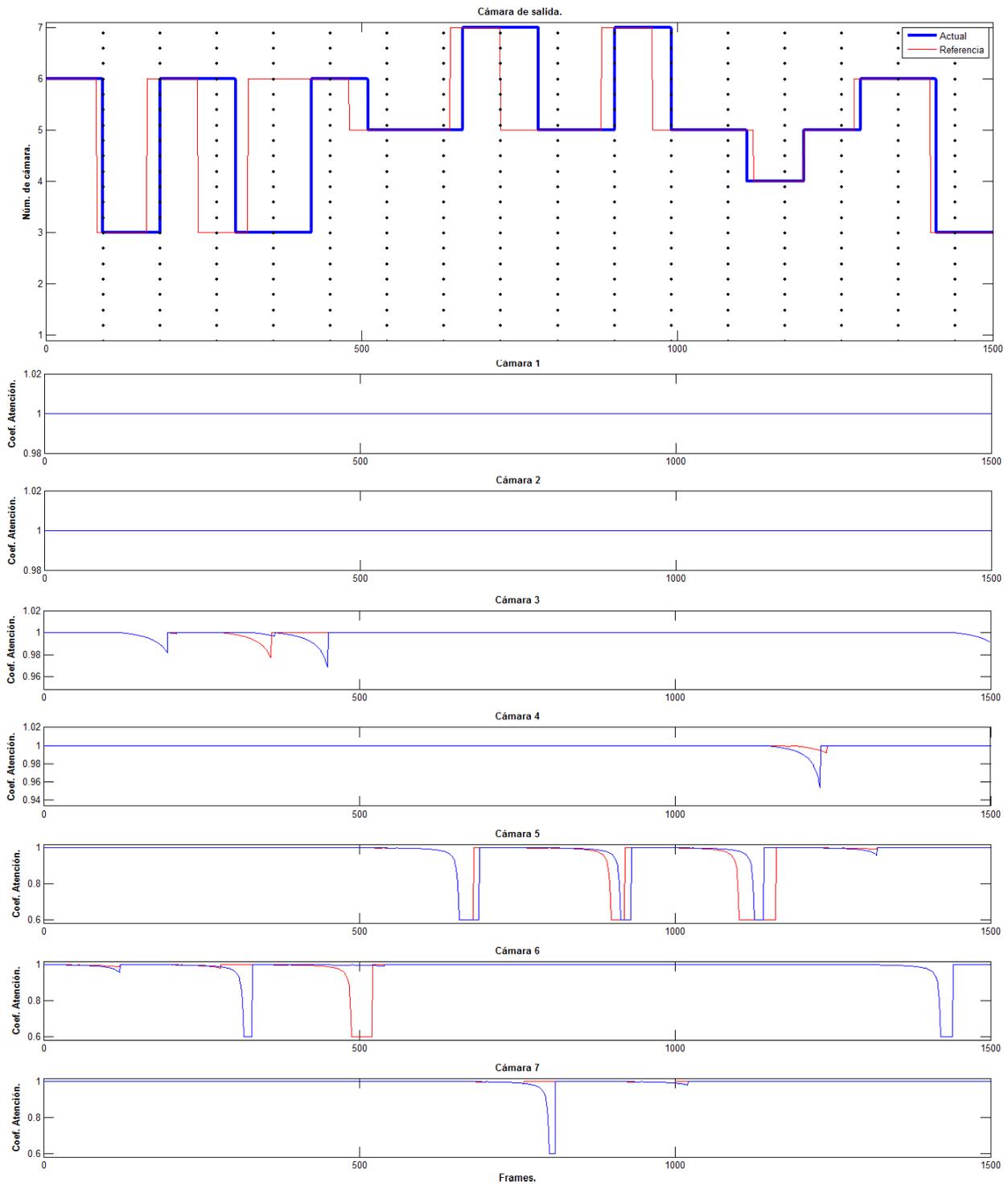


Figura B.21: Coeficiente de atención. SET 6 vs SET 1

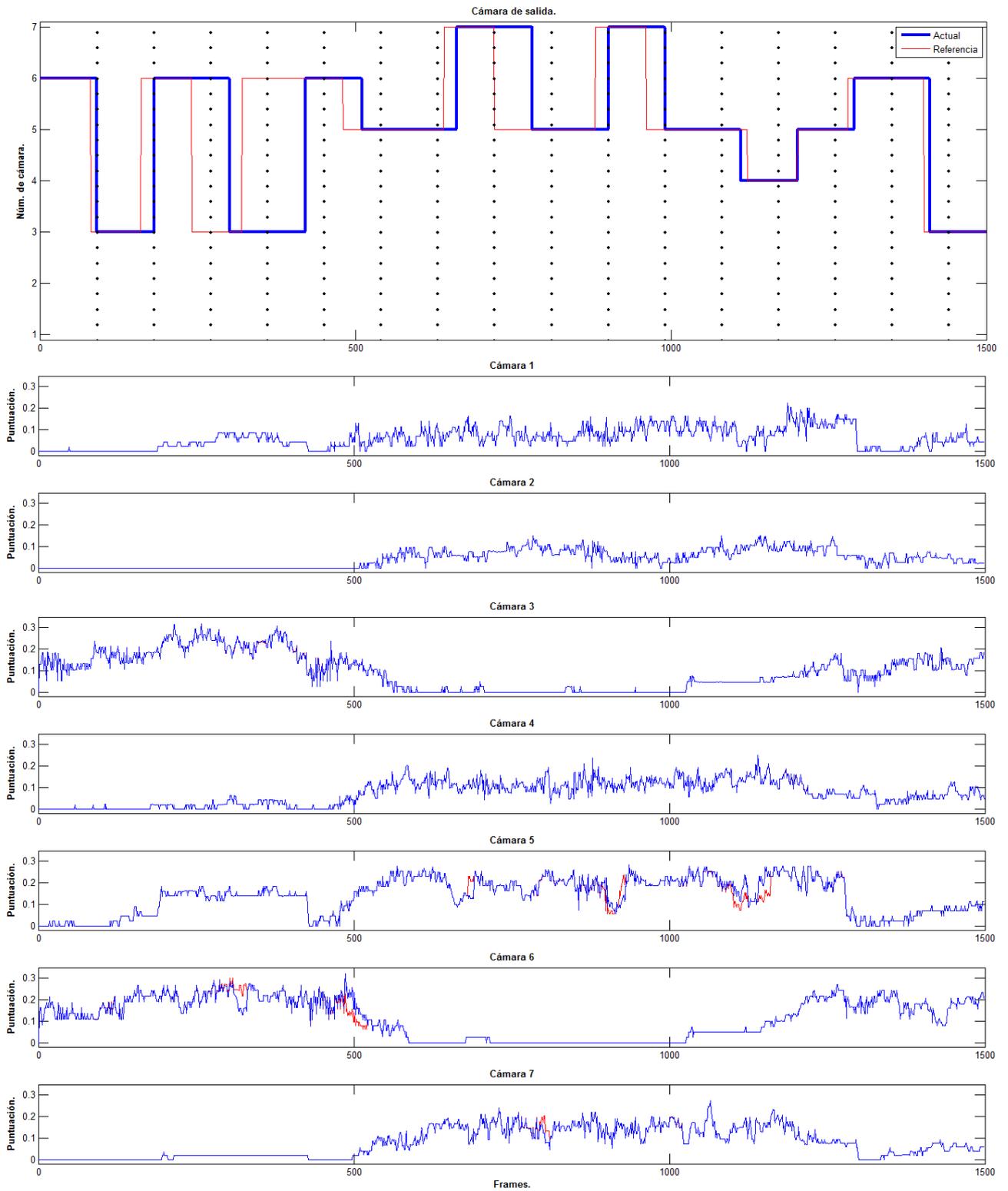


Figura B.22: Puntuación de las cámaras (vídeo completo). SET 6 vs SET 1

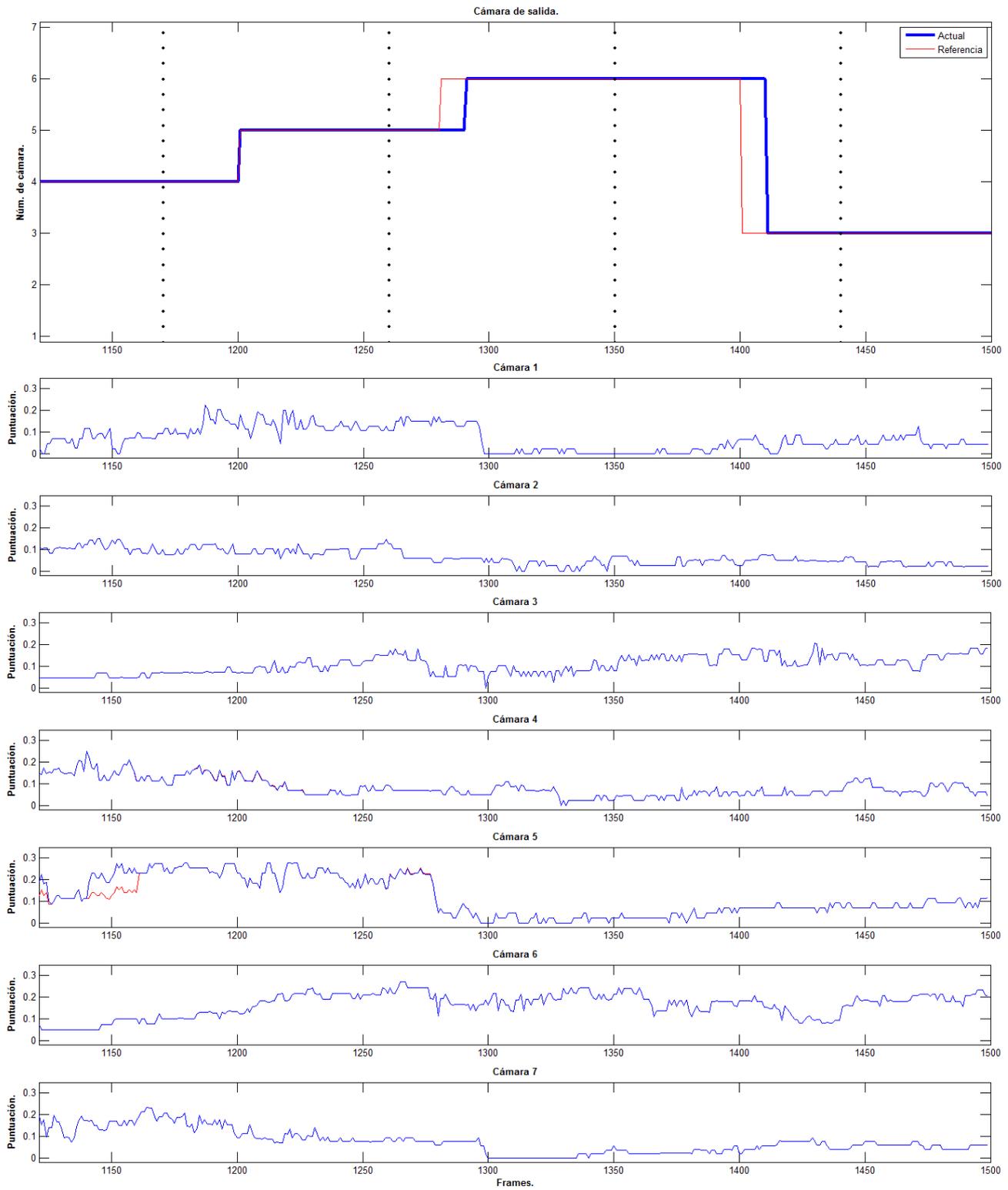


Figura B.23: Puntuación de las cámaras (parcial). SET 6 vs SET 1

B.3. Resultados para distintos dataset

B.3.1. Fútbol

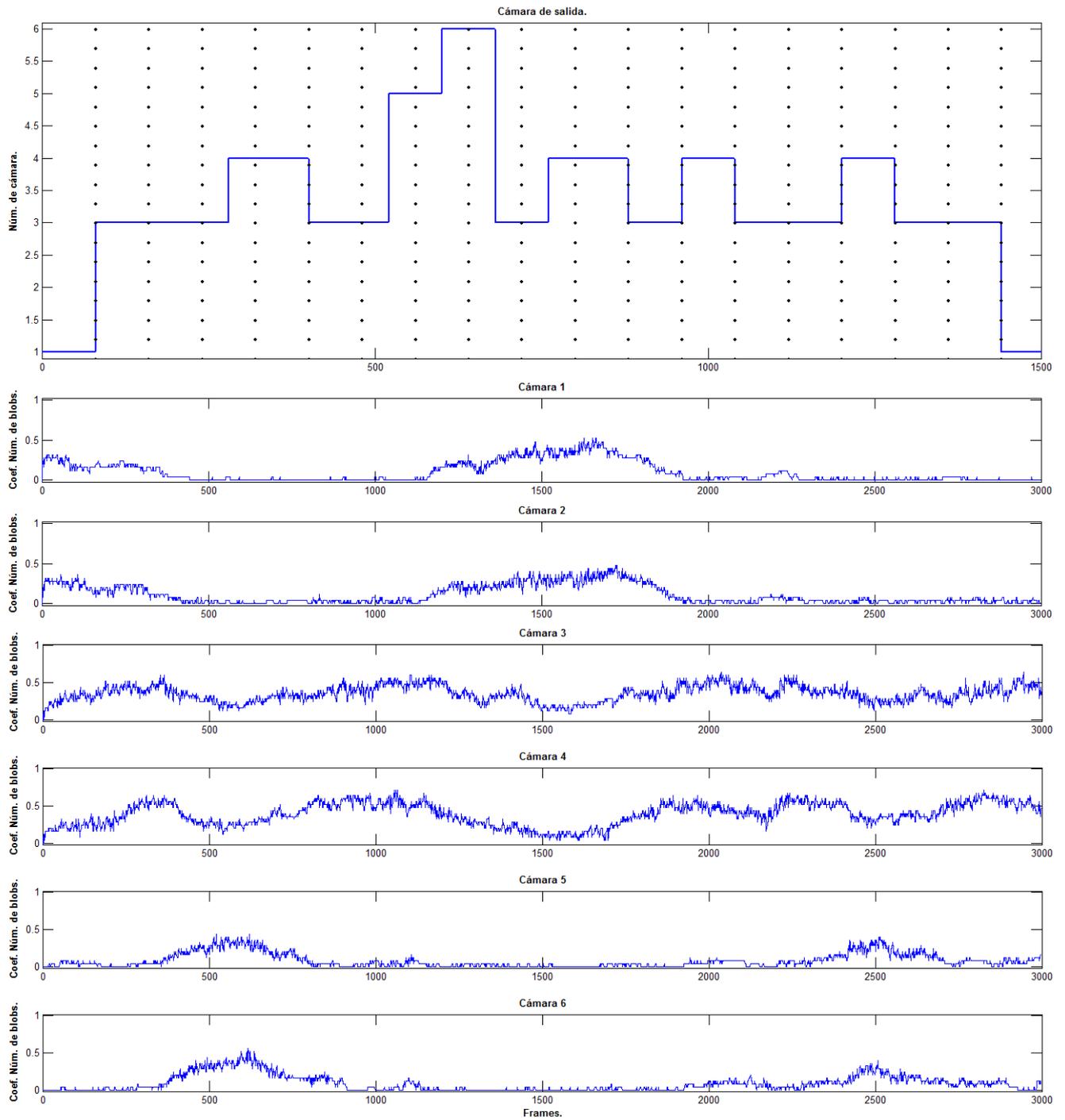


Figura B.24: Número de blobs

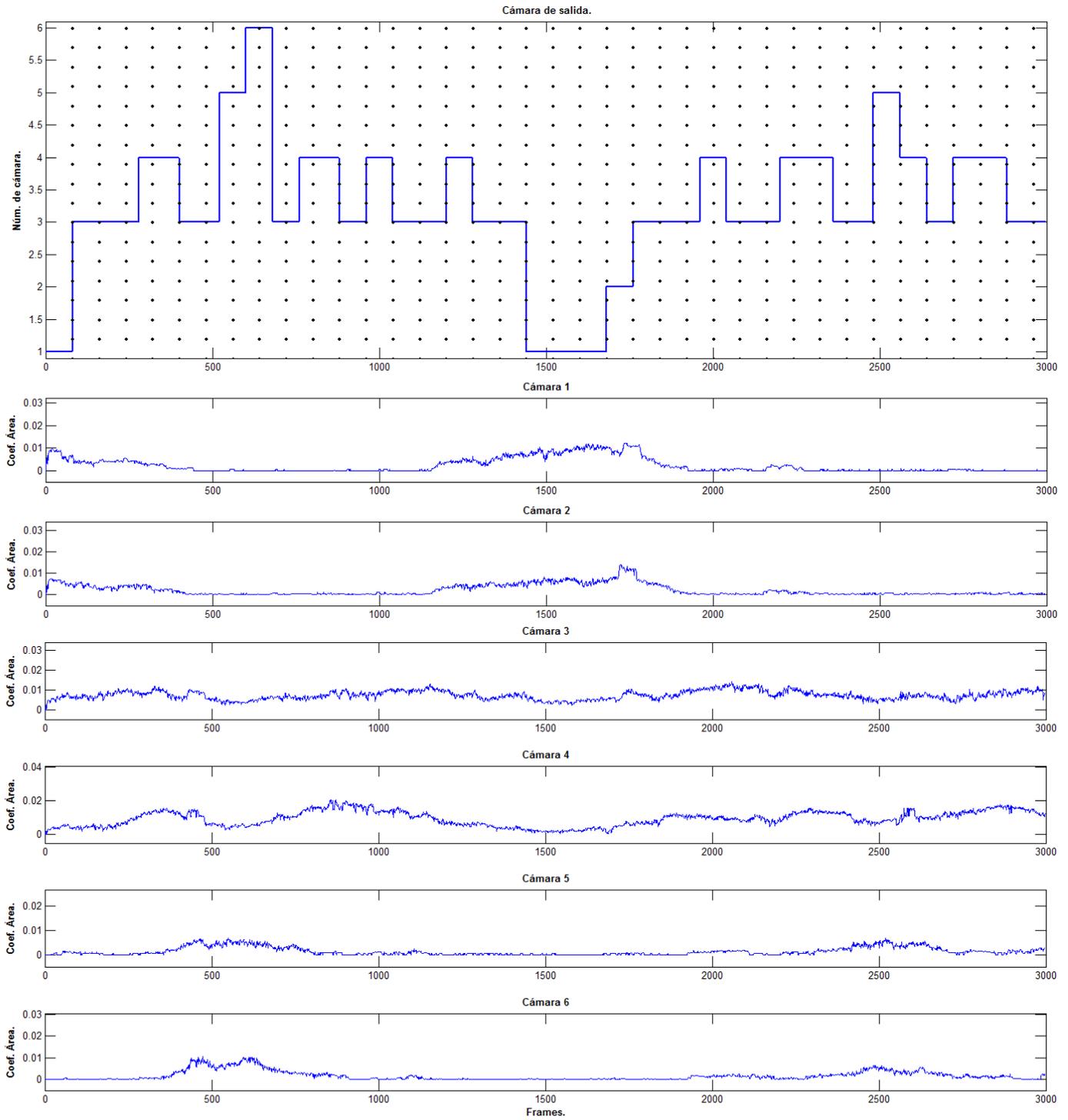


Figura B.25: Coeficiente de área

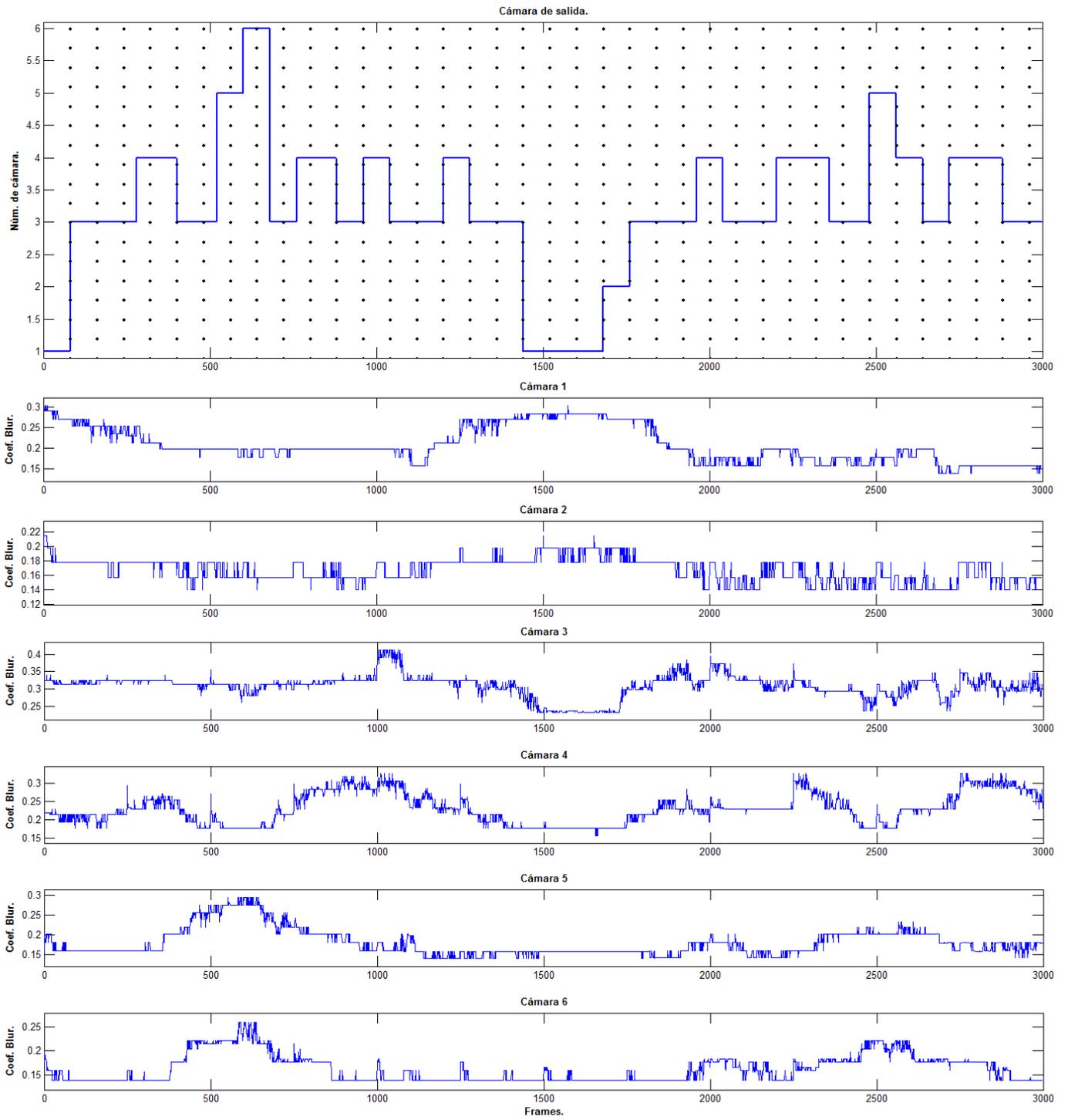


Figura B.26: Coeficiente de emborronamiento

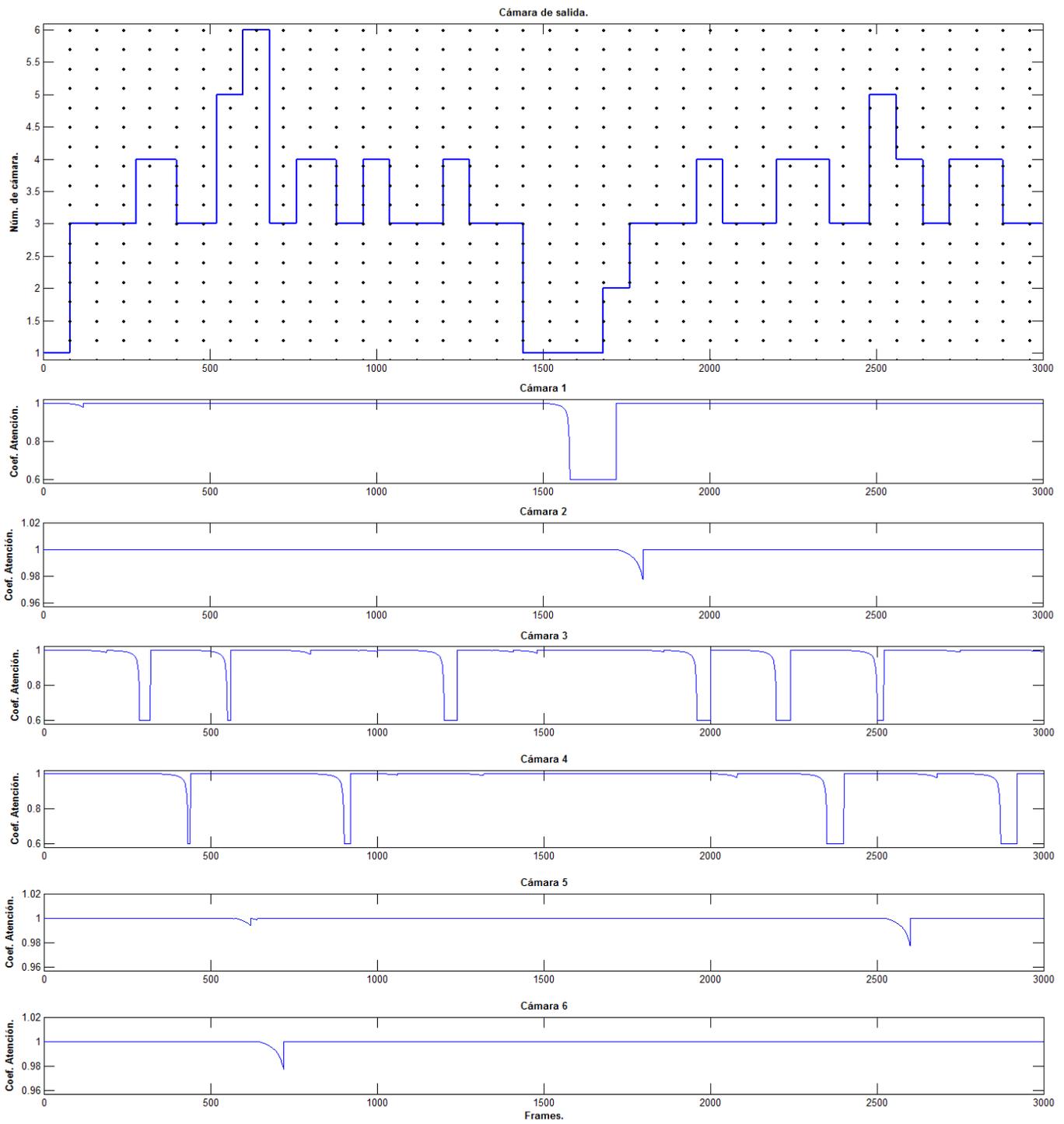


Figura B.27: Coeficiente de atención

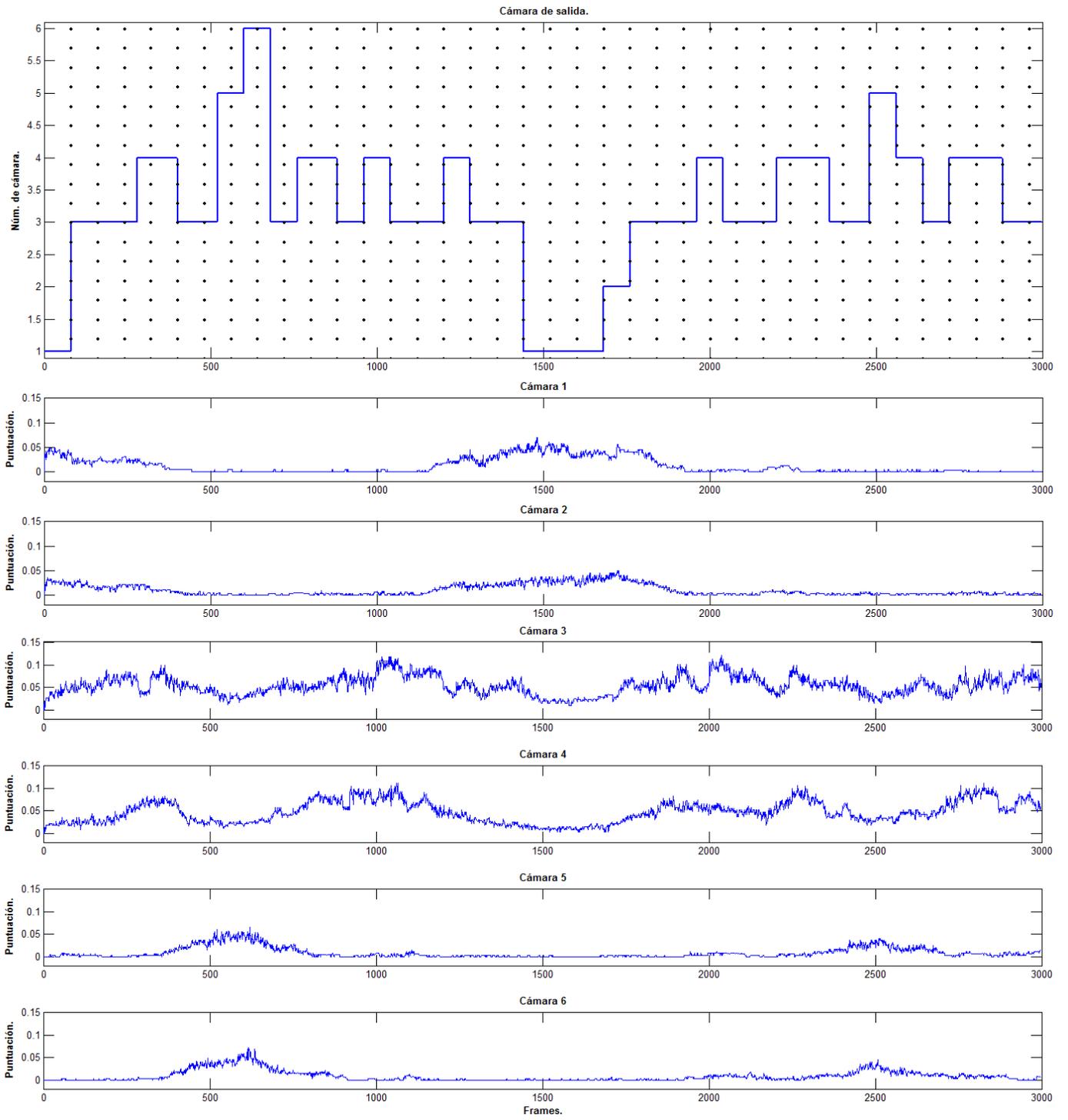


Figura B.28: Puntuación de las cámaras (vídeo completo)

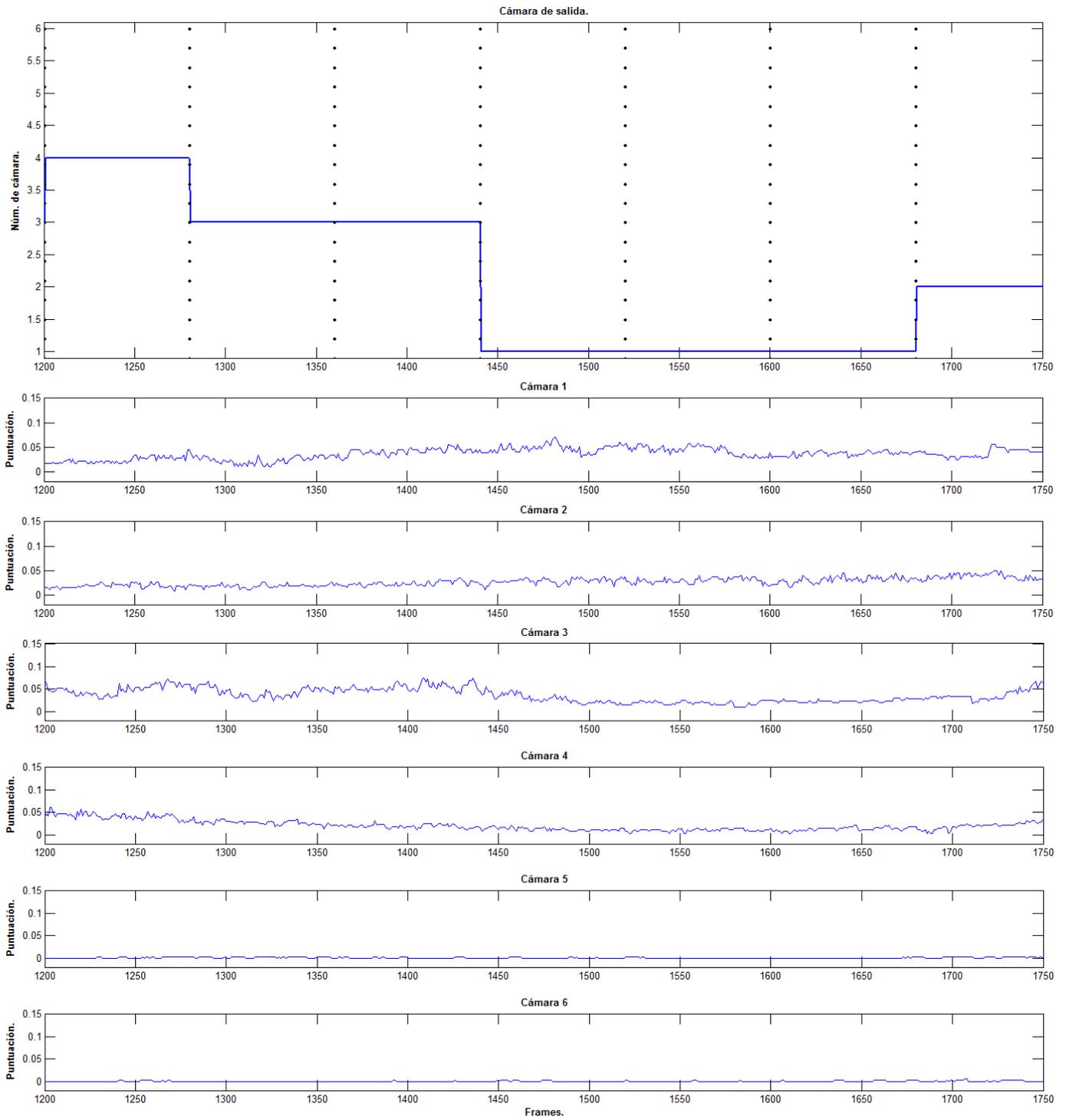


Figura B.29: Puntuación de las cámaras (parcial)

B.3.2. Tenis

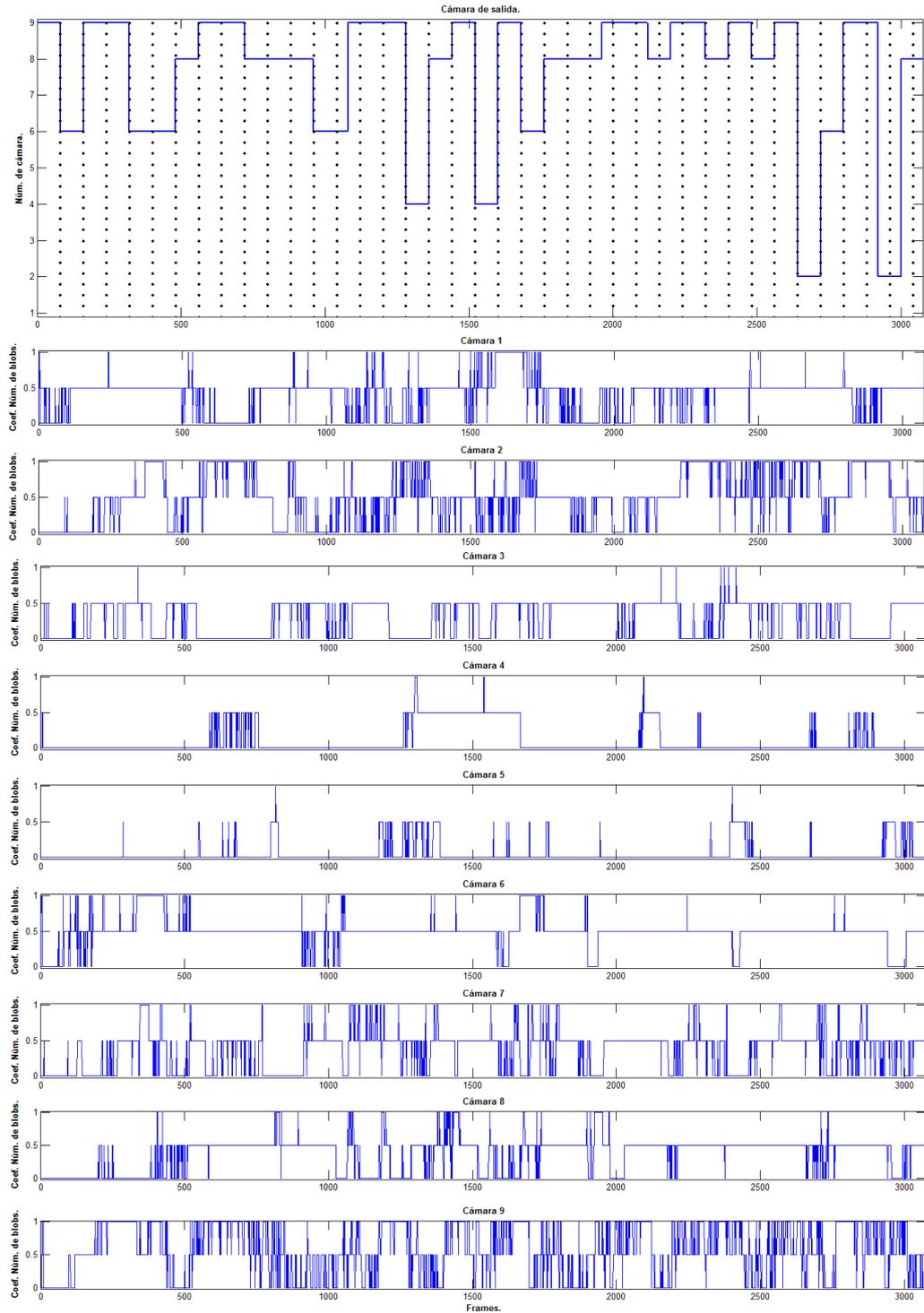


Figura B.30: Número de blobs

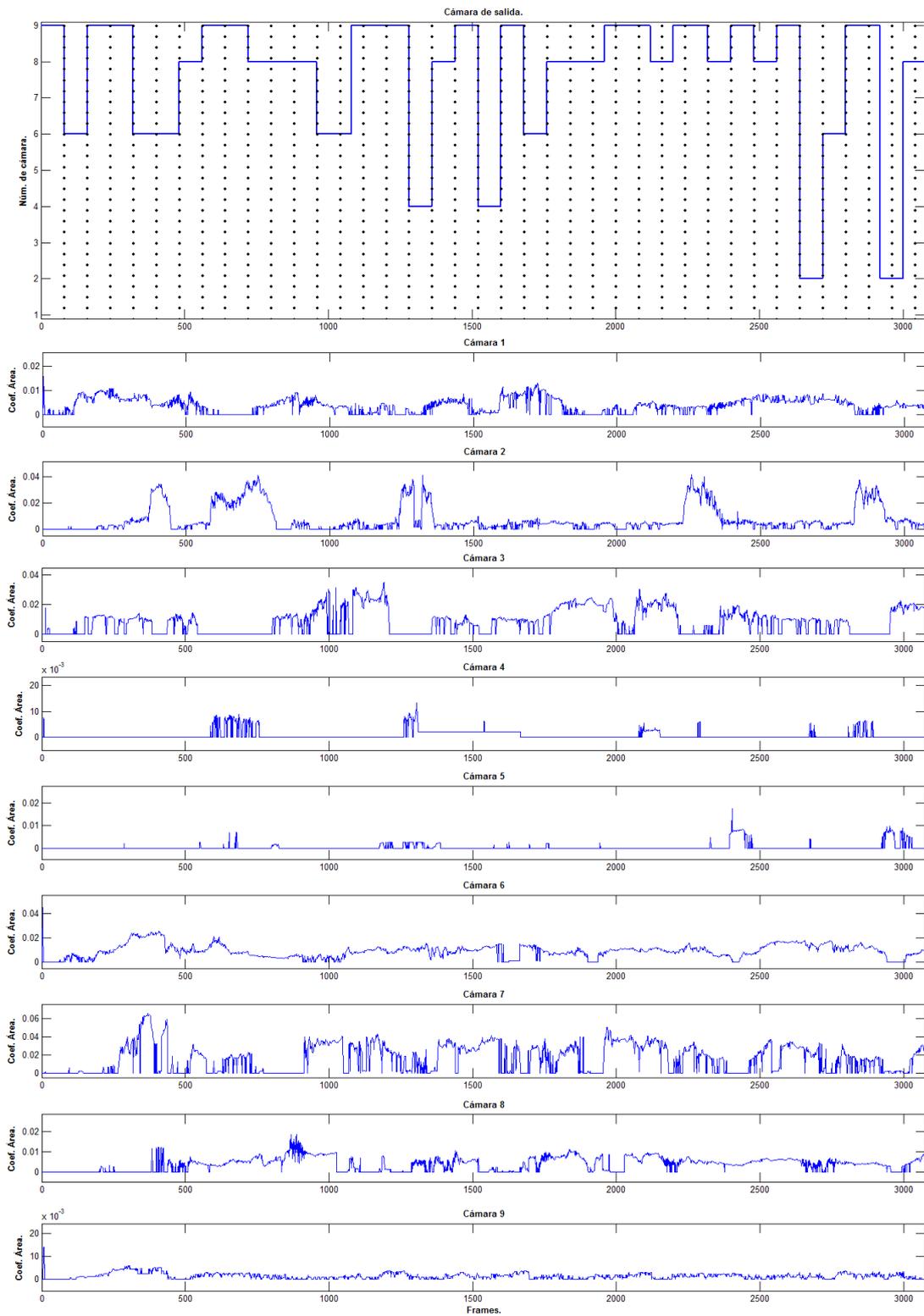


Figura B.31: Coeficiente de área

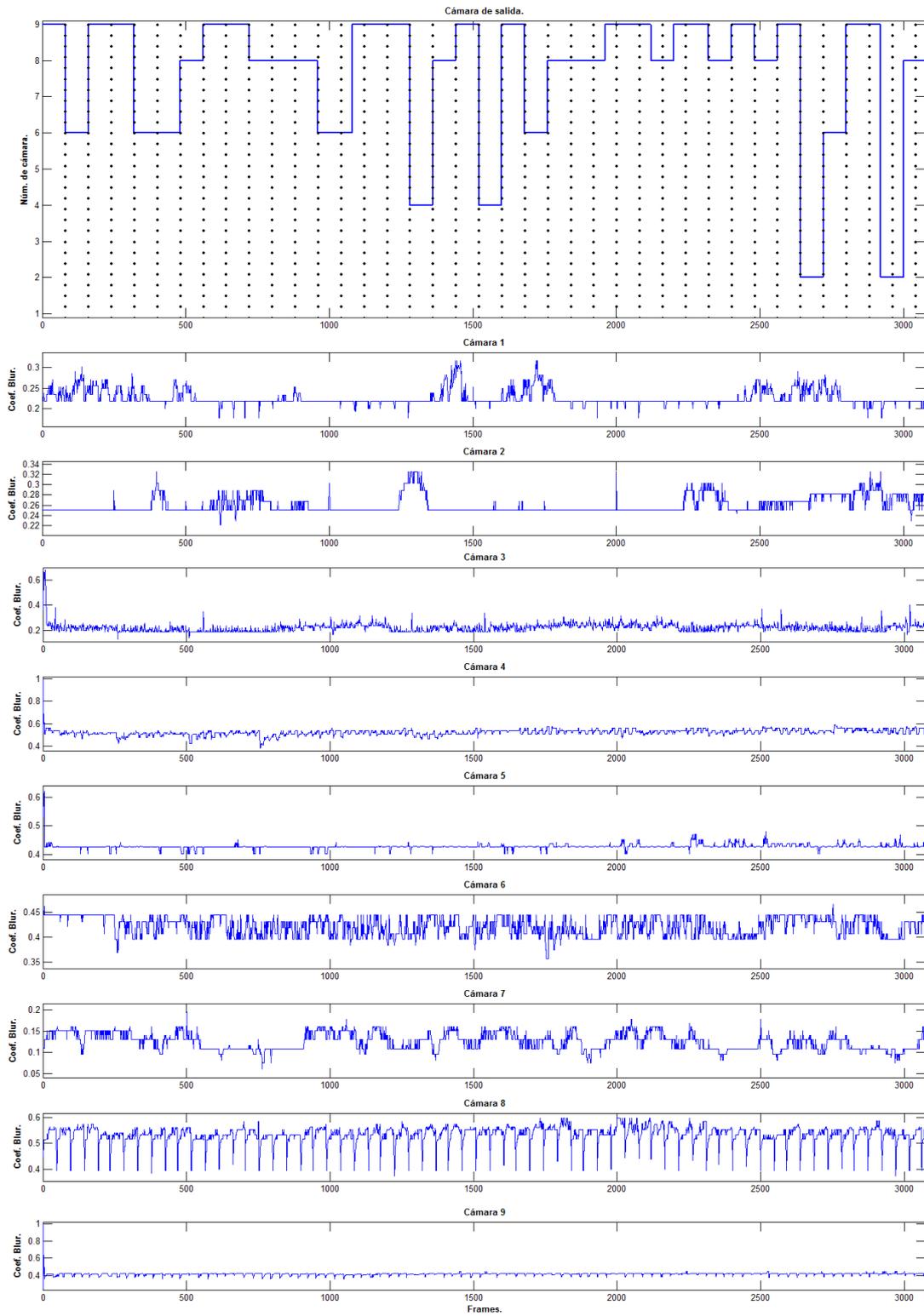


Figura B.32: Coeficiente de emborronamiento

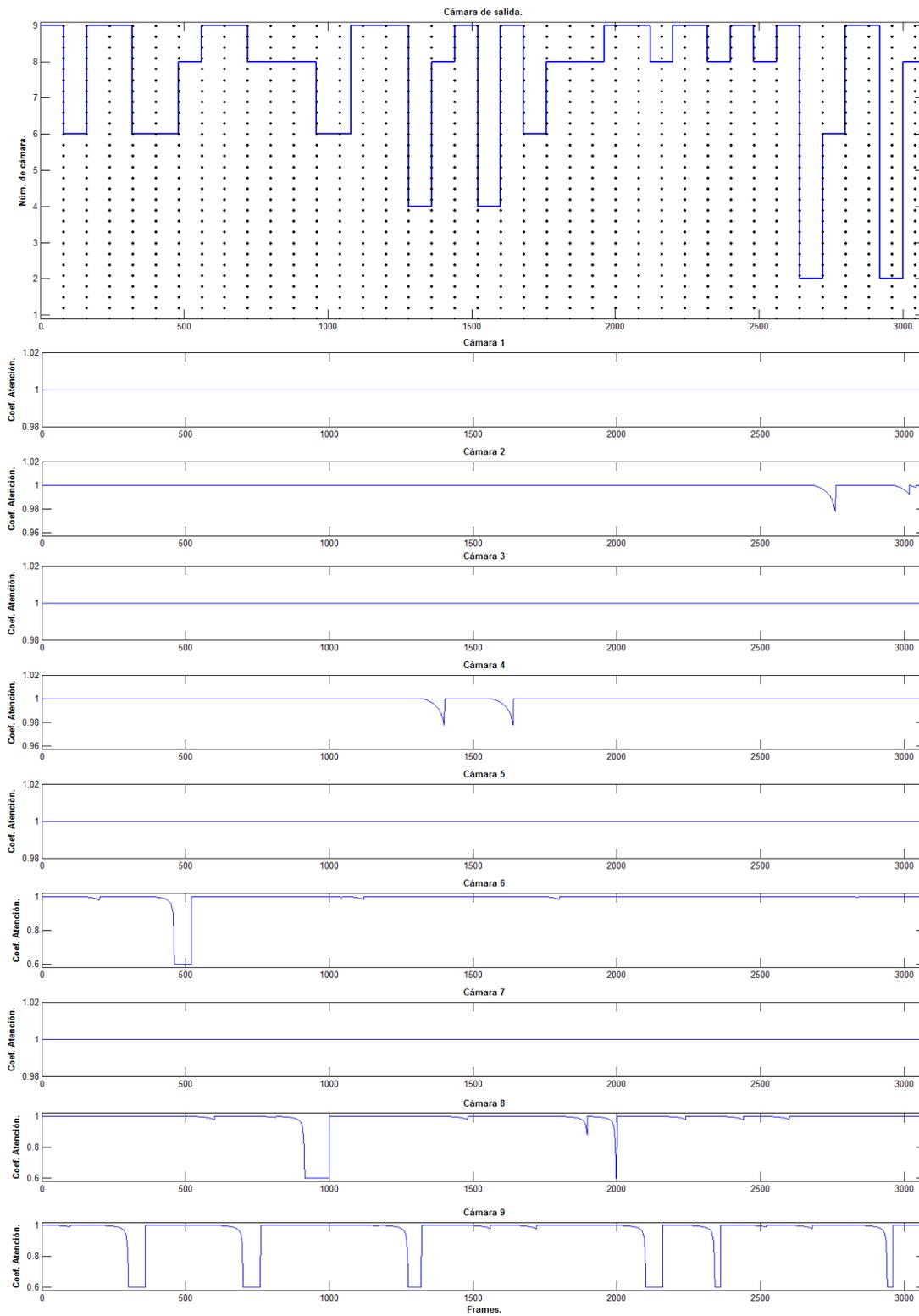


Figura B.33: Coeficiente de atención

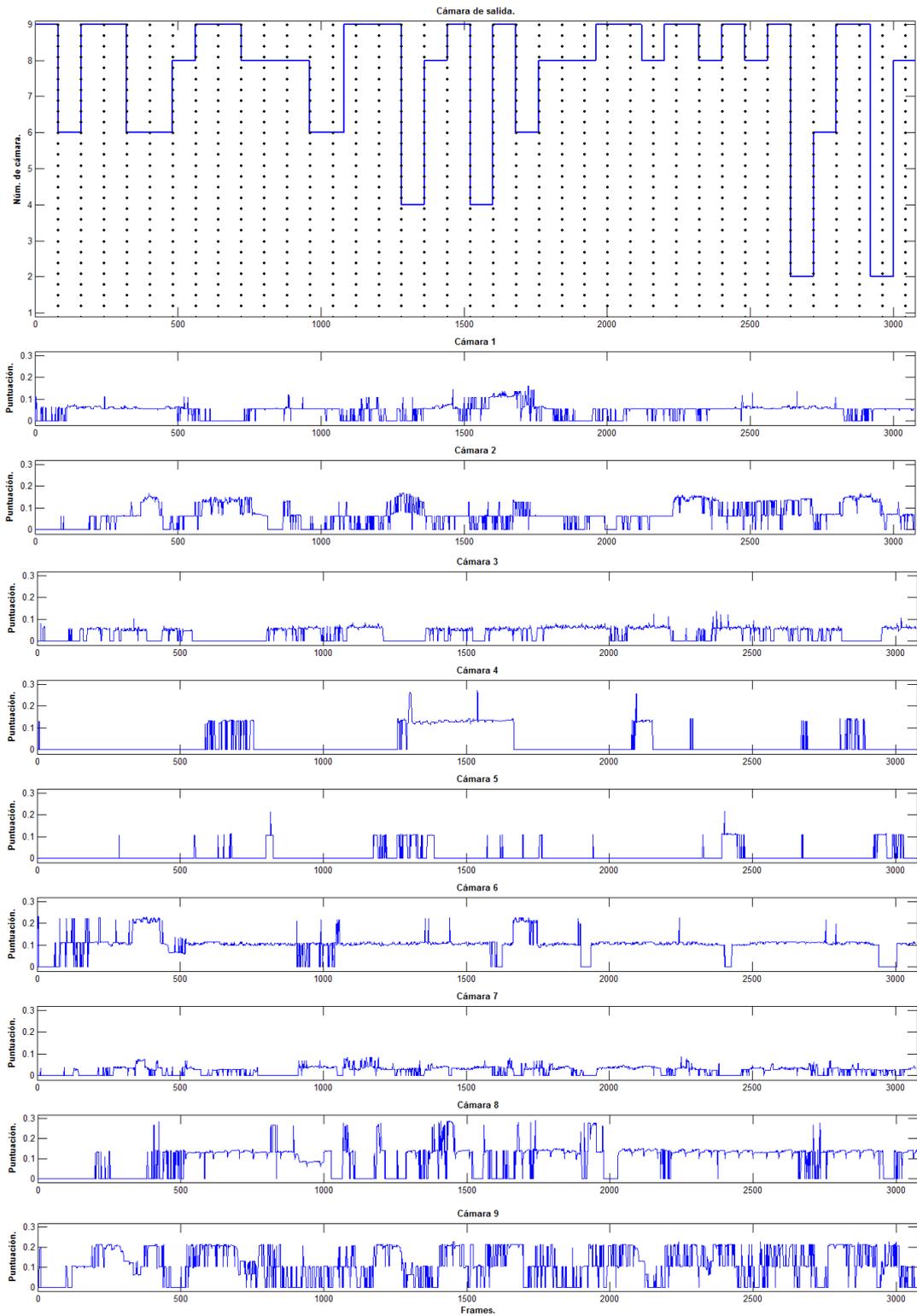


Figura B.34: Puntuación de las cámaras (vídeo total)

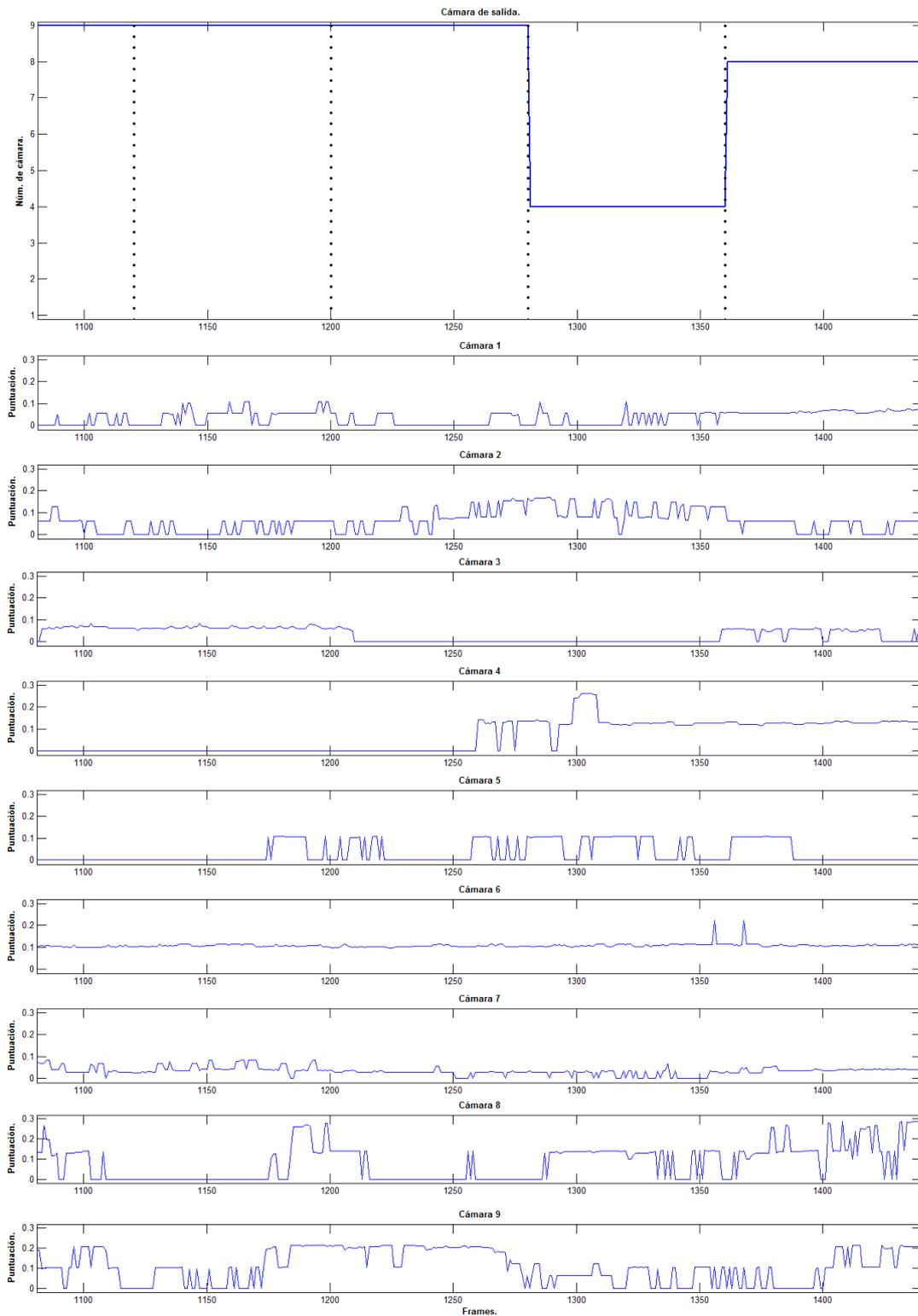


Figura B.35: Puntuación de las cámaras (parcial)

Apéndice C

Presupuesto

1. Ejecución Material

- Compra de ordenador personal (Software incluido) 2.000 €
 - Alquiler de impresora láser durante 6 meses 260 €
 - Material de oficina 150 €
 - Total de ejecución material 2.400 €

1. Gastos generales

- 16 % sobre Ejecución Material 352 €

2. Beneficio Industrial

- 6 % sobre Ejecución Material 132 €

3. Honorarios Proyecto

- 1800 horas a 15 € / hora 27.000 €

4. Material fungible

- Gastos de impresión 280 €
- Encuadernación 200 €

5. Subtotal del presupuesto

- Subtotal Presupuesto 32.774 €

6. I.V.A. aplicable

- 18 % Subtotal Presupuesto 5.899,3 €

7. Total presupuesto

- Total Presupuesto38.673,3 €

Madrid, Abril 2012

El Ingeniero Jefe de Proyecto

Fdo.: Javier Díez Salcedo

Ingeniero Superior de Telecomunicación .

Apéndice D

Pliego de condiciones

Este documento contiene las condiciones legales que guiarán la realización, en este proyecto, de un Sistema de selección automática de vista en sistemas multicámara para ser visto en pantallas de baja resolución. En lo que sigue, se supondrá que el proyecto ha sido encargado por una empresa cliente a una empresa consultora con la finalidad de realizar dicho sistema. Dicha empresa ha debido desarrollar una línea de investigación con objeto de elaborar el proyecto. Esta línea de investigación, junto con el posterior desarrollo de los programas está amparada por las condiciones particulares del siguiente pliego.

Supuesto que la utilización industrial de los métodos recogidos en el presente proyecto ha sido decidida por parte de la empresa cliente o de otras, la obra a realizar se regulará por las siguientes:

Condiciones generales

1. La modalidad de contratación será el concurso. La adjudicación se hará, por tanto, a la proposición más favorable sin atender exclusivamente al valor económico, dependiendo de las mayores garantías ofrecidas. La empresa que somete el proyecto a concurso se reserva el derecho a declararlo desierto.
2. El montaje y mecanización completa de los equipos que intervengan será realizado totalmente por la empresa licitadora.
3. En la oferta, se hará constar el precio total por el que se compromete a realizar la obra y el tanto por ciento de baja que supone este precio en relación con un importe límite si este se hubiera fijado.
4. La obra se realizará bajo la dirección técnica de un Ingeniero Superior de Telecomunicación, auxiliado por el número de Ingenieros Técnicos y Programadores que se estime preciso para el desarrollo de la misma.
5. Aparte del Ingeniero Director, el contratista tendrá derecho a contratar al resto del personal, pudiendo ceder esta prerrogativa a favor del Ingeniero Director, quien no estará obligado a aceptarla.
6. El contratista tiene derecho a sacar copias a su costa de los planos, pliego de condiciones y presupuestos. El Ingeniero autor del proyecto autorizará con su firma las copias solicitadas por el contratista después de confrontarlas.

7. Se abonará al contratista la obra que realmente ejecute con sujeción al proyecto que sirvió de base para la contratación, a las modificaciones autorizadas por la superioridad o a las órdenes que con arreglo a sus facultades le hayan comunicado por escrito al Ingeniero Director de obras siempre que dicha obra se haya ajustado a los preceptos de los pliegos de condiciones, con arreglo a los cuales, se harán las modificaciones y la valoración de las diversas unidades sin que el importe total pueda exceder de los presupuestos aprobados. Por consiguiente, el número de unidades que se consignan en el proyecto o en el presupuesto, no podrá servirle de fundamento para entablar reclamaciones de ninguna clase, salvo en los casos de rescisión.
8. Tanto en las certificaciones de obras como en la liquidación final, se abonarán los trabajos realizados por el contratista a los precios de ejecución material que figuran en el presupuesto para cada unidad de la obra.
9. Si excepcionalmente se hubiera ejecutado algún trabajo que no se ajustase a las condiciones de la contrata pero que sin embargo es admisible a juicio del Ingeniero Director de obras, se dará conocimiento a la Dirección, proponiendo a la vez la rebaja de precios que el Ingeniero estime justa y si la Dirección resolviera aceptar la obra, quedará el contratista obligado a conformarse con la rebaja acordada.
10. Cuando se juzgue necesario emplear materiales o ejecutar obras que no figuren en el presupuesto de la contrata, se evaluará su importe a los precios asignados a otras obras o materiales análogos si los hubiere y cuando no, se discutirán entre el Ingeniero Director y el contratista, sometidos a la aprobación de la Dirección. Los nuevos precios convenidos por uno u otro procedimiento, se sujetarán siempre al establecido en el punto anterior.
11. Cuando el contratista, con autorización del Ingeniero Director de obras, emplee materiales de calidad más elevada o de mayores dimensiones de lo estipulado en el proyecto, o sustituya una clase de fabricación por otra que tenga asignado mayor precio o ejecute con mayores dimensiones cualquier otra parte de las obras, o en general, introduzca en ellas cualquier modificación que sea beneficiosa a juicio del Ingeniero Director de obras, no tendrá derecho sin embargo, sino a lo que le correspondería si hubiera realizado la obra con estricta sujeción a lo proyectado y contratado.
12. Las cantidades calculadas para obras accesorias, aunque figuren por partida alzada en el presupuesto final (general), no serán abonadas sino a los precios de la contrata, según las condiciones de la misma y los proyectos particulares que para ellas se formen, o en su defecto, por lo que resulte de su medición final.
13. El contratista queda obligado a abonar al Ingeniero autor del proyecto y director de obras así como a los Ingenieros Técnicos, el importe de sus respectivos honorarios facultativos por formación del proyecto, dirección técnica y administración en su caso, con arreglo a las tarifas y honorarios vigentes.
14. Concluida la ejecución de la obra, será reconocida por el Ingeniero Director que a tal efecto designe la empresa.

15. La garantía definitiva será del 4 % del presupuesto y la provisional del 2 %.
16. La forma de pago será por certificaciones mensuales de la obra ejecutada, de acuerdo con los precios del presupuesto, deducida la baja si la hubiera.
17. La fecha de comienzo de las obras será a partir de los 15 días naturales del replanteo oficial de las mismas y la definitiva, al año de haber ejecutado la provisional, procediéndose si no existe reclamación alguna, a la reclamación de la fianza.
18. Si el contratista al efectuar el replanteo, observase algún error en el proyecto, deberá comunicarlo en el plazo de quince días al Ingeniero Director de obras, pues transcurrido ese plazo será responsable de la exactitud del proyecto.
19. El contratista está obligado a designar una persona responsable que se entenderá con el Ingeniero Director de obras, o con el delegado que éste designe, para todo relacionado con ella. Al ser el Ingeniero Director de obras el que interpreta el proyecto, el contratista deberá consultarle cualquier duda que surja en su realización.
20. Durante la realización de la obra, se girarán visitas de inspección por personal facultativo de la empresa cliente, para hacer las comprobaciones que se crean oportunas. Es obligación del contratista, la conservación de la obra ya ejecutada hasta la recepción de la misma, por lo que el deterioro parcial o total de ella, aunque sea por agentes atmosféricos u otras causas, deberá ser reparado o reconstruido por su cuenta.
21. El contratista, deberá realizar la obra en el plazo mencionado a partir de la fecha del contrato, incurriendo en multa, por retraso de la ejecución siempre que éste no sea debido a causas de fuerza mayor. A la terminación de la obra, se hará una recepción provisional previo reconocimiento y examen por la dirección técnica, el depositario de efectos, el interventor y el jefe de servicio o un representante, estampando su conformidad el contratista.
22. Hecha la recepción provisional, se certificará al contratista el resto de la obra, reservándose la administración el importe de los gastos de conservación de la misma hasta su recepción definitiva y la fianza durante el tiempo señalado como plazo de garantía. La recepción definitiva se hará en las mismas condiciones que la provisional, extendiéndose el acta correspondiente. El Director Técnico propondrá a la Junta Económica la devolución de la fianza al contratista de acuerdo con las condiciones económicas legales establecidas.
23. Las tarifas para la determinación de honorarios, reguladas por orden de la Presidencia del Gobierno el 19 de Octubre de 1961, se aplicarán sobre el denominado en la actualidad "Presupuesto de Ejecución de Contrata" y anteriormente llamado "Presupuesto de Ejecución Material" que hoy designa otro concepto.

Condiciones particulares

La empresa consultora, que ha desarrollado el presente proyecto, lo entregará a la empresa cliente bajo las condiciones generales ya formuladas, debiendo añadirse las siguientes condiciones particulares:

1. La propiedad intelectual de los procesos descritos y analizados en el presente trabajo, pertenece por entero a la empresa consultora representada por el Ingeniero Director del Proyecto.
2. La empresa consultora se reserva el derecho a la utilización total o parcial de los resultados de la investigación realizada para desarrollar el siguiente proyecto, bien para su publicación o bien para su uso en trabajos o proyectos posteriores, para la misma empresa cliente o para otra.
3. Cualquier tipo de reproducción aparte de las reseñadas en las condiciones generales, bien sea para uso particular de la empresa cliente, o para cualquier otra aplicación, contará con autorización expresa y por escrito del Ingeniero Director del Proyecto, que actuará en representación de la empresa consultora.
4. En la autorización se ha de hacer constar la aplicación a que se destinan sus reproducciones así como su cantidad.
5. En todas las reproducciones se indicará su procedencia, explicitando el nombre del proyecto, nombre del Ingeniero Director y de la empresa consultora.
6. Si el proyecto pasa la etapa de desarrollo, cualquier modificación que se realice sobre él, deberá ser notificada al Ingeniero Director del Proyecto y a criterio de éste, la empresa consultora decidirá aceptar o no la modificación propuesta.
7. Si la modificación se acepta, la empresa consultora se hará responsable al mismo nivel que el proyecto inicial del que resulta el añadirla.
8. Si la modificación no es aceptada, por el contrario, la empresa consultora declinará toda responsabilidad que se derive de la aplicación o influencia de la misma.
9. Si la empresa cliente decide desarrollar industrialmente uno o varios productos en los que resulte parcial o totalmente aplicable el estudio de este proyecto, deberá comunicarlo a la empresa consultora.
10. La empresa consultora no se responsabiliza de los efectos laterales que se puedan producir en el momento en que se utilice la herramienta objeto del presente proyecto para la realización de otras aplicaciones.
11. La empresa consultora tendrá prioridad respecto a otras en la elaboración de los proyectos auxiliares que fuese necesario desarrollar para dicha aplicación industrial, siempre que no haga explícita renuncia a este hecho. En este caso, deberá autorizar expresamente los proyectos presentados por otros.
12. El Ingeniero Director del presente proyecto, será el responsable de la dirección de la aplicación industrial siempre que la empresa consultora lo estime oportuno. En caso contrario, la persona designada deberá contar con la autorización del mismo, quien delegará en él las responsabilidades que ostente.