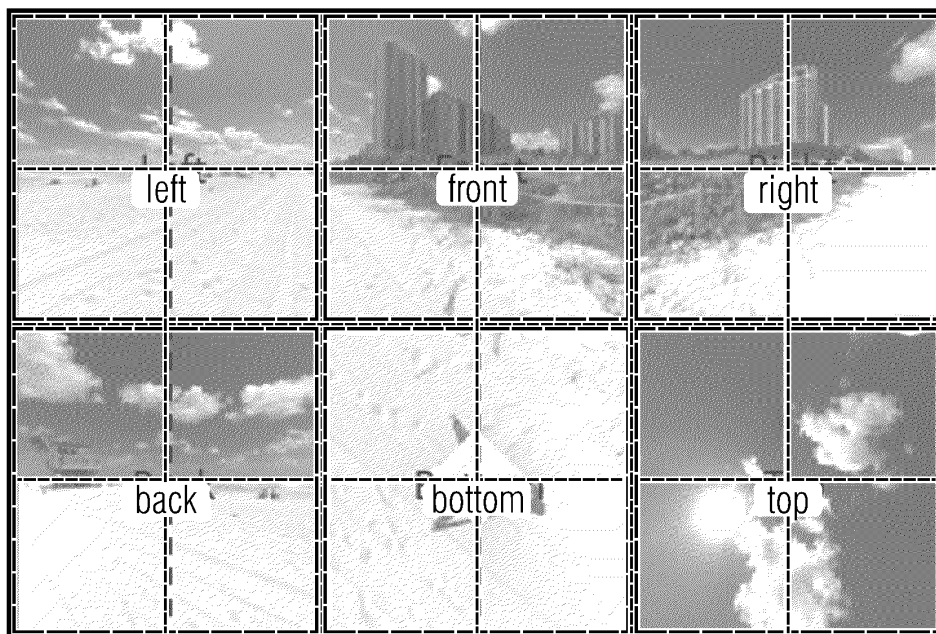
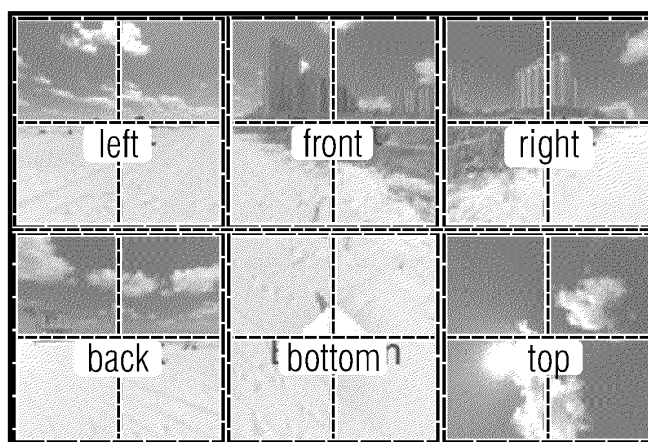
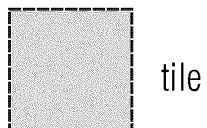




high resolution



low resolution

Fig. 1

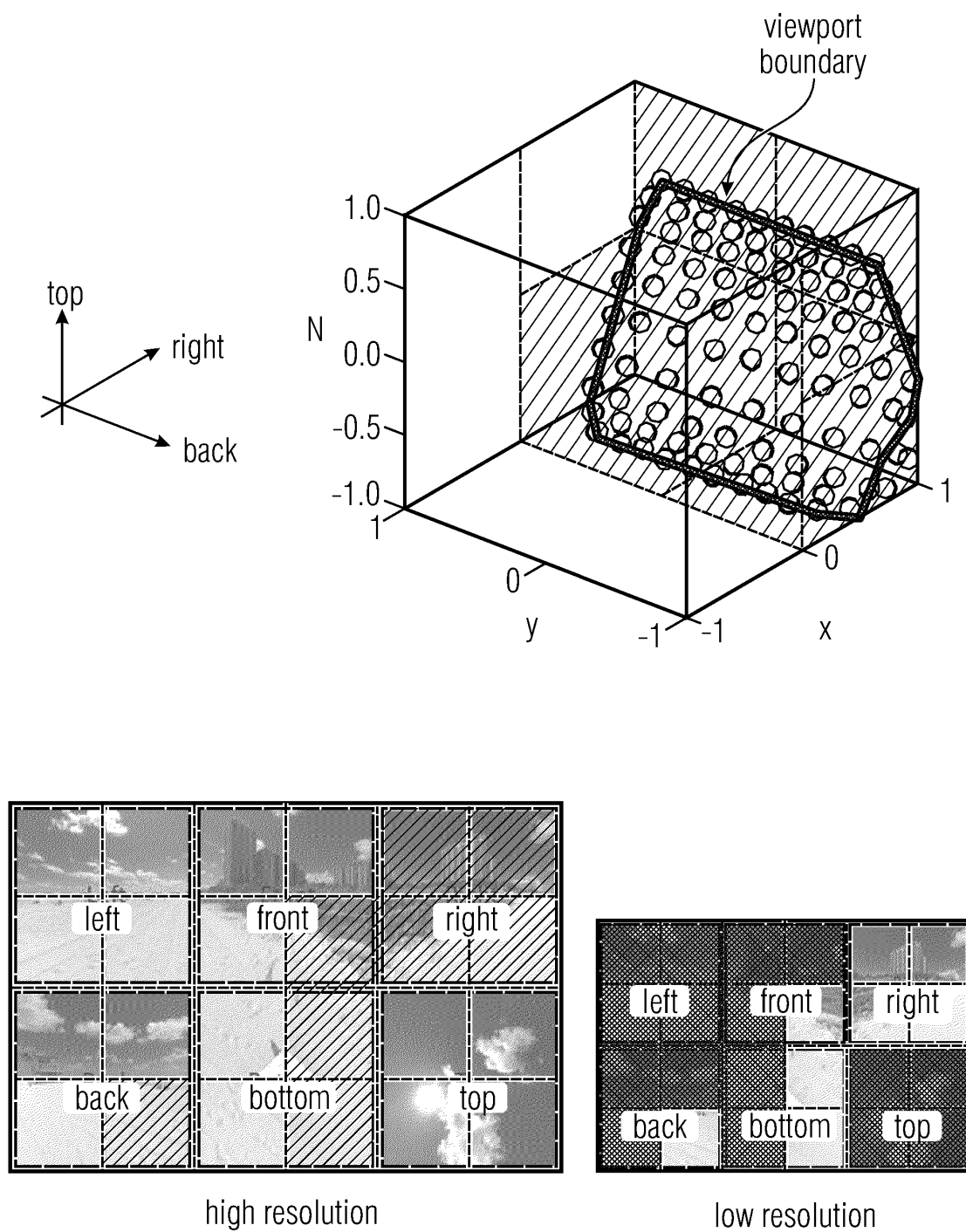


Fig. 2

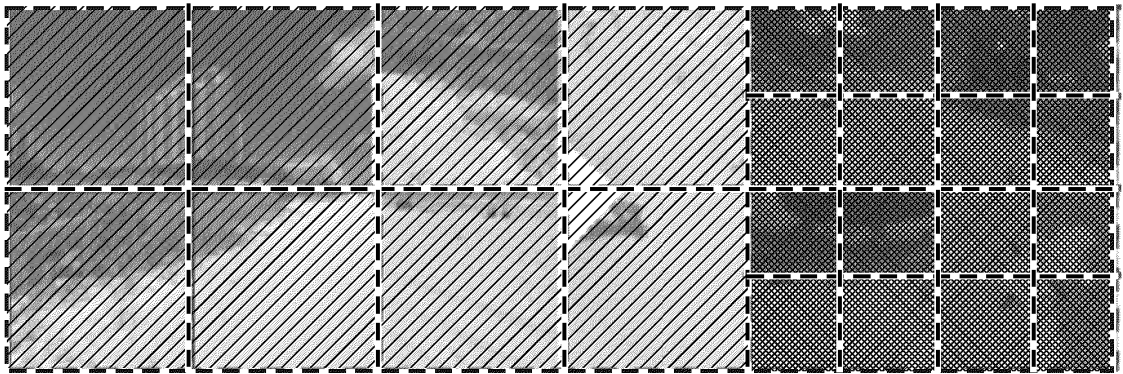


Fig. 3

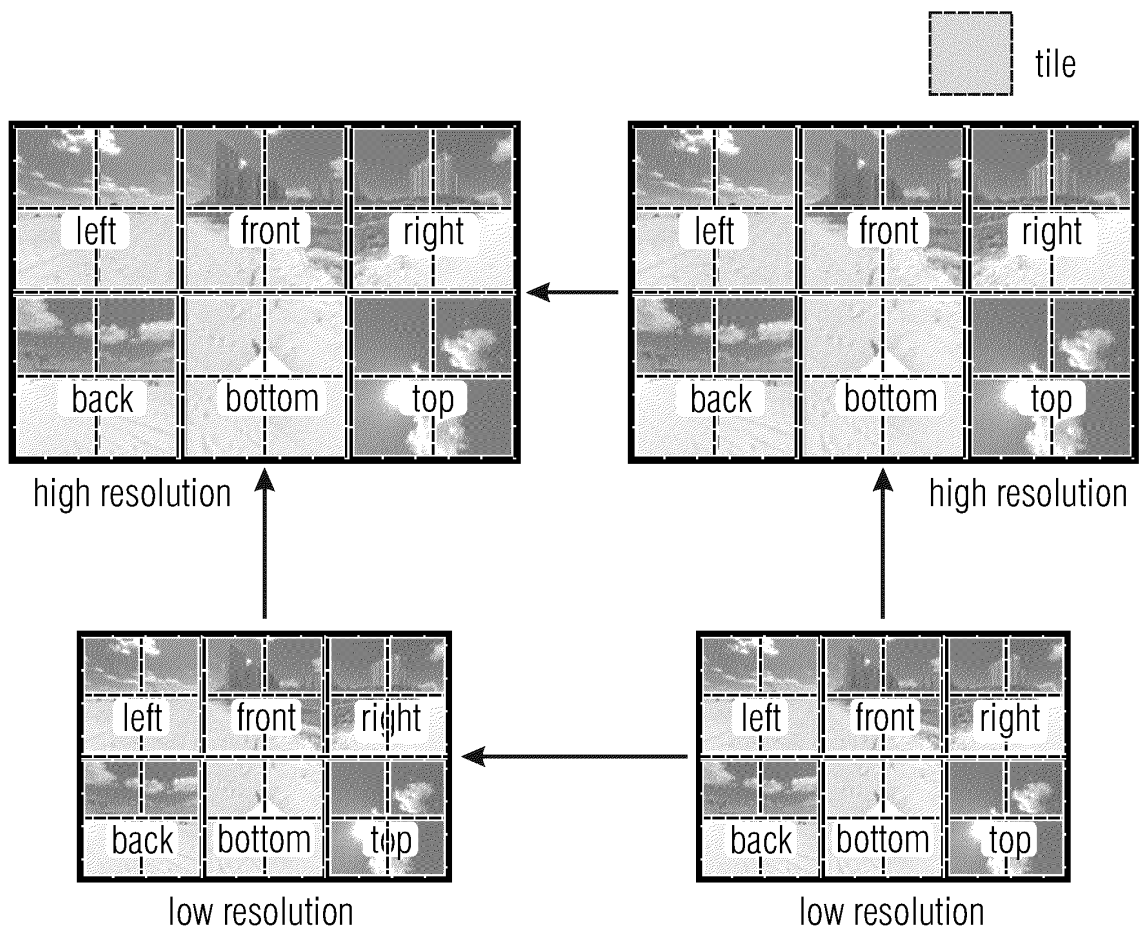


Fig. 4

general_constraint_info() {	Descriptor
general_progressive_source_flag	u(1)
...	
subpictures_in_ols_aligned_flag	u(1)
}	

Fig. 5

general_constraint_info() {	Descriptor
general_progressive_source_flag	u(1)
...	
unique_subpictures_ids_in_ols_flag	u(1)
}	

Fig. 6

general_constraint_info() {	Descriptor
general_progressive_source_flag	u(1)
...	
aligned_subpictures_ids_in_ols_flag	u(1)
}	

Fig. 7

seq_parameter_set_rbsp() {	Descriptor
sps_decoding_parameter_set_id	u(4)
...	
subpics_present_flag	u(1)
if(subpics_present_flag) {	
sps_num_subpics_minus1	u(8)
for(i = 0; i <= sps_num_subpics_minus1; i++) {	
subpic_ctu_top_left_x[i]	u(v)
subpic_ctu_top_left_y[i]	u(v)
subpic_width_minus1[i]	u(v)
subpic_height_minus1[i]	u(v)
subpic_treated_as_pic_flag[i]	u(1)
loop_filter_across_subpic_enabled_flag[i]	u(1)
}	
}	
...	u(1)

Fig. 8

Sequence Parameter Set

sps_subpic_id_present_flag	u(1)
if(sps_subpics_id_present_flag) {	
sps_subpic_id_signalling_present_flag	u(1)
if(sps_subpics_id_signalling_present_flag) {	
sps_subpic_id_len_minus1	ue(v)
for(i = 0; i <= sps_num_subpics_minus1; i++)	
sps_subpic_id[i]	u(v)
}	
}	

Fig. 9

Picture Parameter Set

pps_subpic_id_signalling_present_flag	u(1)
if(pps_subpics_id_signalling_present_flag) {	
pps_num_subpics_minus1	ue(v)
pps_subpic_id_len_minus1	ue(v)
for(i = 0; i <= pps_num_subpic_minus1; i++)	
pps_subpic_id[i]	u(v)
}	

Fig. 10

Picture Header

if(sps_subpic_id_present_flag && !sps_subpic_id_signalling_flag) {	
ph_subpic_id_signalling_present_flag	u(1)
if(ph_subpics_id_signalling_present_flag) {	
ph_subpic_id_len_minus1	ue(v)
for(i = 0; i <= sps_num_subpics_minus1; i++)	
ph_subpic_id[i]	u(v)
}	
}	

Fig. 11

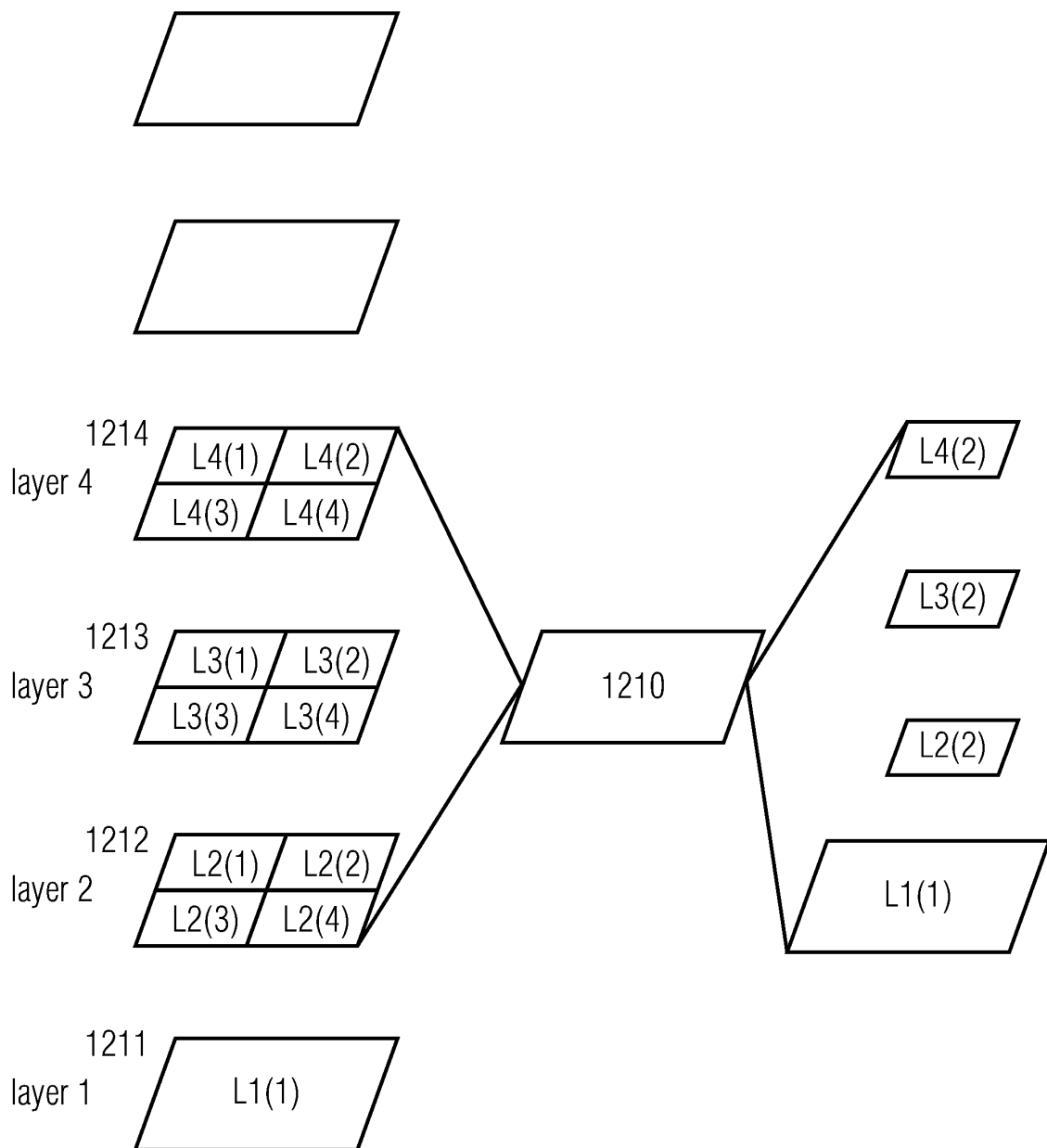


Fig. 12

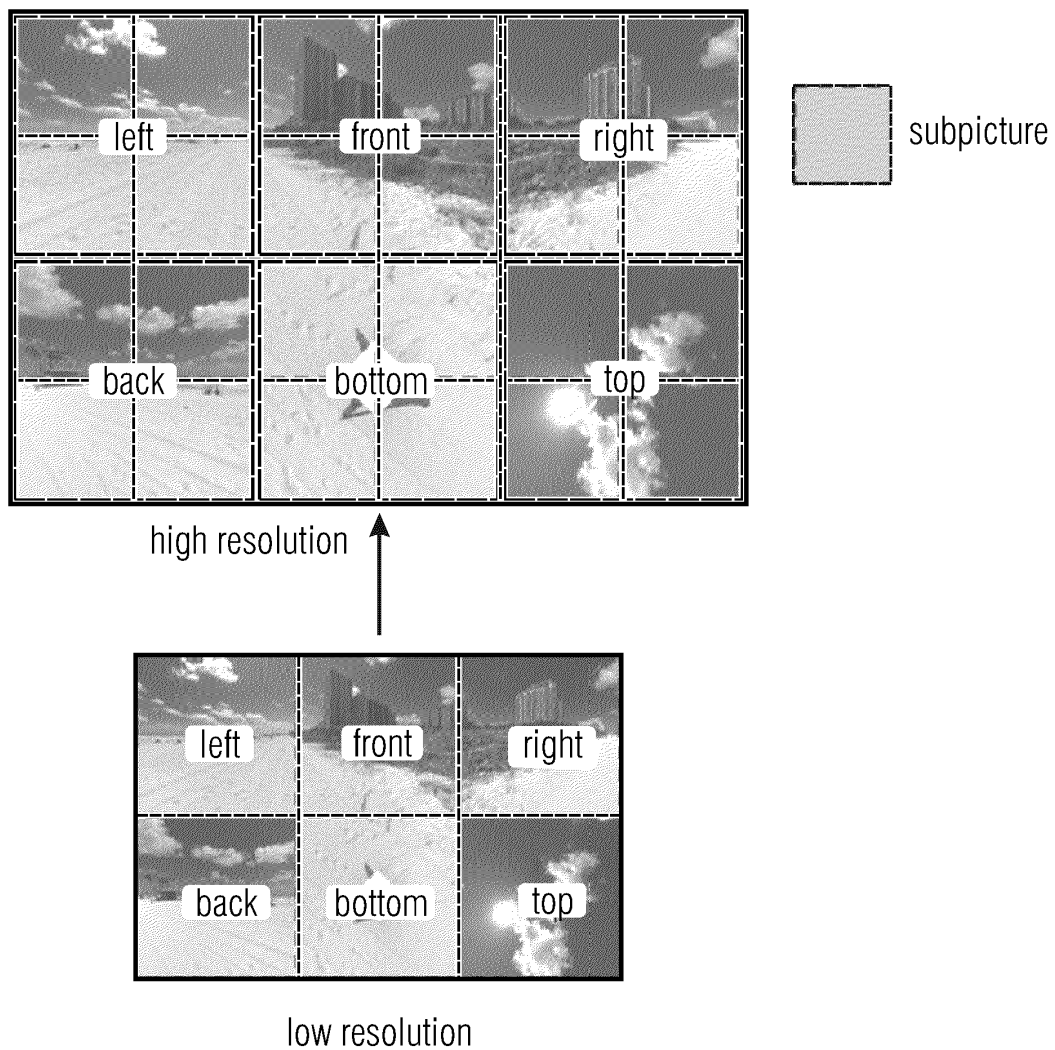


Fig. 13

general_constraint_info() {	Descriptor
general_progressive_source_flag	u(1)
...	
subpictures_in_ols_coherent_flag	u(1)
}	

Fig. 14

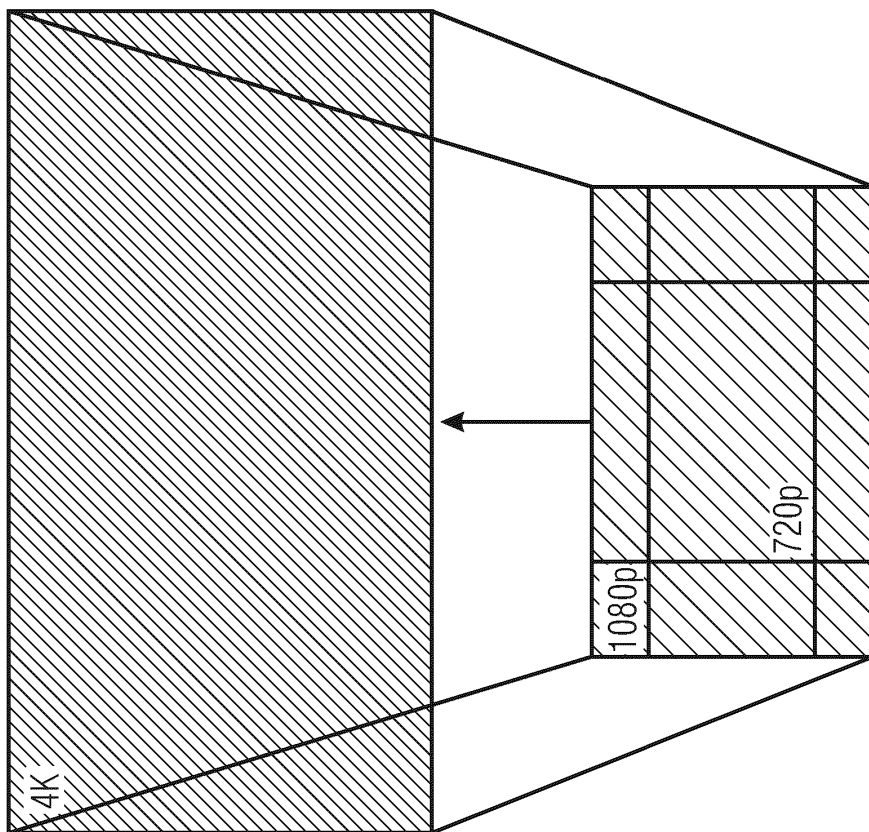


Fig. 15

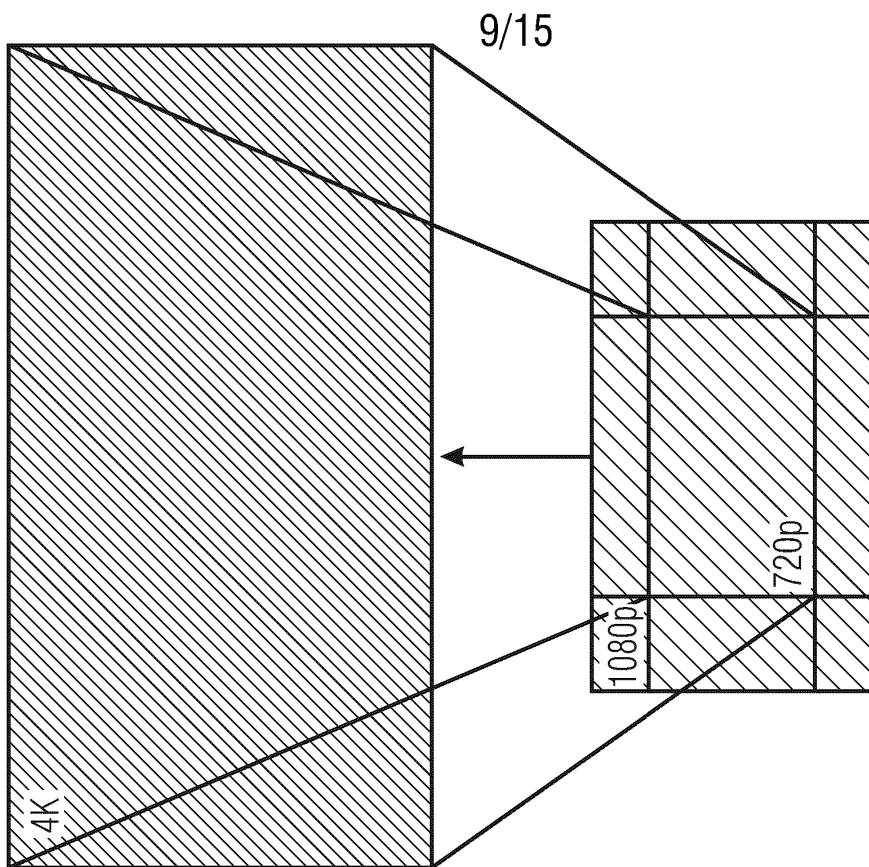


Fig. 16

video_parameter_set_rbsp() {	Descriptor
...	
for(i = 0; i <= vps_max_layers_minus1; i++) {	
vps_layer_id[i]	u(6)
if(i > 0 && !vps_all_independent_layers_flag) {	
vps_independent_layer_flag[i]	u(1)
if(!vps_independent_layer_flag[i])	
for(j = 0; j < i; j++)	
vps_direct_ref_layer_flag[i][j]	u(1)
}	
}	
if(vps_max_layers_minus1 > 0) {	
if(vps_all_independent_layers_flag)	
each_layer_is_an_ols_flag	u(1)
if(!each_layer_is_an_ols_flag) {	
if(!vps_all_independent_layers_flag)	
ols_mode_idc	u(2)
if(ols_mode_idc == 2) {	
num_output_layer_sets_minus1	u(8)
for(i = 1; i <= num_output_layer_sets_minus1; i++)	
for(j = 0; j <= vps_max_layers_minus1; j++)	
ols_output_layer_flag[i][j]	u(1)
reduced_subpic_reference_flag	u(1)
num_sub_pic_ids	u(8)
for(i = 0; i <= num_sub_pic_ids; i++)	
subPicIdToDecodeForReference	u(v)
}	
}	
}	
}	
...	
}	

Fig. 17

video_parameter_set_rbsp() {	Descriptor
vps_video_parameter_set_id	u(4)
...	
vps_num_ptls	u(8)
for(i = 0; i < vps_num_ptls; i++) {	
if(i > 0)	
pt_present_flag[i]	u(1)
if(vps_max_sublayers_minus1 > 0 && !vps_all_layers_same_num_sublayers_flag)	
ptl_max_temporal_id[i]	u(3)
}	
while(!byte_aligned())	
vps_ptl_byte_alignment_zero_bit /* equal to 0 */	u(1)
for(i = 0; i < vps_num_ptls; i++)	
profile_tier_level(pt_present_flag[i], ptl_max_temporal_id[i])	
for(i = 0; i < TotalNumOls; i++)	
if(NumLayersInOls[i] > 1 && vps_num_ptls > 1) {	
ols_ptl_idx[i]	u(8)
subpic_add_ptl_present_flag[i]	u(1)
if(subpic_add_ptl_present_flag[i])	
subpic_add_ols_ptl_idx[i]	u(8)
}	
...	
vps_extension_flag	u(1)
if(vps_extension_flag)	
while(more_rbsp_data())	
vps_extension_data_flag	u(1)
rbsp_trailing_bits()	
}	

Fig. 18

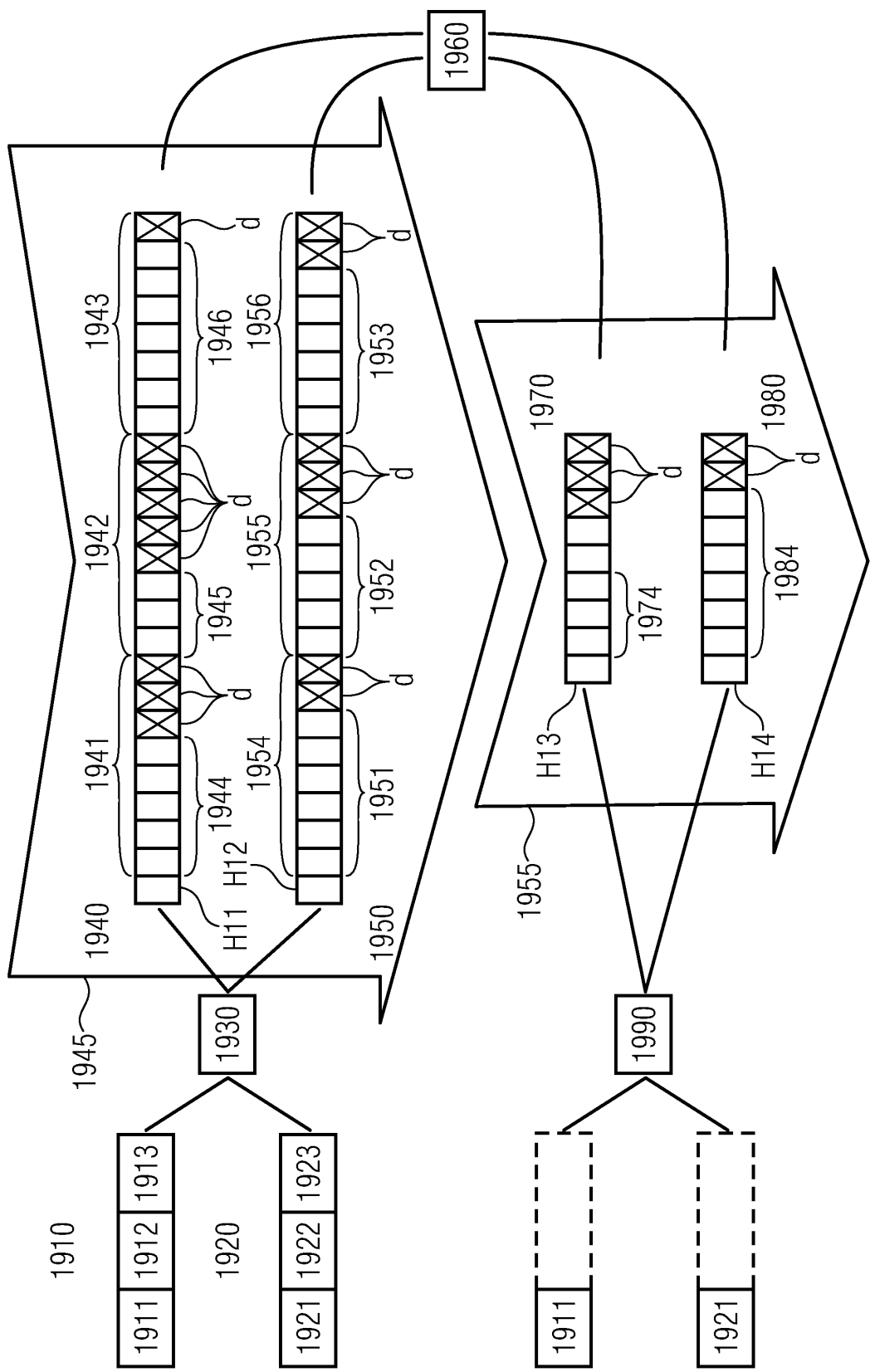


Fig. 19

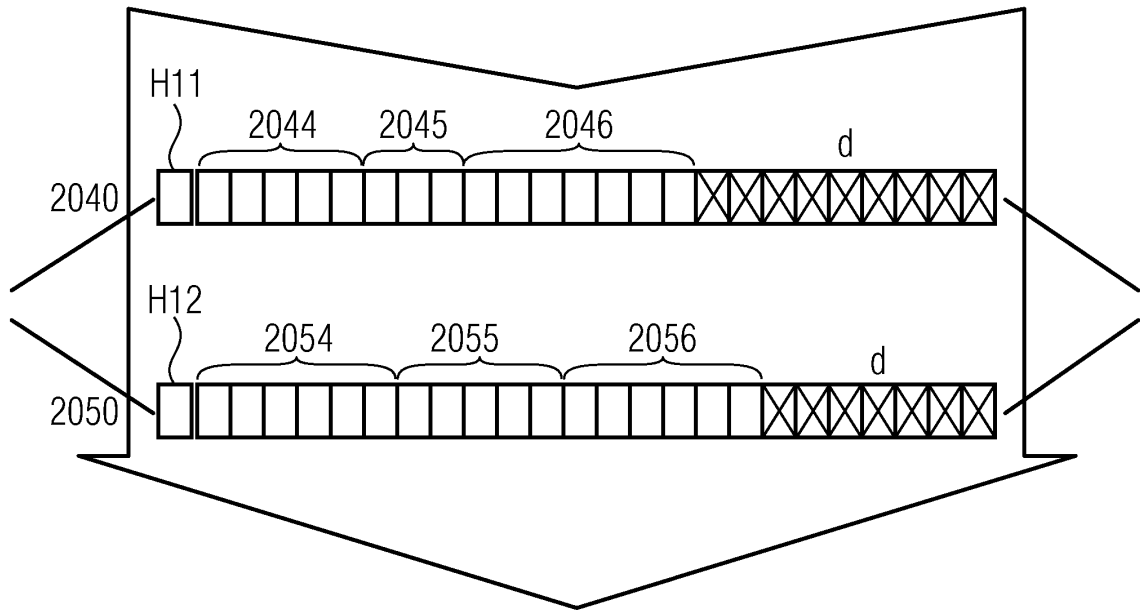


Fig. 20

subpic_ctu_top_left_x[i]	u(v)
subpic_ctu_top_left_y[i]	u(v)
subpic_width_minus1[i]	u(v)
subpic_height_minus1[i]	u(v)

Fig. 21

slice_header() {	Descriptor
slice_pic_order_cnt_lsb	u(v)
if(subpics_present_flag)	
slice_subpic_id	u(v)
...	

Fig. 22

slice_header() {	Descriptor
slice_pic_order_cnt_lsb_plus1	u(v)
one_bit	u(1)
if(subpics_present_flag){	
slice_subpic_id_plus1	u(v)
one_bit	u(1)
}	
...	

Fig. 23

slice_header() {	Descriptor
slice_pic_order_cnt_lsb_plus1	u(v)
if(subpics_present_flag){	
slice_subpic_id_plusVal	u(v)
one_bit	u(1)
}	
...	

Fig. 24

general_constraint_info() {	Descriptor
general_progressive_source_flag	u(1)
...	
cabac_zero_word_for_each_subpictures_present_flag	u(1)
}	

Fig. 25

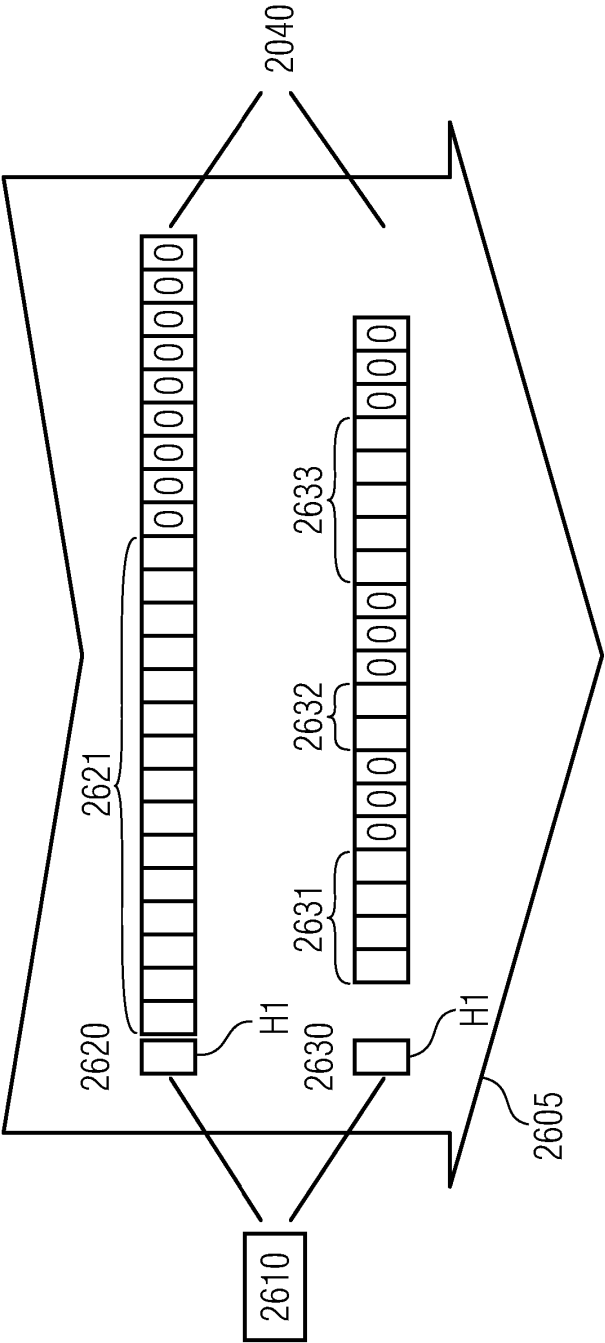


Fig. 26